U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

MANIKA KAR

SUMMARIZATION OF CHANGES IN DYNAMIC TEXT COLLECTIONS

SUPERVISOR: SÉRGIO NUNES (PH.D.)
CO-SUPERVISOR: CRISTINA RIBEIRO (PH.D.)

Doctoral Program in Informatics of the Universities of Minho, Aveiro, and Porto (MAPi)

universidade de aveiro
Universidade do Minho
U.PORTO

E-MAIL:
manika.kar@gmail.com

ABSTRACT

In the area of Information Retrieval, the task of automatic text summarization usually assumes a static underlying collection of documents, disregarding the temporal dimension of each document. However, in real world settings, collections and individual documents rarely stay unchanged over time. In this context, previous work addressing the summarization of web documents has simply discarded the dynamic nature of the web, considering only the latest published version of each individual document. This thesis addresses a problem that gains relevance in this context—the automatic summarization of changes in dynamic text collections. The goal is to develop new methods and techniques that are able to automatically summarize the significant changes made to a document given a temporal period.

In this thesis, we present our contributions into two phases. In the first phase, four different approaches are proposed to generate the summaries of changes using extractive summarization techniques. First, individual terms are scored and then this information is used to rank and select sentences to produce the final summary. The first approach provides a baseline and is adapted from previous work in this area. In the second approach, the temporal aspect of each term is investigated by considering the joint probabilities of both insertion and deletion events over a set of document versions within the given period. The third approach is based on Latent Dirichlet Allocation (LDA) model for finding latent topic structures associated to changes. The fourth approach is a combination of the previous two approaches in which the top-ranked sentences generated from the third approach are re-ranked using a combined score from the second and third approaches.

In the second phase, the exploration of the LDA model proceeds to detect multiple significant changes for a wider temporal interval. The number of latent changes for LDA model is estimated using Bayesian model selection without the constraint of specifying a default model selection criterion a *priori*. The number of estimated latent changes is thereafter assumed as the number of different categories of candidate changes, which are likely to include both significant and non-significant ones. For each category of candidate changes, a burst region is identified. A set of sentences is then selected from the burst region to present a meaningful and coherent summary for each significant topic. These summaries are generated hierarchically—a summary is presented for each significant topic in an intermediate level and, at the top-level, a single summary is generated in order to consolidate the most significant changes.

To evaluate the results, we use a collection of articles from Wikipedia, including their revision history. For each article, a temporal interval and a reference summary from the article's content are selected manually. The summaries produced by each of the approaches are evaluated comparatively to the manual summaries using ROUGE metrics. It is observed that the approach using the LDA model outperforms all the other approaches. In the second phase, the results are compared against the results of the LDA-based approach using our proposed best match mapping metric. The comparison shows that, although the evaluation scores for the top summaries are similar, the performance for the intermediate summaries is improved marginally. However, other aspects of the summaries, namely focus and coherence are

assessed through pairwise comparison, proving that the summaries generated in the second phase are preferred.

RESUMO

Na área da Recuperação de Informação, a tarefa de sumarização automática de textos assume tipicamente a existência de uma coleção estática de documentos, não considerando a dimensão temporal de cada documento. No entanto, em cenários reais, coleções e documentos individuais raramente permanecem inalterados ao longo do tempo. Neste contexto, o trabalho existente na área da sumarização de documentos web simplesmente descarta a natureza dinâmica da web, considerando apenas a última versão publicada de cada documento individual. Esta tese aborda um problema que ganha relevância nesse contexto—a sumarização automática de mudanças em coleções de texto dinâmicas. O objetivo é desenvolver novos métodos e técnicas capazes de resumir automaticamente as mudanças significativas feitas num documento para um determinado período temporal.

Nesta tese, apresentamos a nossa contribuição em duas fases. Na primeira fase, quatro abordagens são propostas para gerar os resumos das alterações usando técnicas de sumarização extrativas. Em primeiro lugar, os termos individuais são ponderados e, com base nessa informação, são selecionadas as frases para a geração do resumo final. A primeira abordagem fornece um resultado de referência e resulta da adaptação de trabalho anterior nesta área. Na segunda abordagem, o aspeto temporal de cada termo é investigado, considerando as probabilidades conjuntas de ambos os eventos de inserção e remoção num conjunto de revisões, dentro do intervalo temporal definido. A terceira abordagem baseia-se no modelo Latent Dirichlet Allocation (LDA) para encontrar estruturas de tópicos latentes associados às alterações. A quarta abordagem é uma combinação das duas anteriores, em que as frases geradas com base na terceira abordagem são reclassificadas com base numa ponderação da segunda e terceira.

Na segunda fase, o modelo LDA é usado para detetar múltiplas mudanças significativas para um intervalo de tempo mais alargado. O número de alterações latentes para o modelo LDA é estimado utilizando um modelo Bayesiano sem a restrição de existência de um critério de seleção do modelo padrão a *priori*. O número estimado de alterações latentes é, portanto, considerado como o número de diferentes categorias de alterações candidatas e poderá incluir alterações significativas ou não significativas. Para cada categoria de mudanças candidatas, é identificada uma região de atividade elevada. Um conjunto de frases é então selecionado a partir de cada região de atividade elevada para a apresentação de um resumo completo e coerente para cada tópico relevante. Estes resumos são gerados de forma hierárquica—um resumo independente é apresentado para cada tópico significativo a um nível intermédio e, no nível superior, um resumo simples é gerado com o objetivo de consolidar as mudanças mais significativas.

Para avaliar os resultados, usamos uma coleção de artigos da Wikipédia, incluindo o respetivo historial de revisões. Para cada artigo, é selecionado manualmente um intervalo temporal e um sumário de referência. Os sumários produzidos por cada uma das abordagens são comparados com os sumários manuais usando métricas ROUGE. Observa-se que a abordagem baseada no modelo LDA supera todas as outras abordagens. Na segunda fase, os resultados são comparados com os resultados da abordagem baseada em LDA. A comparação mostra que, embora os resultados de avaliação para

os melhores resumos sejam semelhantes, o desempenho para os resumos intermédios é melhorado marginalmente. No entanto, outros aspetos dos resumos, nomeadamente o foco e a coerência, são avaliados através de comparações emparelhadas, provando que os resumos gerados na segunda fase são os preferidos.

ACKNOWLEDGEMENTS

I heard about the challenges and hardships in carrying out the PhD Theses but I actually faced them with my own experiences while pursuing the MAPi joint Doctoral program in Porto, Portugal. Erasmus Mundus Mover project gave me the opportunity to see the other side of the world and explore myself inter-culturally with new people, ideas, places, foods, weathers, festivals and many more. During this mobility period what I have learnt will always be part of whoever I become. I owe thanks to many individuals who have made this journey possible. This thesis could not have been completed without their help.

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Sérgio Nunes and and my co-supervisor Professor Cristina Ribeiro. They have provided invaluable guidance, constructive feedback and encouragement at every stage of the work. I appreciate my supervisor's research vision that recognized this research area as an important challenge to be explored. An important advice that I got from him was to make a habit of writing little by little everyday besides doing technical tasks; I felt that this really works out for me. The writing of this thesis would have been far more difficult without his writing tips. I am also grateful to him for helping me to take feasible decisions on time and for the interesting and useful discussions we had. My co-supervisor has been a role model for me. I admire her work ethic, intellectual rigor and management skills. A special thanks goes to the former Director of MAPi, Professor Gabriel David, who has always taken care of administrative procedures whenever I needed.

I am specially grateful to Dr. Tapan Kumar Bhowmik; without his motivation and encouragement I would not have considered pursuing a Doctoral program so far away from home. He always motivated me when I was confronted with obstacles, even in the hardest moments. His optimism undoubtedly contributed to this achievement. I would like to express my regards to Professor Swapan Kumar Parui, under whose supervision I completed my Masters dissertation. Whenever I discussed my research with him, I always got valuable advice. My special thanks to Subhra Sundar Goswami, who provided me with information about the whole application procedures of Erasmus Mundus Project.

I would also like to thank all InfoLab members. InfoLab always has a welcoming work environment which makes people happy. I express my thanks to João Rocha da Silva and appreciate his help for showing me how to work more efficiently with useful tools. I must thank Ana Castro Paiva and Sofia Silva Santos, who have handled the scholarship and registration procedures perfectly. I also give thanks to all the Indian friends I met in Porto for their efforts of arranging spicy Indian cuisines which make me feel a homely atmosphere.

Moreover, many thanks to the anonymous reviewers whose very insightful comments have greatly improved my work.

Last but not least, I would like to thank my parents and my brother Sandip Kar, who always support me unconditionally. Though I know they miss me a lot, they cheer me up whenever I feel down. They are always in my heart.

SUPPORT FUNDING ACKNOWLEDGEMENTS

This work's main funding support was given by the Erasmus Mundus Mover Project. This work is also supported by the "NORTE—07—0124—FEDER—000059" project by the North Portugal Regional Operational Programme (ON.2—O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT).

"And once the storm is over, you won't remember how you made it through, how you managed to survive. You won't even be sure, whether the storm is really over. But one thing is certain. When you come out of the storm, you won't be the same person who walked in. That's what this storm's all about."

HARUKI MURAKAMI

CONTENTS

1 INTRODUCTION 1
- 1.1 Context 2
- 1.2 Problem Definition 3
  - 1.2.1 Research Questions 4
  - 1.2.2 Research Hypotheses 6
  - 1.2.3 Research Objectives 6
- 1.3 Contributions 7
- 1.4 Thesis Structure 8

2 TEMPORAL EXTRACTIVE TEXT SUMMARIZATION 11
- 2.1 Introduction 11
- 2.2 Approaches for Intermediate Representation of the Input 14
  - 2.2.1 Classical Approaches 15
  - 2.2.2 Time-biased Approaches 22
- 2.3 Approaches for Scoring Sentences & Selecting Summary Sentences 33
  - 2.3.1 Classical Approaches 34
  - 2.3.2 Time-biased Approaches 36
- 2.4 Evaluation 37
  - 2.4.1 Automatic Evaluation 38
  - 2.4.2 User Evaluation 42
- 2.5 Summary 43

3 SUMMARIZATION OF CHANGES USING LDA MODEL 45
- 3.1 Introduction 45
- 3.2 System Architecture 46
- 3.3 Sentence Ranking 50
  - 3.3.1 Approach-I: Baseline Temporal Sentence Score (BTSS) 50
  - 3.3.2 Approach-II: Temporal Sentence Score (TSS) 52
  - 3.3.3 Approach-III: Latent Topic Sentence Score (LTSS) using Latent Dirichlet Allocation Model 53
  - 3.3.4 Approach-IV: A Combination of Temporal Sentence Score (TSS) & Latent Topic Sentence Score (LTSS) 57
  - 3.3.5 Equality Measurement between Two Sentences 57
- 3.4 Experimental Setup 58
  - 3.4.1 Dataset Preparation 58
  - 3.4.2 Filtering Inserted Vandalism & Reverted Revisions 59
  - 3.4.3 Automatic Evaluation 60
  - 3.4.4 A Detailed Case Study 63
- 3.5 Results and Discussion 67
- 3.6 Summary 74

4 MULTI-LEVEL CHANGES SUMMARIZATION 75
- 4.1 Introduction 75
- 4.2 The MultiSummar System 77
  - 4.2.1 Extraction of Changes 78
  - 4.2.2 Detection of Candidate Categories of Changes 79
  - 4.2.3 Burst Detection 80

4.2.4 Identification of Significant Categories of Changes 85
4.2.5 Intermediate Summary Generation 86
4.2.6 Top Summary Generation 86
4.3 Evaluation Framework 86
4.3.1 ROUGE 87
4.3.2 Normalized Cosine Similarity 88
4.4 Experimental Setup 88
4.4.1 Validating K in the Context of Summarizing Changes 88
4.4.2 Finding the Topic Ratio Constant (λ_{th}) 90
4.4.3 Top Summary Evaluation 90
4.4.4 Intermediate Summary Evaluation 91
4.5 Results and Discussion 91
4.5.1 A Case Study 93
4.6 Summary 95

5 CONCLUSIONS 99
5.1 Extraction of Changes 99
5.2 Intermediate Representation of the Changes 99
5.3 Score Sentences & Select Summary Sentences 100
5.4 Evaluation 101
5.5 Future Research 101

LIST OF FIGURES

Figure 1 An overview which highlights the differences between the approaches used in Stage 1, the intermediate representation of the input for classical and time-biased summarization 15

Figure 2 An overview of the evaluation approaches for classical and time-biased summarization 38

Figure 3 System architecture 47

Figure 4 LDA plate diagram 56

Figure 5 WikiChanges system [NRD08] showing the number of revisions edited in monthly basis for the Wikipedia article on *Narendra Modi* 64

Figure 6 Effects on ROUGE-1 scores of the increasing number of revisions in the selected articles using different approaches 72

Figure 7 Effects on ROUGE-2 scores of the increasing number of revisions in the selected articles using different approaches 73

Figure 8 Boxplots on all ROUGE scores for different approaches on 54 different case studies 73

Figure 9 Hierarchical structure for multi-level changes summarization 77

Figure 10 Schematic diagram for multi-level changes summarization system (*MultiSummar*) 78

Figure 11 Model selection results showing the log-likelihood of the data for different values of K 80

Figure 12 Three different change categories ($K = 3$) are marked with colors red, green and blue for the Wikipedia article on *Narendra Modi* 81

Figure 13 Overlaying the curves for P_k (solid) and P'_k (dashed) for the three different change categories ($K = 3$) over the cosine similarity scores for each diff 83

Figure 14 A prototype example illustrating the detection of candidate burst segments 84

Figure 15 λ_{th} selection on 54 case studies for 49 different Wikipedia articles 90

LIST OF TABLES

Table 1 A synthetic comparison between classical and temporal text summarization 14

Table 2 Explanation of the symbols used in Equation 48 for both the systems 51

Table 3 The current (up to January 2015) complete revision history statistics for the selected 49 distinct Wikipedia articles 61

Table 4 Top 30 scoring terms obtained using different approaches on the Wikipedia article on *Narendra Modi* for the month of May, 2014 66

Table 5 Sentences selected by the summarization of changes system using different approaches on the Wikipedia article on *Narendra Modi* 68

Table 6 An example of a human-generated summary as reference summary on the Wikipedia article on *Narendra Modi* 69

Table 7 ROUGE scores using all the proposed approaches on the Wikipedia article on *Narendra Modi* for the month of May, 2014 69

Table 8 Number of edits made to the different Wikipedia articles for 54 selected time periods 70

Table 9 Overall ROUGE scores using all proposed approaches for 54 different case studies on 49 distinct Wikipedia articles 71

Table 10 Pairwise comparisons of **ROUGE-1** scores for all proposed approaches using Nemenyi post-hoc test. There are 54 different case studies on 49 distinct Wikipedia articles. 71

Table 11 Pairwise comparisons of **ROUGE-L** scores for all proposed approaches using Nemenyi post-hoc test. There are 54 different case studies on 49 distinct Wikipedia articles. 71

Table 12 Sentences selected for different time periods by the summarization of changes system from the Wikipedia article on *Steve Fossett* 72

Table 13 Statistics of expected K ($K^{(R)}$) and log-likelihood K ($K^{(L)}$) on 54 case studies for 49 distinct Wikipedia articles 89

Table 14 Statistics of ranked topics ($z_i^{(R)}$) which give the maximum ROUGE scores ($z_i^{(ROUGE_{max})}$) to the summaries for 54 case studies on 49 different Wikipedia articles 91

Table 15 Overall ROUGE scores using both default K and detected K value by the given short time ranges for 54 case studies 92

Table 16 Statistics for the use of diffs due to an effect of burst. The statistics are made on 54 case studies on 49 different Wikipedia articles. 92
Table 17 Frequency results of manual user evaluation through pairwise comparison. Tie indicates evaluations where two summaries are rated equal. 93
Table 18 Sentences are selected by different topic-ID's from the Wikipedia article on *Steve Fossett* between September, 2007 and October, 2008 by giving the default number of topics as 2 94
Table 19 Detected Burst details are shown for the Wikipedia article on *Steve Fossett* between September, 2007 and October, 2008 95
Table 20 Sentences are selected by different topic-ID's from the Wikipedia article on *Steve Fossett* between September, 2007 and October, 2008 by the automatic detection of number of topics as 4 96
Table 21 Intermediate summaries evaluation using BMM for the Wikipedia article on *Steve Fossett* between September, 2007 and October, 2008 97

LIST OF ALGORITHMS

1 Filtering Inserted Vandalism and Reverted Revisions 60

ACRONYMS

BMM	Best Match Mapping. 8, 98, 101
DUC	Document Understanding Conference. 3, 26, 41, 62, 88
IHSC	Incremental hierarchical sentence clustering. 27
IR	Information Retrieval. 1, 38, 45
IUS	Incremental update summarization. 27
LCS	Longest Common Subsequence. 41, 63, 88
LDA	Latent Dirichlet Allocation. 7, 8, 17, 18, 26, 30, 45, 48, 49, 53–57, 65, 66, 74, 75, 86, 96, 100–102
MACD	Moving Average Convergence/Divergence. 29, 30
MMR	Maximum Marginal Relevance. 19, 34, 35
MRSP	Manifold ranking with sink points. 27
MRW	Markov Random Walk. 20, 21
NLP	Natural Language Processing. 11, 28, 74
PNR^2	Positive and Negative Reinforcement. 27
QCQP	Quadratically constrained quadratic programming. 27
ROUGE	Recall-Oriented Understudy for Gisting Evaluation. 41, 42, 45, 50, 62, 63, 67, 69, 72, 74, 88, 89, 100
SVD	Singular value decomposition. 17, 40, 41
TAC	Text Analysis Conference. 3, 41, 62, 88

NOMENCLATURE

$[a_k]$ A masking array for the k-th category of changes using the piecewise polynomial P_k.

$[a'_k]$ A masking array for the k-th category of changes using the piecewise polynomial P'_k.

λ_j The topic ratio value of j-th topic.

λ_{th} The topic ratio constant.

$\mathcal{D}_k$ The valid diffs within the time range $\mathcal{T}_k$ for a burst region for the k-th category of changes.

$\mathcal{T}_k$ The time range for a burst region for the k-th category of changes.

$diff_t$ The t-th diff.

$diff_t^{(s)}$ The score of the t-th diff.

ρ The similarity score between $sentence_i$ and $sentence_j$.

$\mathbf{w}_i$ The i-th feature vector which consists of a sequence of words.

BOW A set of V distinct words.

$BTSS(sentence_i)$ The baseline temporal sentence score (BTSS) for the i-th sentence.

$BTTS(term_j)$ The baseline temporal term score (BTTS) for the j-th term.

$cos(F, F')$ The cosine similarity between the feature vectors for a topic and a block-diff.

CP_{ki} The center point of the grid R_{ki}.

CP'_{ki} The temporal center point of the grid R_{ki}.

$D_{block}^{(del)}$ The subset of differences, which occurred due to deletions, are extracted in paragraph basis.

$D_{block}^{(ins)}$ The subset of differences, which occurred due to insertions and modifications, are extracted in paragraph basis.

$D_{word}^{(del)}$ The subset of differences, which occurred due to deletions, are extracted in word basis.

$D_{word}^{(ins)}$ The subset of differences, which occurred due to insertions and modifications, are extracted in word basis.

$itf(term_{ij})$ The Inverse topic frequency (itf) for $term_{ij}$.

$LTSS(sentence_{ik})$ The latent topic sentence score (LTSS) for the k-th sentence, $sentence_k$ from the i-th topic.

$LTSS(sentence_k)$ The latent topic sentence score (LTSS) for the k-th sentence, $sentence_k$ regardless of the i-th topic.

$\mathrm{LTTS}(\mathrm{term}_{ij})$ The latent topic term score (LTTS) for term_{ij}.

$\text{n-cos}(S_i^{(R)}, S_j^{(A)})$ The normalized cosine similarity score between the i-th reference summary $S_i^{(R)}$ and j-th system-generated summary $S_j^{(A)}$.

$P^{(del)}(\mathrm{term}_j)$ The probability of term_j occurring as a result of deletions.

$P^{(ins)}(\mathrm{term}_j)$ The probability of term_j occurring as a result of insertions.

P_k The piecewise polynomial for representing the normalized grid scores for all points throughout the set of grids.

P'_k The piecewise polynomial for representing the temporal scores for all points throughout the set of grids.

R_k The subset of all grids for the k-th category of changes.

$R_k^{\prime(s)}$ The burst segment with the maximum score for the k-th category of changes.

R'_k A burst segment for the k-th category of changes.

R_{ki} The i-th grid from the subset of grids, R_k that contains the diffs of the k-th category.

$R_{ki}^{(s)}$ The score of the grid R_{ki}.

$\mathrm{sentence}_i$ The i-th sentence.

$\mathrm{sentence}_i^{(s)}$ The corresponding score of $\mathrm{sentence}_i$.

$\mathrm{sentence}_{ik}$ The k-th sentence, $\mathrm{sentence}_k$ from the i-th topic, $z_i = i$.

term_{ij} The j-th term, term_j belongs to the i-th topic, say $z_i = i$.

$\mathrm{term}_{ij}^{(s)}$ The corresponding score of term_{ij}.

$\mathrm{tf}(\mathrm{term}_j, D_{word}^{(del)})$ The frequency of j-th term (term_j) in the set $D_{word}^{(del)}$.

$\mathrm{tf}(\mathrm{term}_j, D_{word}^{(ins)})$ The frequency of j-th term (term_j) in the set $D_{word}^{(ins)}$.

$\mathrm{TSS}(\mathrm{sentence}_i)$ The temporal sentence score (TSS) for the i-th sentence.

$\mathrm{TTS}(\mathrm{term}_j)$ The temporal term score (TTS) for the j-th term.

z The label is to identify in which topic a sentence belongs.

$z_i^{(R)}$ A set of significant topics in ranked order.

block-diff The diff process extracts the changed paragraphs by comparing the consecutive document versions.

word-diff The diff process extracts the changed words by comparing the consecutive document versions.

1 INTRODUCTION

In the area of Information Retrieval (IR), the task of automatic text summarization usually assumes a static underlying collection of documents, disregarding the temporal dimension of each document. However, in real world settings, collections and individual documents rarely stay unchanged over time. The World Wide Web is a prime example of a collection where information changes both frequently and significantly over time, with documents being added, modified or just deleted at different times. Generally, a web page is considered as a dynamic document where the contents can be placed for an indefinite time. The changes in the contents of a web page can be of different sizes. Usually, either some new information is added or some old texts are deleted in addition to the rest of unchanged, static contents in the page. The ultimate case occurs when the whole page is deleted or a new one is created. Regarding the textual changes of the page, though they can have various types of changes, we assume that, to high extent, they are devoted to a common topic to the entire collection. Previous work addressing the summarization of web documents has frequently discarded the dynamic nature of the web, considering only the latest published version of each individual document. Thus, the inclusion of a temporal dimension in text summarization addresses a new challenge— the automatic summarization of changes in dynamic text collections. In other words, the idea is to collect the textual changes in the lifetime of a text collection in order to produce their summary considering a certain time interval.

Wikipedia is one of the major examples of a dynamic collection, with clearly identified documents—the articles, whose evolution in time can be seen in the revision history. When searching "Pete Seeger" on Wikipedia, for example, it is possible to access the history of collaborative editing for the corresponding article, as all revisions of this article are stored. In order to obtain the summary of changes of "Pete Seeger" within the time range "January 2014", we would expect to obtain "Pete died in New York City on January 27, 2014". Even though it is true that "he was an American folk singer and activist", this summary expresses a more general and more static information which, in a context of summarization of changes, does not pertain to the period "January, 2014". The following two examples clarify the distinction between inclusion and exclusion of the temporal period into the summary.

Example 1. *A summary that reflects only the changes made to "Pete Seeger" article for the given time range "January 2014".*

- *Pete died in New York City on January 27, 2014.*

Example 2. *A summary of "Pete Seeger" article without taking a temporal period into account.*

- *He was an American folk singer and activist.*

1.1 CONTEXT

Temporal summarization is a new and important variant of text summarization research. With the dramatical increase in online information from web, forum, weblog, collaborative documents, or even the collection of messages shared in a social network, the strong dynamic nature of the information becomes more and more evident. In this situation, how to summarize the most significant content becomes a challenging problem. The difficulty of constructing an appropriate model where the dynamic facet of document collections is taken into account for automatically summarizing the major changes is not fully recognized earlier.

Research has been conducted in the past on related tasks but a very few systems make explicit use of time, let alone produce summaries according to the dynamic content. The concept of *monitoring changes* [AGK01] has been proposed in 2001. This concept focuses on unstructured dynamic text collections such as news articles. The goal of *monitoring changes* is to keep users informed by extracting the main events from a stream of news stories on the same topic. Similarly, the problem of summarizing changes can be used in any unstructured dynamic text collections to generate a summary of changes devoted to a common topic within a given time period. Given a collection of document groups, *Comparative Extractive Document Summarization* (CDS) [WL12] was proposed to generate a summary of the differences among these document groups sharing a similar topic. If these document groups have evolved over time, the goal is to generalize the CDS problem, extending it to the evolution of the differences over time. The result will then be a summary of the differences among time-dependent comparable document groups for a selected time period. In this way, the summarization of changes can easily be adapted to other problems.

ChangeDetect [Cha] has been proposed to detect a list of changes made to a user-given web page, and to notify the users via email. This allows web users to track the important changes made to their favorite web pages. Because it is impossible to see every possible change, with this service users can pay attention to the details only when a summary of significant changes triggers enough interest. A similar situation can occur in enterprise and public environments, where information is always being updated in the existing shared repositories. A summary of changes will make people aware of the changes in a concise form, either on a daily or weekly basis. ChangeSummarizer [JBI04] periodically monitors a web collection to look for new textual changes and present them in a condensed form. However, unlike to the problem defined here, the information needs are specified as "recent, important changes". In this thesis, we take into consideration a broader view for defining a time period. We assume that the changes are not only limited to recent changes but can address any user defined period. Nunes et al. [NRD08] proposed WikiChanges, a web-based application designed to plot the distribution of the revisions made to an article over time. They introduced the *revisions summarization* task that addresses the need to understand what the revisions made. In this work, we intend to investigate new algorithms for summarization of changes in dynamic collections. Wikipedia Event Reporter [GKKNS13] extracts the event-related updates automatically by analyzing the entire history of updates for a given entity. This system combines two tasks, *event detection* and *temporal summarization* using Wikipedia revision history as a source of data.

On the other hand, query-oriented update summarization [WFQY08] poses new challenges to the sentence-ranking algorithms as it requires not only important and query-relevant information to be identified, but also novelty to be captured when document collections evolve. There are two main differences between this task and the task of summarization of changes. First, the information needs addressed in query-oriented update summarization are restricted to current and novel updates whereas the summarization of changes is not limited to current changes and should address any user-defined period. Second, the assumption in query-oriented update summarization is that the user is already familiar with the past information related to the topics. In contrast, in summarization of changes there is no explicit prior assumption, and the user may or may not be familiar with the topic.

Another evidence of the importance of the inclusion of time in summarization is the organization of contests and workshops focusing on temporality. The Text REtrieval Conference (TREC) [Treb], co-sponsored by the National Institute of Standards and Technology (NIST) has started the *temporal summarization* track [Trea] in 2013 to develop systems that allow users to efficiently monitor the information associated with an event over time. The Document Understanding Conference (DUC) & Text Analysis Conference (TAC) launched *update summarization* as a pilot task in 2007 [Usu]. This task focuses on generating an update summary for multiple documents based on a common topic under the assumption that the user is familiar with a set of past documents. The objective of NTCIR Temporal Information Access (Temporalia) task [Ntc] is to foster research in temporal information access. Besides this, the workshops WWW Temporal Web Analytics workshop (TWAW) [Tem] and the SIGIR Time-Aware Information Access workshop (TAIA) [Tai] open up an entirely new range of challenges and possibilities by accounting the temporal dimension.

Based on these factors, it is expected that there will be an upsurge of applications that can handle the dynamically changing information by building adequate models. This emerging research can be applied in several scenarios, one of them being search. When queries include temporal information, summarization can provide more focused snippets for search results [Cam13]. In enterprise and public environments, users frequently modify information in shared repositories; a summary can make them aware of the changes in a concise form, for instance on a daily or weekly basis. A summary of changes can also be very useful for a journalist or a student exploring historical information that is no longer available in the current version of the documents on a specific topic. The summarization of changes can also play an important role in online social networks. On Twitter or on Facebook, people often comment on events in real time by posting messages (tweets) or updating status publicly and instantly. Similarly, in blogs people express their views or opinions on a particular topic. From the collection of the tweets/status updates/blog posts on a specific topic within a time frame, a summary of changes can provide an overview of the significant alterations made to the topic during that period.

1.2 PROBLEM DEFINITION

When addressing the problem of summarizing changes in dynamic text collection, we are interested in producing an automatic summary that describes

the alterations that were made to a series of versions of a document or set of documents on a similar topic for a given time period. In other words, the idea is to have a summary of changes in the lifetime of a text collection for a temporal period. The text collection may include either a series of revisions to a document or multiple documents sharing the same topic.

The following three properties are expected from a summary of changes:

- Time-dependency. The summary is expected to highlight the information that has been changed on the set of documents between two points in time. Moreover, it should also exclude the static information existing in the documents.

- Significance. During a given time period, changes to the text take place for different reasons. Often, changes are not very significant, such as the correction of syntax or grammatical errors, the modification of links or changes regarding a past time period. This outdated information seems to be updated simultaneously when an event draws attention to a Wikipedia article. However, these updates do not focus on the reason for those changes within that particular time period. Hence, the challenge is to identify the meaningful information which has the potential to be a significant part of the summary, besides all other unnecessary changed texts for the given period. Irrelevant details do not belong in the summary.

- Non-redundancy. The summary is expected to be synthetic and therefore avoid redundant information. Two similar sentences carrying the same information should not be selected simultaneously.

1.2.1 Research Questions

The main research question of this thesis is the development of different approaches to produce the significant changes that have occurred in a collection of documents between two dates, as a temporal summary. We are particularly interested to identify the changes which have the potential to be the main reasons for the updates, besides all other unnecessary changed texts for the given period. This has induced an important challenge for summarizing changes in highly active contexts. To approach our problem in a detailed way we take the problem into consideration by setting up the set of questions below. We divide them into questions regarding extraction of changes, intermediate representation of the changes, score sentences & select summary sentences, and summary evaluation.

Extraction of changes

The extraction of changes from the text collection plays an important role in finding novel information. First, we should ensure that the information extracted from the dynamic collection includes only the changes, leaving out the static information throughout the evolution of the collection. In practice, the changes in a web page can be made at different times in three ways: insertion, modification or deletion. These changes are accompanied by valuable metadata including the timestamp. The timestamp for any extracted change can be used in order to check whether its time period lies within the given temporal period. Based on this, we formulate our first research question:

- **Q1:** Can we extract information from text such that it includes only the changes made to the text collection within a specified time period?

Intermediate representation of the changes

The first task in extractive text summarization is to derive some intermediate representation of the input which captures only the key aspects of the text. This representation will help further to identify the important contents among all. The inclusion of temporality in summarization brings the same challenge as classical summarization to identify the meaningful information which has the potential to be a significant part of the summary. During a given time period, changes to the text take place for different reasons. It is discussed earlier that the relevant details should obtain more weight compared to all other unnecessary changed texts for the given period. This brings a new challenge to the sentence-ranking algorithm as it requires not only the novel information to be identified, but also the identified information needs to be significant. This is an important challenge for summarizing changes in highly active contexts. Besides this challenge, for a given time interval, if there are multiple categories of changes, they are likely to be distinguished. The number of different categories of changes that were made throughout the evolution of the collection needs to be estimated. Thus, we formulate our next research question:

- **Q2:** Is it possible to fit a model that can derive an intermediate representation capturing the key aspects of the extracted information?

Score sentences & select summary sentences

Once an intermediate representation of the extracted information has been derived, each of the sentences from the text collection is assigned with a score indicating its importance. To identify important content, the weight of each sentence is determined based on that intermediate representation. Finally, the best combination of significant sentences is selected to form a summary. One of the properties of the summary of changes is to be synthetic and therefore sentences that are similar to already chosen sentences should not be selected. For the selection of the sentences in the summary, other factors also come into play. For example, the maximum potential regions of updates for different types of changes can influence the approaches used to select the sentences in the summary. Another factor which can affect sentence ranking is the assessment of the importance of different types of changes.

The representation of multiple summaries for different types of changes in order to provide information to the users depending on their requirements leads us to consider a multi-level approach. One of the possible ways of representation is to present the summaries hierarchically: at an intermediate level, a separate summary for each significant category of changes can be generated and at the top-level, a single summary consolidates the most significant changes. It is worth to mention that each intermediate summary might contain more detailed information for a particular category of changes. Here, we address the third and fourth research questions which are:

- **Q3:** Does the measurement for scoring a sentence identify the significant changes?

- **Q4:** Do other factors help in determining if summary sentences are focused (sentences should only contain information that is related to the rest of the summary) and coherent (consistency among sentences)?

Evaluation

It is a common practice to evaluate a system generated summary against a reference summary using an evaluation metric. But, a framework needs to be built that can deal with multiple summaries generated from different topics. This is usually a difficult task as, in practice, a one to one mapping between the reference and system generated summaries is not provided a *priori*. Moreover, there can be a situation where the number of reference and system generated summaries are not the same. Thus, our fifth research question is:

- **Q5:** Is it possible to build an evaluation framework that evaluates multiple summaries generated from different topics?

1.2.2 Research Hypotheses

Keeping in mind the questions posed, we will now define the research hypotheses:

- **H1.** Emphasizing only the changed information while filtering out the static parts during information extraction makes it easier to identify the key aspects of the extracted information within the given temporal period.
- **H2.** The finding of different clusters in which each cluster is likely to reflect one significant change whereas different clusters represent different changes, improves the identification of the significant changes.
- **H3.** The sub-changes related to any significant change tend to co-occur within a temporal proximity and thus, if it is possible to determine the potential region of updates for any kind of changes, the summary is likely to be more focused and coherent.

1.2.3 Research Objectives

We start by studying how to incorporate the temporal dimension of documents when a user gives a time interval in the lifetime of a text collection. Then, we explore deriving an intermediate representation with the temporal features of the input which captures only the key aspects of the text. This step is of the utmost importance as it investigates the temporal features that are used to identify the important changes discussed in the given temporal period. Different approaches are therefore investigated for converting the text to an intermediate representation.

Next, we show a way to determine the number of different categories of changes that were made throughout an active collections of documents for a given time interval. The number of estimated latent categories is considered as a set of candidates which is likely to be comprised of both significant and non-significant ones. For each category, a burst region, where a succession of changes for that category occurred within a short enough period of time, is identified. This would help to filter out the non-significant categories from the set of candidates. The sub-changes surrounding a category of changes tend to co-occur within a temporal proximity. Following this, we assess whether this intuition achieves coherence among the selected sentences in order to produce a summary.

The objectives of this thesis allow users to answer questions such as:

- What is the single consolidated summary that reflects the most significant categories of changes that were made to a document between two specific revisions or in a collection of documents?
- How many candidate categories of changes were made throughout the revisions of a document given a time interval and which categories are significant among this set of candidate categories?
- Which are the summaries containing more detailed information for each significant category of changes?

1.3 CONTRIBUTIONS

Our research extracts the changes from the dynamic text collection for a given temporal period and investigates how this information can be used to present the important changes as in a summary. In what follows, we point out a list of contributions with reference to the research questions as previously discussed and indicate the chapters where further details can be found.

- **C1:** We provide a way to extract the temporal features from a set of documents for a given temporal period. The time period actually determines how many versions of the document are used for the extraction of the temporal features. These temporal features take into account only the changes while filtering out the static parts from the collection of documents.

  [*Related to Q1, which will be further discussed in Chapter 3*]

- **C2:** We propose four different approaches to generate summaries using extractive summarization techniques. The first approach provides a baseline that is adapted from previous related work, which periodically monitors a web collection in search for recent changes and generates their summary with respect to a specific topic. In the second approach, the temporal aspect of each term is investigated by considering the joint probabilities of both insertion and deletion events over a set of document versions within the given period. The third approach is based on Latent Dirichlet Allocation (LDA) model for finding hidden/latent topic structures of changes. The fourth approach is a combination of the previous two approaches. In addition, we show a simple similarity measurement to address the non-redundancy requirement in summary.

  [*Related to Q2 & Q3, which will be further discussed in Chapter 3*]

- **C3:** We propose a burst region detection algorithm that identifies a potential region for each categories of changes. Unlike conventional approaches for burst detection, the proposed algorithm is focused only on changes of a similar category instead of detecting burst by considering all categories of changes altogether.

  [*Related to Q4, which will be further discussed in Chapter 4*]

- **C4:** We estimate the number of different categories of changes that were made throughout the evolution of the collection via model selection criteria using Bayesian statistics. We assess the importance of each

category of changes by analyzing its burst region, as well as the topic ratios, in order to filter out the non-significant ones. We define the propositions in order to determine whether a category is significant or not. A higher topic proportion probability value indicates a more significant category. We show a way to produce more focused and coherent summaries by selecting the sentences from the burst region for each category of changes. The sub-changes related to any significant change also tend to co-occur within a temporal proximity. This hypothesis is explored.

[*Related to Q4, which will be further discussed in Chapter 4*]

- **C5:** We describe a multi-level changes summarization framework that can automatically detect multiple significant changes in hierarchical levels within a user-defined time period. At the top-level, a single summary is produced that consolidates the most significant changes, whereas each intermediate summary contains the changes for every significant category in detail. This multi-level framework facilitates the exploration of information at different levels so that a user can use them on the basis of their interest (i.e more generic or more specific).

  [*Related to Q3, which will be further discussed in Chapter 4*]

- **C6:** Finally, we propose an evaluation framework by taking into consideration the time periods where exactly one significant change has occurred within the given range. Later by removing this constraint, we build another evaluation framework that can deal with multiple summaries generated from different categories of changes for a wider temporal period. An automatic mapping technique called Best Match Mapping (BMM) is proposed for this purpose.

  [*Related to Q5, which will be further discussed in both Chapter 3 & Chapter 4*]

1.4 THESIS STRUCTURE

In this thesis, the objectives stated above are addressed by proposing different algorithms. For every objective, we present the challenges associated with it and describe the set of experiments undertaken. The remainder of this thesis is structured as follows.

- **Chapter 2: Temporal Extractive Text Summarization** presents an extensive survey of the most prominent recent extractive approaches in order to understand the distinction between classical text summarization & temporal text summarization techniques. In other words, we have outlined the connection to the approaches used in classical text summarization and have contrasted the approaches addressed in temporal text summarization in terms of how they represent the input, score sentences, select the summary and the evaluation framework.

- **Chapter 3: Summarization of Changes using Latent Dirichlet Allocation Model** introduces a new framework for summarizing changes of document modifications in dynamic text collections. Four different approaches are proposed to estimate the term scores, and then to rank the sentences based on those scores. A system based on Latent Dirichlet Allocation model is used to find the hidden topic structures

of changes. The purpose of using the LDA model is to identify separate topics where the changed terms from each topic are likely to carry at least one significant change. The different approaches are then compared with the previous work in this area with a proposed evaluation framework. Statistical tests are performed to assess the significance of the proposed solutions.

- **Chapter 4: Multi-level Changes Summarization** focuses on to estimation of the number of latent topics within a given time period. For each detected topic, a burst region is identified. The importance of each topic is assessed by analyzing its burst region, as well as the topic ratios. A set of sentences is then selected from the burst region to present a meaningful and coherent summary, for each topic. These summaries are generated hierarchically: a separate summary is presented for each significant topic in an intermediate level, and at the top-level a single summary is generated in order to consolidate the most significant changes.

- **Chapter 5: Conclusions and Future Research** discusses the answers to the research questions raised in this study. We also highlight several issues that can provide future research within the scope of temporal summarization.

2 TEMPORAL EXTRACTIVE TEXT SUMMARIZATION

Automatic text summarization presents a concise and fluent summary to the user conveying the key information from a single (single-document summarization) or a set of documents (multi-document summarization). It helps the users to quickly find the specific information they are looking for within documents. For example, a number of news services (e.g. Google News) have been developed to group news articles into different topics, and then produce a short summary for each topic. The users can easily look into their topics of interest, checking these short summaries for further details. In general, there are two approaches followed for automatic text summarization: extractive summarization and abstractive summarization [JM00; KM02]. Extractive summarization selects a set of important sentences from the original documents. In contrast, abstractive summarization builds an internal semantic representation, and then uses Natural Language Processing (NLP) techniques to create a summary that is likely to be closer to what a human might generate. We intend to lay stress upon the techniques used in extractive summarization. The choice to focus on extractive techniques excludes the exploration of the approaches developed for abstractive summarization, but allows us to point out commonalities and differences among the most dominant extractive approaches through different stages.

Publication

This chapter is partly based on the following publication:

- Kar, M. (2013). Summarization of Changes in Dynamic Text Collections. In Proceedings of the 5th BCS IRSG Symposium on Future Directions in Information Access (FDIA 2013) (pp. 14–19). [Kar13]

2.1 INTRODUCTION

With the unprecedented growth of the web in recent years, a massive amount of documents is now continuously being published, and most of this information is strongly time-dependent. In this respect, time has been gaining an increasing importance within different subareas of Information Retrieval [ASBYG11; CDJJ14; KBg15]. The inclusion of a temporal dimension can play an important role in automatic text summarization, namely classical text summarization. Previous work addressing this area usually assumes either a static underlying collection of documents or only the latest published version of each individual document, disregarding the temporal dimension of each of them. Thus, a compelling research interest is going on towards the time-biased summarization domain, namely temporal summarization which gives the birth to new requirements and scenarios that has led to relevant research trends and avenues.

In particular, temporal summarization as another stride towards summarization is a time-biased multi-constrained summarization in dynamic text

collections. It is not only concerned with choosing the salient information and removing redundancy in the presented summary, but also concerned with the dynamic evolution of the document collection to capture the important changes over time. This research is now beyond extracting only the temporal expressions from text or searching and normalizing references to dates, times and elapsed times [MW00; MZ05]. Rather, this research has its roots in different relevant areas, such as update summarization, timeline summarization, new event detection within topic detection and tracking, and summarization of changes.

One of the main challenges in classical text summarization (such as single or multi-document summaries, generic or query-focused, etc.) is that a single or a cluster of documents might contain diverse information, which is either related or unrelated to the main topic, and hence there is a need to analyze the information which is globally important to reflect the main topic. Nenkova and McKeown [NM12] pointed out three relatively independent stages performed by virtually all the extractive text summarizer systems:

- **Stage 1:** Create an intermediate representation from the input text which captures only the major aspects.
- **Stage 2:** Score sentences based on that representation.
- **Stage 3:** Select a summary consisting of several sentences.

The first stage derives some intermediate representation of the text and based on this representation, it will help further to identify the important content. Once the main themes of a document have been identified from the intermediate representation, the second stage is needed in order to distinguish between relevant and irrelevant information. Each sentence is assigned a score which indicates its importance. The final stage addresses the generation of a summary by selecting the best combination of important sentences previously identified. Incorporating changes in the approaches that are associated with any stage can markedly affect the performance of the summarizer, and we will discuss these specific changes made in the classical summarization methods for the purpose of including the temporal aspects.

Good surveys on previous research have already been published [Jon07; NM12], describing many of the systems and techniques that have been developed since the beginning of classical text summarization. On the other hand, a general review of the state-of-the-art is provided, emphasizing recent summary types [LP12]. To the best of our knowledge, this is the first comprehensive review of the state-of-the-art to provide a more scrutinizing overview that discusses the approaches which follow the same hierarchy of stages in classical text summarization and at the same time, to highlight the similarities and the differences of the approaches addressed for temporal summarization. The analysis of these comparisons in both approaches allows us to have a wide and useful background on the main important aspects in this research field.

Table 1 depicts the most common factors distinguishing between classical and temporal text summarization. The birth of web has encouraged new types of textual genres along with the feature of versioning or non-versioning, and containing various degrees of changes in text. This allows the emergence of novel dynamic text collections, such as Wikipedia, Twitter streams, blogs, etc. which have gained an increased attention in contrast to the traditional datasets based on newswire or scientific documents.

In addition, different kinds of summaries, such as *update* summaries, *query-oriented update* summaries, *timeline* summaries, *summaries of changes* and *temporal snippets* are stressing upon the fact that we now have vast amounts of information are evolving with respect to time. The time dimension comes into play when not only the current document is observed but also the previous documents are taken into account during the feature extraction process. *Update* summaries [DA12] attempt to generate summaries for multiple documents based on a common topic under the assumption that the user is familiar with a set of past documents. *Query-oriented update* summaries [WFQY08] are similar to *update* summaries but besides that, it poses query-relevant information to be identified. *Timeline* summaries [CL04; YWOKLZ11], which organizes events by date on news articles is a special case of multi-document summaries. This kind of summaries allows users to have an quick overview of events and fast news browsing relating to their interest from a collection of various news sources. *Summaries of changes* [LPT00; JBI04] basically mean summarizing the changes observed in dynamic text collections over specific period of time. *Temporal snippets*, called TSnippet [ABYG09; AGBY11] is introduced as document surrogates for document retrieval and exploration.

The classical summarization types [LP12], such as generic or query-focused, indicative or informative, personalised or sentiment-based do not explicitly focus on the impact of time on the evolution of information. The *generic* summaries can serve as substitute of the original text as they try to represent all relevant facts of a source text. Whereas, the content of a *query-focused* summary is compelled by a user query. *Indicative* summaries are used to point out what topics are addressed in the source text. On the other hand, *informative* summaries are intended to cover the topics with more detailed information from the input text. The purpose of *personalised* summaries [AKDZ05; BBZ08] is to provide the specific information that matches with the corresponding user profile. *Sentiment-based* summaries [HCGL08; BGP08] present a coherent text considering the sentiment a person has towards a topic, product, place and service. It is clear from the definition of these classical summary types that they do not exploit the use of time. Next, we review the properties which are expected from a classical and time-biased summary. The main difference between them is that a time-biased summary is expected to highlight the novel information within time and it should exclude the static information existing in the documents. We identify several related tasks (e.g., new event detection, burst detection, novelty detection and important date selection) that are required in addition to meet the needs of finding novel contents while deriving an intermediate representation from the input text. Different tasks may have different purposes but the common ground is utilizing time. In the following sections, we discuss several time-biased extractive summarization approaches within each identified task, depending on the nature of the purposes they have employed.

The remainder of this chapter is organized as follows. In Section 2.2 we give a broad overview of existing approaches for intermediate representation of the text and have contrasted approaches in terms of how the first stage is derived, while producing an extractive summary over time. Section 2.3 is devoted to reviewing the approaches used for the second stage from both classical and temporal perspectives. Besides this, different approaches considering both cases for the final stage, where the selection of the most important sentences will be analyzed, is also reflected in this section. Section

Table 1: A synthetic comparison between classical and temporal text summarization

	Classical Summarization	**Temporal summarization**
Document category	Static	Dynamic
Document version	Non-versioning (or, if versioning then the most recent version)	Versioning
Document collection	News archives Scientific documents Legal documents The most recent version of a website	Wikipedia Social media (Facebook, Twitter, blogs etc.) Emails Web archives A stream of news stories Multiple versions of a website
Summary types	Generic Query-oriented Indicative Informative Personalised Sentiment-based	Update Query-oriented update Timeline Change
Properties are expected from a summary	Significance Non-redundancy Coherence Focused	Time-dependency Significance Non-redundancy Coherence Focused
Deriving an intermediate representation for Stage 1	The representation takes into consideration the needs of finding important content	The representation takes into consideration the needs of finding both important and novel content

2.4 describes existing evaluation frameworks and metrics, as well as new proposed frameworks that have emerged concerning the automatic evaluation of time-biased summaries.

2.2 APPROACHES FOR INTERMEDIATE REPRESENTATION OF THE INPUT

We have attempted to give a comprehensive overview that presents the classical approaches of how they represent the input. At the same time, it highlights the contrasts with the time-biased approaches. We discuss several classical extractive approaches. Regarding time-biased approaches, different explicit tasks are identified at first, then within every task we discuss different approaches which distinguish these new ones from the aforementioned classical approaches. Figure 1 represents an overview of the approaches for intermediate representation of the input that emphasizes the differences between the Stage 1 in classical and time-biased summarization. This stage, specially pay an extra attention for finding novel information in case of time-biased summarization.

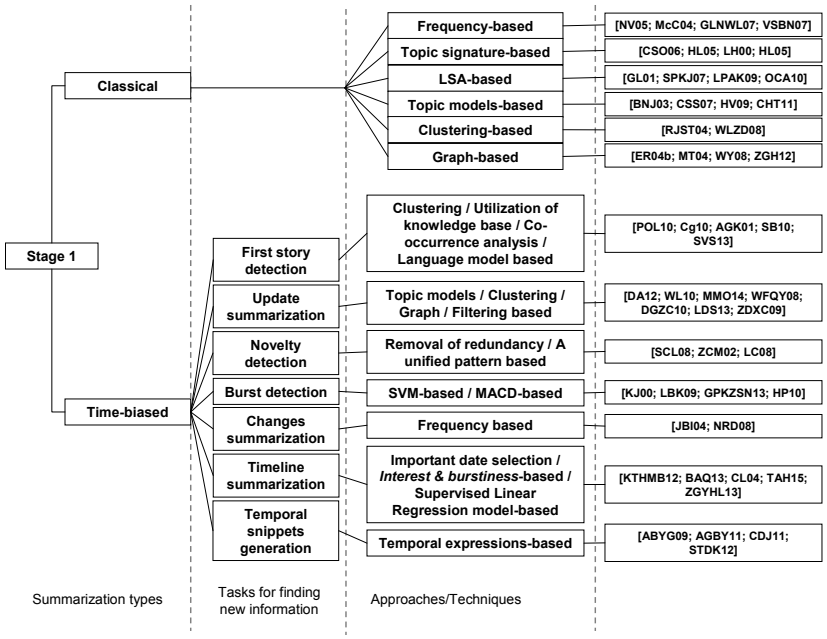

Figure 1: An overview which highlights the differences between the approaches used in Stage 1, the intermediate representation of the input for classical and time-biased summarization

2.2.1 Classical Approaches

We present some of the most widely applied classical approaches which are: *frequency-based*, *topic signature-based*, *Latent Semantic Analysis-based*, *Bayesian topic models-based*, *clustering-based* and *graph-based*.

Frequency Based Works

One of the remarkable works based on the frequency of word occurrences started in the late 1950's with H.P. Luhn's classic work [Luh58]. The justification of measuring word significance by using its frequency stands on the fact that the repetition of word occurrences emphasizes as an indicator of word significance. For finding the list of significant words, the words are arranged in descending order according to their frequencies after removal of common words.

SUMBASIC [NV05] is the system which is motivated by the following observation: "words occurring frequently in the document cluster occur with higher probability in the human summaries than words occurring less frequently." Basically, it computes the probability of each word, $p(w_i)$ appeared in the input as:

$$p(w_i) = \frac{n}{N} \tag{1}$$

where n is the number of occurrences of a word and N is the number of all words in the input.

SUMBASIC [NV05] only uses the frequency information for scoring the words but in contrast to SUMBASIC, Yi et al. [YGVS07] identified the positional information of words as a key additional feature along with the

frequency information. Each word position is computed according to its relative position in a document cluster, e.g. 0 is for the first word and 1 is for the last; then for each word, the average of its occurrences throughout the document cluster is calculated. A scoring function for each term was proposed to combine both frequency and position, using two different approaches: *generative* and *discriminative*. In *generative* approach, a term is selected at random from the document cluster, with a probability proportional to the overall frequency of the word in the cluster or the frequency at the beginning of documents. This term is added to the summary and then all occurrences of the term are deleted from the cluster. This process is repeated until a summary with the required length is generated. Whereas in *discriminative* approach, the probability of a term is learnt. For each content word in a document cluster, a label is assigned to 1, if it appears in the given human summary; otherwise, the label is 0. Then, the probability of the term that has label 1 is learnt using its features.

The term frequency/inverse document frequency (tf*idf) weighing has been employed to score the terms [McC04; GLNWL07], where the terms in a document are considered as important only if they are not very frequent in the whole collection. As it is claimed in the work of Filatova and Hatzivassiloglou [FH04], this tf*idf weighing may not be always sufficient for building high-quality summaries, and other types of features, for instance named entities or frequent nouns can be extracted and utilized for summarization.

There has been research in topic-focused multi-document summarization [VSBN07], where automatic summaries are generated in response to a given topic. In order to incorporate topic constraints in multi-document summarization, the system, SUMFOCUS [VSBN07] computes the weight for each word as a linear combination of the unigram probabilities derived from the topic description and the unigram probabilities from the document:

$$WordWeight = (1 - \lambda) * DocWeight + \lambda * TopicWeight \tag{2}$$

where $\lambda \in [0, 1]$ is a constant providing the flexibility to decide how much proportions *DocWeight* and *TopicWeight* need to be considered. Moreover, Vanderwende et al. [VSBN07] explored the sentence simplification (also known as *sentence shortening* or *sentence compression*) as a means to produce more content to the user, given a limit of summary length.

Topic Signature Based Works

The use of words corresponding to topic signatures, as a representation of the input has led to improvements in selecting important content for multi-document summarization of newswire texts [CSO06; HL05]. As first proposed in Lin and Hovy's work [LH00], the topic of a document (or a set of documents) can be represented using a set of terms that are highly correlated with a target concept. This is known as "topic signature". A topic signature is defined as a family of related terms, as follows:

$$\begin{aligned} TS &= \{topic, signature\} \\ &= \{topic, < (t_1, w_1), \ldots, (t_n, w_n) >\} \end{aligned} \tag{3}$$

where *topic* is the target concept and *signature* is a vector of related terms. Each t_i is a term highly correlated to *topic* with weight w_i. The number of related terms n can be decided depending on a cut-off associated weight. On the assumption that semantically related terms tend to co-occur, Lin and

Hovy [LH00] constructed topic signatures from *likelihood ratio* λ [Dun93]. To find the topic terms, a set of documents is classified into a) topic relevant texts $\mathcal{R}$, and b) topic non-relevant texts $\bar{\mathcal{R}}$.

The $-2\log\lambda$ value is computed for each term and then rank the terms according to their $-2\log\lambda$ value. A confidence level is selected from the χ^2 distribution table for a specific $-2\log\lambda$ value to determine the number of terms to be included in the topic signature. Harabagiu and Lacatusu [Har04; HL05] proposed an extension to this topic signature representation by considering the relations between topic signature terms and introduced a new representation of topics based on topic themes.

Latent Semantic Analysis Based Works

Latent semantic analysis (LSA) [DDLFH90] is an unsupervised technique of identifying important topics in documents by finding their underlying latent semantic structure. Gong and Liu [GL01] have used LSA for both single and multi-document summarization of news.

In LSA, each document is represented by a n by m matrix A, in which each row corresponds to a word and each column corresponds to a sentence i.e. $A = [a_{ij}]_{n\times m}$. If the sentence does not have the word, the weight is zero, otherwise the weight is equal to the tf * idf weight of the word. Singular value decomposition (SVD) from linear algebra is applied to the matrix A to make it as the product of three matrices: $A = U \sum V^T$. U is a n by m matrix of real numbers, in which each column can be interpreted as a topic i.e. some combination of words and each row gives the weight of each words. $\sum$ is a diagonal m by m matrix. Matrix V^T is a new interpretation of the sentences, each of which is not a combination of words that occur in the sentence but rather the words that are in terms of the topics given in U. The matrix $D = \sum V^T$ is a combination of the topic weights and the sentence representation to point out to what extent the sentence conveys the topic, with d_{ij} indicating the weight for topic i in sentence j.

The main drawback of the proposed summarization algorithm by Gong and Liu [GL01] is that it simply chooses the most important sentence for each 'topic' by assuming all topics are equally important. As a result, a summary may include some sentences, which are not really important. Instead of choosing one sentence from each topic, Steinberger et al. [SPKJ07] chose the sentences with the highest combined weights across all topics. Other improvements of the LSA approach [YKYM05; HMR06; OCA10] have been further explored.

The LSA-related methods represent a sentence by using a linear combination of semantic features. However, the obtained results are less meaningful, as LSA-related methods of document summarization may fail to extract meaningful sentences [LPAK09]. Lee et al. [LPAK09] proposed a new unsupervised generic document summarization method using Non-negative Matrix Factorization (NMF).

Bayesian Topic Models Based Works

Bayesian models have successfully been applied to *multi-document* and *query-focused* summarization to capture both general and specific information from documents in a probabilistic way that many other techniques lack [CSS07; HV09; MLM07; TYC09].

The basic idea in the LDA model [BNJ03] is that documents are represented as random mixtures over latent topics, where each topic is character-

ized by a distribution over words. There are D documents and document d has N_d words. It is well known that the Beta distribution is the conjugate prior of the Bernoulli distribution and the Dirichlet distribution is the conjugate prior of the multinomial distribution. In Bayesian probability theory, if the posterior distributions are in the same family as the prior probability distribution, the prior is called a conjugate prior for the likelihood function. α and β are the hyper-parameters of symmetric Dirichlet priors for the D document-topic multinomial distribution with parameter θ and the T topic-word multinomial distribution with parameter ϕ. In the generative model, for each document d, the N_d words are generated by drawing a topic t from the document-topic distribution $p(z|\theta_d)$ and then drawing a word w from the topic-word distribution $p(w|z = t, \phi_t)$.

The Special words with background (SWB) model [CSS07] based on latent topics is similar to the LDA model but in addition the multinomial distribution ψ is used to handle special words and the multinomial distribution Ω is to handle background words (with symmetric Dirichlet priors parameterized by β_1 and β_2). So, the main difference of the SWB model over the LDA model is that the SWB model can draw a word in three different ways: via topics, via a special word distribution or via a background distribution. Therefore, the conditional probability of a word w given a document d can be written as follows:

$$\begin{aligned} p(w|d) = p(x = 0|d) \sum_{t=1}^{T} p(w|z = t)p(z = t|d) \\ + p(x = 1|d)p'(w|d) + p(x = 2|d)p''(w) \end{aligned} \tag{4}$$

where x is a latent random variable and can take values $x = 0$ if the word w is generated via the topic route, $x = 1$ if the word w is generated as a special word for that document and $x = 2$ if the word is generated from a background distribution specific for the corpus, $p'(w|d)$ is the special word distribution for document d, and $p''(w)$ is the background word distribution for the corpus.

BAYESUM [DM06] and TOPICSUM [HV09] are also very similar to the SWB model. All these models are learnt to discriminate between the collection and the document-specific distributions in order to capture the major pieces of information in a collection of documents. This distinction helps directly to identify the important pieces of information in the context of summarization.

HIERSUM [HV09] uses a hierarchical Bayesian approach [GT04] to represent content specificity in a hierarchical way. Barzilay and Lee [BL04] observed that the document collection can have several topical themes with their own specific vocabulary. A user might have interest either in general content of a document collection or one or more of the sub-stories. HIERSUM adapts this kind of flexibility to produce multiple 'topical summaries' in order to ease content discovery and navigation. As in SWB [CSS07] or TOPICSUM [HV09] models, a word can be drawn in three different ways: background, document-specific and content. But in HIERSUM [HV09], instead of a single content distribution for a document collection, a general content distribution and as well as a specific content distribution for each specific topic are drawn. The intention is that the general content distribution prefers words which appear both in many documents and consistently throughout a document. But, each specific content distribution chooses the

words which are used in several documents but tend to be used in concentrated positions in a document.

Celikyilmaz and Hakkani-Tür [CHT10; CHT11] presented a hybrid model based on a discovery of hierarchical topics for generating summaries with higher linguistic quality in terms of coherence, readability, and redundancy. The Two-tiered topic model (TTM) [CHT11], which is an extension of the hierarchical topic model [GT04] identifies salient sentences by discovering hierarchical concepts from documents. In the TTM model, each word in each document is associated with three random variables: a sentence S, a higher-level topic H, and a lower-level topic T, where the higher-level topics are multinomial over sub-topics at lower levels. Instead of representing sentences as a layer in hierarchical models [CHT10], the TTM model can represent correlations from the lower-level topics given sentences. The aim is to eliminate the redundant sentences in a summary by discovering these correlations from lower-level topics.

Clustering Based Works

Radev et al. [RJST04] presented a multi-document summarizer, MEAD, which generates summaries using cluster centroids. A centroid is a set of words that are statistically important to a cluster of documents. The authors have used three features to compute the salience of a sentence: centroid value, positional value, and first-sentence overlap.

Although clustering techniques were already being used [MR95; BME99] for identification of themes/events, Radev et al. [RJST04] are the pioneers of exploiting cluster centroids in summarization. Several web-based news clustering systems are, for example, *Google News*[1], *Columbia News Blaster*[2] or *News In Essence*[3].
The first step in this approach is to make clusters of relative documents. To accomplish this task, an agglomerative clustering algorithm is used on the documents presented as a vector of weighted terms (e.g. tf*idf), successively adding documents to clusters and recomputing the centroids according as follows [RHM99]:

$$c_j = \frac{\sum_{d \in C_j} \tilde{d}}{|C_j|} \tag{5}$$

where c_j is the centroid of the j-th cluster, C_j is the set of documents of the j-th cluster, its cardinality being $|C_j|$ and $\tilde{d}$ is a truncated version of d that removes lower weighted words below a threshold. Centroids are thus defined as a set of words that are statistically important to a cluster of documents. In the next step, the centroids are used to identify sentences from each cluster that are central to the topic of the entire cluster. Two metrics are defined in Radev et al.'s work: Cluster-based Relative Utility (CBRU) and Cross-sentence Informational Subsumption (CSIS). The first is for how relevant a particular sentence with respect to the general topic of the entire cluster is; the second is a measure for removing redundancy. The main difference of centroid-based summarization with Maximum Marginal Relevance (MMR) is that the metrics used in the former are not query-dependent. Given one cluster C of documents segmented into n sentences, and a compression rate R, a sequence of nR sentences are extracted in the order as they appear in

1 http://news.google.com
2 http://newsblaster.cs.columbia.edu
3 http://NewsInEssence.com

the original documents. The selection of the sentences is made by approximating their CBRU and CSIS. For each sentence s_i, three features are used to compute its salience:

- Centroid value: The centroid value C_i for sentence S_i is defined as $C_i = \sum_w C_{w,i}$, the sum of the centroid values $C_{w,i}$ of all words in the sentence.
- Positional value: It is used for giving leading sentences more importance. Let C_{max} be the centroid value of the highest ranked sentence in the document. Then the positional value is defined as $P_i = \frac{n-i+1}{n} C_{max}$, where n denotes the number of sentences in a cluster of documents.
- First-sentence overlap: The overlap value is defined as $F_i = \overrightarrow{S_1}\overrightarrow{S_i}$, the inner product between the sentence vectors for the current sentence i and the first sentence of the document.

The final score of each sentence is a combination of the three scores minus a redundancy penalty (R_s) for those sentences that overlap with sentences that have higher score values.

In general, most document clustering algorithms form a rectangular data matrix (e.g., document-term matrix or sentence-term matrix) to form separate groups of sentences. A centroid score is then assigned to each sentence that is based on the average cosine similarity between the sentence and the rest of the sentences from the same cluster. Finally, the sentences with the highest scores from each cluster are selected to form the summary [HKHBKM01; Zha02].

Wang et al. [WLZD08] proposed a multi-document summarization framework based on sentence-sentence similarities using semantic analysis and then applied symmetric matrix factorization in order to group the sentences into clusters. The standard non-negative matrix factorization (NMF) deals with a lower rank approximation of a nonnegative matrix, and has been performed successfully in clustering. The limitations of NMF is when the clusters have nonlinear structure, NMF cannot find the basis vectors that represent those clusters. The authors formulated NMF of a similarity matrix that contains similarity scores between each pair of sentences.

Graph Based Works

Graph-based approaches have been applied for both single-document and multi-document summarization [ER04b; MT04]. After the *PageRank* algorithm [PBMW99], graph models became popular [ER04b; MT04]. In graph based approach, vertices represent sentences and edges between vertices are given weights which are equal to the similarity between the two sentences. Sometimes, rather than giving weights to edges, the connections between vertices can be detected in a binary way: the vertices are connected only if the similarity between the two sentences exceeds a predefined threshold. Sentences that are related to many other sentences are likely to be the central and would have assigned more weight to be included in the summary.

The Markov Random Walk (MRW) model has been exploited for multi-document summarization by using 'voting' or 'recommendations' between sentences in the documents [ER04a; WY06; MT05; WY08]. The model [WY08] first constructs a directed or undirected graph to reflect the relationships between the sentences and then applies the graph-based ranking algorithm to

compute the rank scores for the sentences. Formally, let us assume, a set of documents D and $G = (V, E)$ be a graph, where V is the set of vertices and each vertex v_i in V is a sentence in the document set and E is the set of edges, each edge e_{ij} in E is associated with an affinity weight $f(i \to j)$ between sentences v_i and v_j $(i \neq j)$. The weight is computed using the cosine similarity between the two sentences:

$$f(i \to j) = sim_{cosine}(v_i, v_j) = \frac{\overrightarrow{v_i} \,.\, \overrightarrow{v_j}}{|\overrightarrow{v_i}| \times |\overrightarrow{v_j}|} \tag{6}$$

where $\overrightarrow{v_i}$ and $\overrightarrow{v_j}$ are the corresponding term vectors of v_i and v_j. Two vertices are connected if their affinity weight is greater than 0. The transition probability from v_i to v_j is defined by normalizing the corresponding affinity weight using the following equation:

$$p(i \to j) = \begin{cases} \frac{f(i \to j)}{\sum_{k=1}^{|V|} f(i \to k)}, & \text{if } \sum f \neq 0. \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

The row-normalized matrix $\tilde{M} = (\tilde{M}_{i,j})_{|V| \times |V|}$ describes G in which each entry corresponds to the transition probability, $\tilde{M}_{i,j} = p(i \to j)$. In order to make $\tilde{M}$ be a stochastic matrix, the rows with all zero elements are replaced by a smoothing vector with all elements set to $1/|V|$.

The sentence score for v_i sentence is computed from all the other sentences linked with it and it is formulated in a recursive form as defined in the *PageRank* algorithm [PBMW99] as follows:

$$SenScore(v_i) = \mu \,.\, \sum_{all j \neq i} SenScore(v_j).\, \tilde{M}_{j,i} + \frac{(1-\mu)}{|V|} \tag{8}$$

and the matrix form is

$$\overrightarrow{\lambda} = \mu \tilde{M}^T \overrightarrow{\lambda} + \frac{(1-\mu)}{|V|} \overrightarrow{e} \tag{9}$$

where $\overrightarrow{\lambda} = [SenScore(v_i)]_{|V| \times 1}$ is the vector of saliency scores for the sentences, $\overrightarrow{e}$ is a column vector with all elements equaling to 1 and μ is the damping factor usually set to 0.85 as in the *PageRank* algorithm [PBMW99]. For implementation, the initial scores of all sentences are set to 1 and the new scores of the sentences are generated using Equation 8.

In the MRW model, all sentences are treated uniformly. However, a document set can have a set of topic themes and each theme can be represented by a set of topic-related sentences [HL05; HSSTZW02]. The sentences from an important theme cluster should be ranked higher than the sentences in other theme clusters. Based on this idea, Wan and Yang [WY08] proposed Cluster-based Conditional Markov Random Walk Model (Cluster CMRW), which is an improvement of the MRW model or the *PageRank* algorithm [PBMW99] by incorporating the theme-cluster's importance while scoring the sentences. The new transition probability is defined as follows:

$$p(i \to j | clus(v_i), clus(v_j)) = \begin{cases} \frac{f(i \to j | clus(v_i), clus(v_j))}{\sum_{k=1}^{|V|} f(i \to k | clus(v_i), clus(v_k))}, & \text{if } \sum f \neq 0. \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

where $f(i \to j|clus(v_i), clus(v_j))$ is the new affinity weight between two sentences v_i and v_j, conditioned on the two clusters containing the two sentences.

$$\begin{aligned} f(i \to j|clus(v_i), clus(v_j)) &= \lambda.\ f(i \to j|clus(v_i)) + (1-\lambda).\ f(i \to j|clus(v_j)) \\ &= \lambda.\ f(i \to j).\ \pi(clus(v_i)).\ \omega(v_i, clus(v_i)) \\ &+ (1-\lambda).\ f(i \to j).\ \pi(clus(v_j)).\ \omega(v_j, clus(v_j)) \end{aligned} \tag{11}$$

where $\lambda \in [0, 1]$, $\pi(clus(v_i))$ determines the importance of the cluster $clus(v_i)$ in the document set D and it is defined as $\pi(clus(v_i)) = sim_{cosine}(clus(v_i), D)$, $\omega(v_i, clus(v_i))$ determines the importance between the sentence v_i and its cluster $clus(v_i)$ and it is defined as $\omega(v_i, clus(v_i)) = sim_{cosine}(v_i, clus(v_i))$.

Wan and Yang [WY06] improved the graph-ranking algorithm by differentiating intra-document links and inter-document links between sentences. Although the approach does not need any linguistic processing, but incorporating syntactic and semantic information while building the graph model [CJ08] shows better results than using tf*idf weights in cosine similarity. In order to reduce coherence problems, Leskovec et al. [LMFG05] have explored the graph model within which words and phrases (rather than sentences) are considered as vertices while edges represent the syntactic dependencies, inferred through a syntactic parser that leverages machine learning. Zhao et al. [ZWH09] proposed a query expansion algorithm which is similar to the topic-sensitive LexRank algorithm [OER05]. The previous query expansion methods that usually choose word synonyms as expansions in query-focused multi-document summarization. The authors rather select both informative and query relevant words based on the graph ranking results, add them into the original query and use the updated query to perform graph ranking again.

A document summarization framework [ZGH12] has been proposed by Zhang et al. to extract essential sentences from a document by considering sentence clusters information. There are three phrases in the framework: document modeling, sentence clustering and sentence ranking. The story document is modeled by a weighted graph with vertices that represent sentences of the document. The sentences are clustered into different groups; to alleviate the influence of unrelated sentences in clustering, an embedding process is employed to optimize the document model. In graph embedding, the initial weighted graph is embedded into lower dimension space. A sentence is expressed as a linear combination of its most similar sentences. After the graph embedding, the high similarity score between the sentences is enhanced while the low similarity score is reduced.

2.2.2 Time-biased Approaches

Different explicit tasks are identified for understanding the differences between the first stage in classical and time-biased summarization. Within each task, different approaches used for the first stage are reviewed. As discussed earlier, the first stage in summarization is a crucial stage to derive an intermediate representation of the input text. In this stage, we exhibit the approaches that indicate what are the necessary derivations taken into account for the inclusion of time in summarization. To the best of our knowledge, the most prominent identified tasks in this stage are: *first story detection/new*

event detection, *update summarization*, *novelty detection*, *burst detection*, *changes summarization*, *timeline summarization* and *temporal snippets generation*.

First story detection/new event detection

Event detection has been applied not only in the context of topic detection and tracking [ACDYY98; All02; AGK01; LWLM05] but also in other contexts such as tracking of natural disasters [SOM10] or event-based epidemic intelligence [AMM11]. The idea about monitoring changes over time in news coverage has started with Allan's et al. work [AGK01]. If a user has access to a stream of news stories on the same topic, it is difficult to look at every story due to rapid changes. In this situation, the user would go into the details when the changes within the topic trigger enough interest. This work arises out of Topic Detection and Tracking (TDT) [ACDYY98; All02], which consists of three major tasks: 1) segmenting a stream of data into distinct stories: stories are the collections of news articles reporting events that evolve over time; 2) identifying the first news item to discuss a new event from the segmented stories; 3) finding out the evolution of events in a set of stories. According to the authors' description [AGK01], "the problems tackled by TDT are all story-based rather than sentence based. In many ways, the temporal summarization problem is an event and sentence level analogue of TDT's *first story detection* problem, where the task is to identify the first story that discusses each topic in the news."

Allan et al. [AGK01] formalized the temporal summarization problem as a news topic which is made of a set of events and discussed in a sequence of news stories. Most of the sentences that describe one or more events are called *on-event* and the sentences which are not related with any of the events are called *off-event*. All sentences arriving in a specified time period can be considered together. The authors proposed the intermediate representation which is based on *language model* [PC98] to the input text. Specifically, given the text on a particular topic, the probabilistic model tries to estimate how the text from the topic is likely to be generated. For example, a word probability is estimated by:

$$P(w) = \frac{\sum_i tf(w, S_i)}{\sum_i |S_i|} \tag{12}$$

where $tf(w, S_i)$ represents the number of times word w occurs in story S_i. So, in this case a word's probability is estimated by the proportion of the time that it has already occurred. So, the probability of a sentence is the product of it's all word probabilities under the assumption that the word occurrences are independent.

Subašić and Berendt [SB10] considered the problem of tracking and representing the evolution of stories using co-occurrence analysis. They divided the whole corpus C into sets of documents c_i, $i = 1, \ldots, I$ depending on the chronological order of the documents that were published. For each set of documents c_i, the frequency of the co-occurrence of all pairs of content-bearing terms b_j within a window of w terms is calculated as:

$$freq_i(b_1, b_2) = \frac{\begin{array}{c}\#\text{ occurrences of both } b_1, b_2 \\ \text{within } w \text{ terms in all the documents from } c_i\end{array}}{\#\text{ all the documents in } c_i} \tag{13}$$

Then, this frequency measure is normalized to yield the measure *time relevance*, which measures the relevance of a term in a sub-corpus relative to the whole corpus as follows:

$$TR_i(b_1, b_2) = \frac{freq_i(b_1, b_2)}{freq_C(b_1, b_2)} \tag{14}$$

New event detection or *first story detection* traditionally focuses on news articles to discover and cluster events. However, other studies have been conducted with other dynamic text collections. The problem of *first story detection* from a stream of Twitter posts has been addressed using Locality-sensitive hashing (LSH) [POL10], an approach that can overcome the limitations of traditional approaches used in data streams. Given a sequence of stories, the goal of *first story detection* is to identify the first story which discusses a particular event. The problem of *first story detection* from tweets instead of news articles brings additional problems. The problems are to deal with a huge volume of data and noise simultaneously. In Twitter streaming setting, Petrović et al. [POL10] employed LSH [DIIM04], which is an approximation of Nearest Neighbor (NN) clustering algorithm in a sub-linear time. This method is built up in such a way that the probability of accumulating all neighbor points into the same bucket is much higher than the non-neighbors. For very large databases of high dimensional items, LSH is a particularly valuable technique for finding similar items. In these searches, it can drastically reduce the computational time, at the cost of a small probability of failing to find the exact closest match.

Sankaranarayanan et al. [SSTLS09] used a clustering approach to detect events using a text classifier while Becker et al. [BNG10] proposed a general framework for identifying events in social media documents via clustering. They showed that the similarity measure using a combined multiple context features (e.g., title, tags, upload or content creation time) of the document for clustering was more effective that using traditional textual similarity.

The WikiPop [Cg10] service can detect and present popular topics related to the users' interests using Wikipedia page view statistics. The novelty of this approach is the utilization of the knowledge base (Wikipedia link graph) which is based on the hypothesis that events can trigger an increased number of visits to the corresponding articles. Relatedly, Ahn et al. [AVCB11] clustered a set of popular Wikipedia articles by determining those article's page views and to summarize the clusters so that they best explain the relevant events. In Wikipedia, the pageviews for an article reflect its popularity. Based on this, they detected the most popular articles by examining the increased number of pageviews of those articles for the last fifteen days over those of the preceding fifteen day period rather than simply considering for a single day. They used the following algorithm for each article to determine the monthly trend value as increase in pageviews within last 30 days. The monthly trend value t^k of an article k is defined as:

$$t^k = \sum_{i=1}^{15} d_i^k - \sum_{i=16}^{30} d_i^k \tag{15}$$

where d_i^k = daily pageviews $i-1$ days ago for an article k.

Another interesting work by Tsagkias et al. [TRW11] studied approaches for discovering social media utterances (e.g., blog posts, tweets, diggs etc.) implicitly linked with news events. Generally, most of the discussions in social media are about the impact of news events (e.g., 85% of Twitter statuses are related to news [KLPM10]). Social media utterances may be linked

either explicitly or implicitly to news articles. Unlike an explicit-linked utterance, there is no trivial hyperlink in an implicit-linked utterance, rather the implicit utterance directly discusses the article's content. They proposed a three-step approach followed via query modeling, retrieval and fusion. In the query modeling step, multiple query models are derived to generate multiple queries from a given news article based on three strategies: (i) the article's internal document structure, (ii) explicitly linked social media utterances and (iii) term selection strategies. In the retrieval step, these queries are submitted to retrieve utterances separately for each of them from an index of social media utterances. After retrieval, in the final step, multiple ranked lists are merged into a single result list using data fusion techniques.

Becker et al. [BING12] examined how to automatically identify social media content associated with known events through diverse social media sites (such as, Flickr, YouTube, Last.fm, EventBrite, Facebook, Twitter, etc.), involving various combinations of the context features, namely, title, time, date, and location, of each event. They presented a query-oriented solution for retrieving social media documents for planned events, towards an improved browsing and search for event media. Moreover, they demonstrated how documents from these social media sites can be used to enhance document retrieval from other related sites for the same event.

Another work is Wikipedia Live Monitor [SVS13], which tracks article edits on different language versions of Wikipedia as signals for breaking news events. The edits of articles about the same topic, but written in different languages put into one cluster. Then, if this cluster satisfies the following breaking news criteria in which the parameters were determined empirically, it would be identified as breaking news candidate:

- $\geqslant$ 5 Occurrences: The cluster must have occurred in at least 5 edits.
- $\leqslant$ 60 Seconds Between Edits: The cluster may have at maximum 60 seconds in between edits.
- $\geqslant$ 2 Concurrent Editors: The cluster must have been edited by at least 2 concurrent editors.
- $\leqslant$ 240 Seconds Since Last Edit: The cluster's last edit may not be longer ago than 240 seconds.

As an extended application of Wikipedia Live Monitor [SVS13], recently Steiner [Ste14] proposed the breaking news events detection with the connection between Wikipedia and the world of social network sites. Osborne et al. [OPMMO12] showed the improvement of the quality of the potential events from Twitter with the exploration of the connection between Wikipedia and Twitter. Within Wikipedia, events are reflected through edits, page views and new page creation. Among them the authors tracked per-hour page views. The algorithm is simple; at each hour for each page i with page views w_i, a moving window of k hours over previous page view counts $w^i_{j-k}, \ldots, w^i_j$ is maintained. In a new hour, the moving window is updated for all pages and then applied Grubb's test [Gru69] to each moving window, determining if the latest page view number is an outlier with respect to previously seen page views.

Keegan et al. [KGC12] studied how the temporal dynamics of the editorial group are associated with the breaking events stated in Wikipedia articles: more editors means higher-quality "featured" articles [WH07]. They construct "article trajectories" that capture the relationships among editors modifying other editor's contribution from the revision histories of Wikipedia

articles. Four network statistics are captured to identify "tighter" or "looser" patterns in the editorial activity. "Tight" patterns exhibit the editors who have edited previously and returned to the article to make additional revisions. Whereas, "loose" patterns exhibit the editors contribute single revisions and do not return.

For event detection, Georgescu et al. [GKKNS13] studied and analyzed the edit history of Wikipedia. An update in Wikipedia reflects the modifications present in one revision when compared to the previous revision of an article. Each revision has its creation time (timestamp), its author, and, possibly, comments given by the updater. The Wikipedia Event Reporter system [GPKZSN13] determines distinct events through the clustering of updates by exploiting various information such as the update time, textual similarity and the position of the updates within an article. Each detected event is then summarized using a set of ranked sentences. A position-based clustering scheme is used based on the assumption that the sentences on the same topic are located in spatial proximity of each other on the article page. A cluster of positions is defined by a contiguous succession of positions with no more than 10 positions gap in between and each of the sentence cluster is decided to belong to the position cluster if it has the maximum overlap of positions with member sentences. The position clusters are then ranked by how many sentence clusters are assigned to them.The summarization for an individual event presents for each of the top-N position clusters, the representative sentences for the top-M clusters of sentences.

Update Summarization

The DUC launched *update summarization* as a pilot task in 2007 [Usu]. This task focuses on generating an update summary for multiple documents based on a common topic under the assumption that the user is familiar with a set of past documents. The purpose of each update summary is to inform the reader with the novel information on that specific topic. Therefore, the information needs addressed in update summarization are restricted to current and novel updates and the assumption in update summarization is that the user is already familiarized with the past information related to the topics.

There are previous Bayesian works mostly based on topic models to address the problem of *automatic summarization*. Haghighi and Vanderwende [HV09], and Chemudugunta et al. [CSS07] used generative models which learn to discriminate between the collection and the document-specific distributions in order to capture the major pieces of information in a dataset. Although hierarchical topic modeling approaches have shown remarkable performances while learning the major information from document collection, the main drawback of these approaches is that they are not designed to capture the novel information in a collection with respect to the previous one.

DualSum [DA12], a topic-model based approach used explicitly in the *update summarization* task, is basically a variation of the LDA model [BNJ03]. The goal of the DualSum model is to learn to distinguish the collection of earlier documents (the base) and the collection of recent documents (the update). A set of pairs of collections of documents $C = \{(A_i, B_i)\}_{i=1,\ldots,m}$ is given as an input for DualSum model, where A_i is a base document collection and B_i is an update document collection. In DualSum, documents are modeled as a bag of words, which are assumed to be sampled from a mixture of four latent topics. The four latent topics are: i) the background

topic ϕ^G, where the distribution is based on the general background words; ii) the document specific topic ϕ^{cd}, where the distribution is based on each document d in the base and update collection pair c; iii) the joint topic ϕ^{A_c}, where the distribution is based on the common words between the base and the update collection, i.e. the main event that both collections are discussing; and iv) the update topic ϕ^{B_c}, where the distribution is based on the specific words from the update collection. All four distributions in the DualSum model are learnt from a set of news collections but the topic probability for ϕ^{B_c} is zero when generating a base document. Once the learning is done, any of the four distributions or a combination of them can be used to provide the summary that best approximates the collection.

Clustering based methods are also popular in the context of traditional multi-document summarization. These methods usually apply different clustering techniques on the term-sentence matrices formed from the documents. After grouping the sentences into clusters, a centroid score is assigned to each sentence based on the average cosine similarity between one sentence and the rest of the sentences in the same cluster. Finally, the sentences with the highest score from each cluster are selected to form the summary. Besides this, there are incremental text clustering algorithms [GG04], whose main task is to detect novelty when new documents arrive. Wang and Li [WL10] proposed an Incremental hierarchical sentence clustering (IHSC) framework combined with document summarization techniques to update document summaries in real time. When new documents or sentences are added, the IHSC framework re-organizes the sentence clusters so that the corresponding summaries can be updated quickly. The COBWEB algorithm [Fis87; GLF89], which was originally built by Fisher et al. is applied to build a sentence hierarchy of the document collection. When a new element is added, the COBWEB algorithm traverses the tree in a top-down fashion starting from the root. During the traversal, the COBWEB algorithm executes one of the four possible operations (*insert*, *create*, *merge* or *split*) in order to maximize the criterion function. The criteria function is defined as:

$$\frac{\sum_{k=1}^{K} P(C_k) \sum_i \sum_j [P(A_i = V_{ij}|C_k)^2 - P(A_i = V_{ij})^2]}{K} \tag{16}$$

Where $A_i = V_{ij}$ is an attribute-value pair and C_k is a cluster [Fis87]. This criteria function is a trade-off between intra-class similarity (through $P(A_i = V_{ij}|C_k)$) and inter-class dissimilarity (through $P(C_k|A_i = V_{ij})$). Moreover, Wang and Li [WL10] used Katz's distribution based COBWEB algorithm [SCKDP06] to create the sentence hierarchical tree incrementally. When all the documents or sentences arrive, the users can cut the hierarchy tree at any level based on the length of the summary.

McCreadie et al. [MMO14] introduced the task of Incremental update summarization (IUS), which aims to select sentences from news streams to present only the updates about an event towards the purpose of tracking that event. The authors proposed an approach that treats the IUS problem as a rank cut-off problem, inspired by the previous work [AKR09]. This approach follows to predict the optimal rank cut-off for the update summary S_t at time interval t within an event e based on two concepts — *prevalence* (the information is related to an event) and *novelty* (the updates about an event are not discussed yet). A regression model is trained using 330 features that trade-off between a deeper cut-off and a shallower cut-off. A deeper cut-off can return redundant content where as a shallower cut-off has the risk of missing important information.

Graph-based ranking algorithms are also used in update summarization task. Ranking Sentences with Positive and Negative Reinforcement (PNR^2) [WFQY08] and Manifold ranking with sink points (MRSP) [DGZC10] are two such methods. PNR^2 models both positive and negative reinforcements between sentences to decide their scores and, as a result, it extracts the sentences that are not only salient but also novel comparatively to the base collection. Positive reinforcement captures the idea that a sentence (from either the base or update collection) is more important if it correlates to the other important sentences in the same collection whereas negative reinforcement reflects the notion that a sentence from the base collection is less important if it correlates to the important sentences in the update collection and vice versa. In MRSP, the sink points are the sentences whose ranking scores are fixed to the minimum ranking score (i.e. zero in this case) on the manifold during the ranking process. The sentences sharing similar information with the sink points are penalized during the ranking process based on the intrinsic sentence manifold. Recently, QCQPSum [LDS13] was developed to avoid the problem identified in PNR^2 and MRSP, namely that the salience determination of the sentences in the update collection is disturbed by the base collection. QCQPSum is a Quadratically constrained quadratic programming (QCQP) problem which is NP-hard. To overcome this problem, the authors proposed an approximate method (QPSum) that can solve it in polynomial time.

When capturing the information changed in current documents in comparison with previous documents, the first challenge is filtering the redundant information. Zhang et al. [ZDXC09] proposed three filtering approaches to measure the similarity of sentences between earlier and current information: document filtering, summary filtering and union filtering based on the degree of membership from the fuzzy set theory. After that, the filtered sentences are ranked using two approaches. The first is a signature based approach in which temporal topic signatures are extracted from the filtered sentences. The second is a manifold ranking based approach in which the macro-structure of the filtered sentences can be reserved. Support Vector Regression is used as a filter [SKLC08] to extract sentences that resemble *first sentences* in the entire news articles. The purpose behind this idea is that *first sentences* are very focused and contain less anaphoric expressions. After extracting the sentences, a modified version of FastSum [SK08] is applied.

Novelty Detection

Novelty detection is an important task to reduce the amount of redundant as well as non-relevant information presented to a user. According to Li and Croft [LC08], the definition of novelty is given as "novelty or new information means new answers to the potential questions representing a user's request or information need". The TREC novelty tracks, which are related to novelty detection, were conducted for three years [Har02; SH03; Sob04]. It is an important task which can be used in many potential applications, such as new event detection, document filtering, cross-document summarization or temporal summarization.

Novelty detection can be performed at two different levels: the event level and the sentence level. At the event level [YZCJ02; KA04], a novel document is required to be relevant to a topic (i.e., a query) and also to discuss a new event. At the sentence level [AWB03; JZX03; Lit03], a novel sentence needs to be both relevant to a topic and provide new information. This means that a novel sentence may either discuss a new event or present new

information about an old event. The various definitions of novelty, however, are indirectly related to intuitive notions of removing redundancy [SCL08; ZCM02].

A unified pattern-based approach is proposed by Li and Croft [LC08] for novelty detection. The identification and extraction of information patterns (or features) is crucial in this NLP-like approach. The three information patterns studied here are: sentence lengths, named entities and opinion patterns. A statistical analysis of these information patterns is performed to distinguish relevant sentences from non-relevant ones, and novel sentences from non-novel ones.

Burst Detection

Burst detection finds the elevated occurrence of activities over time. Burstiness has been explored in various applications e.g., telecommunication, astrophysics, finance, the databases of scientific publications, news articles, social media, etc. There are the burst detection algorithms which define bursts in terms of an arrival rate [LBK09] or the peaks with an increased level of the update activities within a short period of time in a set of revised documents [GPKZSN13]. However, in other scenarios, the burst is derived as intervals of increasing 'momentum' using the concept from physics [HP10].

As the volume of online documents has drastically increased, the analysis of bursts is an attempt to deal with the temporal effects in a series of document streams or in a set of revised documents. Klinkenberg and Joachims [KJ00] presented a method for detecting the bursts with support vector machines. They have studied this problem in the pattern recognition framework. Each example $\vec{z} = (\vec{x}, y)$ consists of a feature vector $\vec{x} \in \mathbf{R}^N$ and a label $y \in \{-1, +1\}$. Data arrives over time in batches, assuming each batch contains m examples.

$$\vec{z}_{(1,1)}, \ldots, \vec{z}_{(1,m)}, \vec{z}_{(2,1)}, \ldots, \vec{z}_{(2,m)}, \ldots, \vec{z}_{(t,1)}, \ldots, \vec{z}_{(t,m)}, \vec{z}_{(t+1,1)}, \ldots, \vec{z}_{(t+1,m)}$$

$\vec{z}_{(i,j)}$ denotes the j-th example of batch i. For each batch i the data is independently identically distributed with respect to a distribution $Pr_i(\vec{x}, y)$. The distribution $Pr_i(\vec{x}, y)$ and $Pr_{i+1}(\vec{x}, y)$ between batches will differ depending on the burst. In machine learning, capturing bursts is often handled by time windows of either fixed [MCFMZ94] or complicated heuristics [WK96] on the training data. But, Klinkenberg and Joachims [KJ00] presented an approach for selecting an appropriate window size which uses support vector machines [VV98] so that the estimated generalization error on test examples is minimized. This window should include only those examples which are very close to the current target concept. To estimate the generalization error the authors used a special form of $\xi\alpha$-estimates [Joa00], which are an efficient method for estimating the performance of a support vector machine.

Kleinberg's burst algorithm [Kle03] models bursts with an infinite state automaton in which each state represents a message arrival rate and bursts appear naturally as state transitions — from a lower state to a higher state. The most basic model of this type would be constructed from a probabilistic automaton $\mathcal{A}$ with two states q_0 and q_1. In state q_0, messages are emitted at a slow rate with gaps x between consecutive messages according to a density function $f_0(x) = \alpha_0 e^{-\alpha_0 x}$. When $\mathcal{A}$ is in state q_1, messages are emitted at a faster rate according to $f_1(x) = \alpha_1 e^{-\alpha_1 x}$, where $\alpha_1 > \alpha_0$. The higher

the state, the smaller the expected time gap between consecutive messages. $\mathcal{A}$ changes to another state with probability $p \in (0,1)$ and remaining in its current state with probability $1-p$. By assigning costs to state transitions, very short bursts can be avoided.

Similarly, Zhu and Shasha [ZS03] detected bursts as the activity of finding abnormal aggregates in data streams. Given an aggregate function F, the problem of interest is to discover subsequences s of a time series stream such that $F(s) >> F(\acute{s})$ for most subsequences $\acute{s}$ of size $|s|$. This aggregates are based on sliding windows over data streams. The authors designed a data structure, called the Shifted Wavelet Tree for detecting interesting aggregates over many sliding window sizes simultaneously in near linear time. The sliding window size is discovered by the system.

Scholz et al. [SK07] used a new ensemble method by learning several data streams to detect concept drift. However, the ensemble method itself has the problem about how to manage multiple classifiers effectively. He and Parker [HP10] used a Moving Average Convergence/Divergence (MACD) histogram to find bursts, while Fukumoto et al. [FSTM13] applied MACD to find topics, in contrast Suzuki et al. [SF14] applied it to the topic candidates obtained by LDA in order to identify the topic words. The MACD is a technique to analyze stock market trends [Mur99]. The MACD of a variable x_t is defined by the difference of the n_1-day and the n_2-day moving averages:

$$MACD(n_1, n_2) = EMA(n_1) - EMA(n_2)$$

where $EMA(n_i)$ refers to n_i-day Exponential Moving Average (EMA). For a variable $x = x(t)$ which has a corresponding discrete time series $X = \{x_t | t = 0, 1, \ldots\}$, the n-day EMA is defined by as follows:

$$\begin{aligned} EMA(n)[x]_t &= \alpha x_t + (1-\alpha) EMA(n-1)[x]_{t-1} \\ &= \sum_{k=0}^{n} \alpha(1-\alpha)^k x_{t-k} \end{aligned} \tag{17}$$

α refers to a smoothing factor and it is often taken to be $\frac{2}{(n+1)}$. The MACD histogram shows the difference between the MACD and its moving average.

$$hist(n_1, n_2, n_3) = MACD(n_1, n_2) - EMA(n_3)[MACD(n_1, n_2)] \tag{18}$$

Changes Summarization

Changes summarization is described as a task to summarize meaningful changes in the context of dynamic document collections devoted to a common topic [LPT00; JBI04; NRD08]. A change detection application [Cha] has been made to provide a list of changes for the given web page address by email or as a composition of different page versions for better visualization of changes. However, the user is mostly overloaded with the meaningless changes like, for example, modified syntax, color or changed links. In addition to this, there is a little research done on the extraction and summarization of meaningful and significant changes in an algorithmic level from any dynamic text collection.

WebCQ [LPT00] is designed to monitor changes to web pages and to notify users of interesting changes with a personalized customization. There are various types of web page sentinels for detecting changes to any web

page. The changes in a web page can be in text contents, images, links, tables, lists, keywords or any arbitrary text change (any change to the text fragment specified by a regular expression). Furthermore, *WebCQ* allows users to set up two notification methods: email and personalized web bulletins according to their specifications.

ChangeSummarizer [JBI04] periodically monitors the textual changes in web collections and produces their summary related to a specific topic. *ChangeSummarizer* helps users in searching for new relevant information in their interest by providing the summary of recent, important changes related to that topic. Each web page is compared with the old and new web collections. After comparison, the new terms are extracted. The system then calculates the scores for each term according to the popularity of that term in static and dynamic parts of the collection based on its frequencies. The important part is to notice that for *changes summarization* task, among all terms the changed terms are considered for scoring. Each term's score is generated using the following scoring function:

$$S_i = (1 + \frac{\sum_{j=1}^{N_{doc}} [\frac{n_{jc}}{N_{jc}+1} - \alpha \times \frac{n_{js}}{N_{js}+1}]}{N_{doc}}) \times \exp(\frac{n_{icp}}{N_{cdoc}+1} - \alpha \times \frac{n_{isp}}{N_{sdoc}+1}) \tag{19}$$

where the descriptions of the symbols are the following:
S_i - the score for term i
N_{doc} - number of pages in the web collection
N_{sdoc} - number of static pages in the web collection
N_{cdoc} - number of changed pages in the web collection
n_{isp} - number of pages in the web collection where static parts contain term i
n_{icp} - number of pages in the web collection where changed parts contain term i
N_{js} - number of static terms in page j in the web collection
N_{jc} - number of changed terms in page j in the web collection
n_{js} - number of term i in static part of page j in the web collection
n_{jc} - number of term i in changed part of page j in the web collection
α - ranges from 0 to 1.

Another interesting work, WikiChanges [NRD08] which is a web-based application designed to plot Wikipedia article's revision history in real time and to produce a temporal summary. The summary addresses what changes occurred during a given set of revisions. The complete revision history of any Wikipedia article is used as the source of enormous temporal information. A very simple approach is proposed based on the terms inserted between a start and end revisions (all intermediary revisions are ignored) of an article. Each term is then scored by subtracting the old term's frequency count from the new term's frequency count. The final top scored terms are presented as an automatic summary to the final user using tag clouds.

Recently, Google has made a patent [CB15] about the idea of automatically summarizing the changes made to a document in a collaborative environment via electronic messages. An electronic message that includes the summary of the changes made to the document is being sent to at least one recipient. Live information relating to the document (e.g., who is currently editing the document, who is assigned to review the document, is the document in a draft state?, a final state?, a last change to the document, and the like) may also be displayed through electronic messages and is automatically updated when the live information changes.

Timeline Summarization

Timeline summarization [CL04], which organizes events by date on news articles is a special case of *multi-document summarization*. It allows users to have an quick overview of events and fast news browsing relating to their interest from a collection of various news sources. As manually created timelines are very time-consuming, there is a need for automatic approaches. Although, Google News Timeline [Goob] automatically clusters news stories for different topics but the stories in each group are merely sorted in a chronological [JSMH10] order. The challenges in *Timeline summarization* are (i) selecting important dates in the timeline [KTHMB12; BAQ13] and (ii) generating a good summary for each of the selected dates [CL04; YWOKLZ11].

Chieu et al. [CL04] present a framework that extracts events from a collection of documents for a given query and places such events along a timeline. The authors proposed two metrics: *interest* and *burstiness* for ranking important sentences. The metric *interest* is based on the principle that important events are often repeated in many news articles over a time span whereas, the metric *burstiness* is based on the principle that events often cluster surrounding the date of their occurrences. The steps of the framework for timeline extraction are: (i) a sentence is considered relevant to the given query, if one of the terms from the query presents in that sentence; (ii) each sentence is mapped to one date; (iii) rank the sentences based on the *interest* and *burstiness* metrics; (iv) remove duplicate sentences; and (v) place top N sentences along a timeline.

Yan et al. [YWOKLZ11] proposed a framework for summarizing an evolution trajectory along a timeline from the massive collection of time-stamped web documents. So, for a user given query Q, a collection of sentences C is collected from query related documents. Then, the sentences are clustered into $\{C_1, C_2, \ldots, C_{|T|}\}$ associated with the published dates $T = \{t_1, t_2, \ldots, t_{|T|}\}$. An evolutionary timeline, which consists of a series of individual but correlated summaries, i.e. $I = \{I_1, I_2, \ldots, I_{|T|}\}$, where I_i is a subset of C_i on date t_i is presented as an output.

Tran et al. [BAQ13] presented a framework for automatically constructing timeline summaries from a collection of news articles A_q related to a topic q. They have used a supervised Linear Regression model to train the model by exploiting manually created timelines for selecting important dates and contents. The corpus is divided into training and test sets and used the *leave one out* approach for training the model.

Zhao et al. [ZGYHL13] studied the generation of timeline summaries by incorporating social attention into it. The existing methods for timeline generation only consider news streams but not users' collective interests. The authors proposed an approach to capture social attention from tweets through learning a generative mixture model. Then, the learnt model is transformed into a vector of each word dimension to its corresponding probability. They have shown that the incorporation of users' interests is helpful to improve the timeline summaries in the context of both informativeness and interestingness.

Tran et al. [TAH15] presented another approach that exploits only the news headlines instead of the news articles' full content for generating timeline summaries. The earlier works on the generation of summaries for each of the important dates usually focus on the extraction of relevant sentences from the article text but the main drawback is these approaches do not guarantee the content coherence. Unlike the previous approaches, the intuition of using news headlines is to generate more coherent summaries than sum-

maries that are composed of selected sentences from different parts of the news articles. The authors tried to select those headlines that maximize all three aspects influence (the headline can give hint about what will happen in the future), spread (the headlines that are similar with other headlines) and informing value (when the headline describes an event).

Temporal Snippets Generation

Temporal information can be found in every document either explicitly or implicitly. Recognizing such information and exploiting them for document search and exploration tasks would be interesting. The inclusion of time-sensitive information in the snippet may help the user to judge better about document relevance and improve the searcher's experience [ABYG09; AGBY11]. In general, snippets present a couple of lines with highlighted keywords in web search engine. Time-centered snippets, called TSnippet [ABYG09; AGBY11] are introduced as document surrogates for document retrieval and exploration. In this work, they studied how temporal expressions appears in documents and how they can be included in a snippet.

Campos et al. [CDJ11] examined the extent to which queries and search snippets contained explicit temporal expressions. They found that snippets were a rich source of temporal information and could be used in query understanding. On the other hand, Svore et al. [STDK12] explored the effectiveness of including new web page content in search result snippets. According to the authors, the results show that the users find the inclusion of new content in snippet is useful for trending queries and when the page has not been recently crawled.

2.3 APPROACHES FOR SCORING SENTENCES & SELECTING SUMMARY SENTENCES

Once an intermediate representation has been derived after the first stage, the next goal is to assign a score to each sentence. The score of each sentence indicates its significance. In the third successive step, all sentences are ranked according to their significance and a set of higher ranked sentences is included into the summary depending on its length. Before the inclusion of any sentence into the summary, there is similarity checking between the chosen sentences in order to avoid redundant sentences in the presented summary.

By examining all the three stages, we point out the first stage among them to be the main factor to discriminate time-biased summarization over classical. Because, in the first stage, the approaches used for deriving the representations give a list of key indicators in order to generate a time-biased summary. This stage not only maps the extracted information with the time but also finds out the new information within the time. This is the new requirement specific to time-biased summarization. The other aforementioned requirements of a summary such as, salience, non-redundancy and relevance [CG98; DGZC10] are determined by the approaches used in the latter stages. Moreover, the approaches that are applied in second and third stages for classical text summarization can be directly used in time-biased summarization without making any change. The main focus of this survey is to highlight the changes in the way a specific stage is performed and it can markedly change the type of a summary. This is the reason why we stress

upon reviewing the approaches applied in the first stage in more detail. Regarding the latter stages we describe some of the techniques in brief so that if anyone is interested they can go through the techniques used in these stages available in the literature on classical text summarization [NM12; LP12].

2.3.1 Classical Approaches

In Luhn's work [Luh58], the 'significance' factor of a sentence is acquired from an analysis of its words. The author proposed that the significance of a sentence is determined from the relative position of its significant words. A statistical procedure is applied considering those portions of sentences as being significantly related, which are bracketed by significant words. This bracket is taken by measuring the distance at which any two significant words have a useful limit, which is four or five non-significant words between them. A sentence significance is then computed by the square of the number of significant words within the bracket divided by the total number of bracketed words. In multi-document summarization literature [RBGZ01; ZDSML05], various systems used sentence position instead of word position as a feature while scoring candidate sentences. By using topic signatures [CSO06; HL05] as an approach for intermediate representation, the sentence scoring function mainly uses either the number of topic signatures each sentence contains or the proportion of topic signatures in a sentence. The first approach tends to choose longer sentences whereas the second one favors density of topic words.

In SUMBASIC [NV05], a weight assigned to each sentence S_j is equal to the average probability of the words in the sentence, i.e.,

$$Weight(S_j) = \frac{\sum_{w_i \in S_j} p(w_i)}{|\{w_i | w_i \in S_j\}|} \tag{20}$$

where each word probability $p(w_i)$ is calculated using Equation 1. The sentence selection strategy of SUMBASIC [NV05] follows a greedy strategy. For the summary, it picks the best scoring sentence that contains the higher probable words and then updates the probability of each word w_i. The update rule is:

$$p_{new}(w_i) = P_{old}(w_i) \times P_{old}(w_i) \tag{21}$$

By updating the word probabilities in this intuitive way, the words with low probabilities initially can have the higher impact on the choice of subsequent sentences. Instead of using greedy search, Yih et al. [YGVS07] formalized the problem of choosing the best combination of sentences as an optimization problem and proposed an explicit search algorithm, namely a stack decoder to search for the best combination of sentences. This algorithm optimizes the occurrence of important words globally over the entire summary that is better than the heuristic approach [NV05]. Nenkova et al. [NVM06] studied that three factors related to frequency have influence on summarization: content word frequency, composition functions for estimating sentence importance from word frequency, and the adjustment of frequency weights based on context.

In *query-focused* summarization, the importance of each sentence is assessed by a combination of two factors: how relevant it is with respect to the user query (or with respect to query independent summarization, the

sentence needs to carry the highest importance score) and how important it is in the context of the input in which it appears. Carbonell and Goldstein [CG98] have made a major contribution to query-focused summarization by introducing MMR measure. The idea is to include each candidate sentence/passage dynamically if it is considered novel with respect to the previous included sentences/passages. If Q be a query or user profile, R be a ranked list of documents retrieved by a search engine, S be the set of already selected documents in a particular step and $R \setminus S$ be the set of yet unselected documents in R, then the Marginal Relevance (MR) for each candidate document, $D_i \in R \setminus S$ is computed as:

$$MR(D_i) = \lambda \times Sim_1(D_i, Q) - (1 - \lambda) \times \max_{D_j \in S} Sim_2(D_i, D_j) \tag{22}$$

where λ is a parameter in the range of $[0, 1]$ that controls the relative importance between relevance and redundancy. Sim_1 and Sim_2 are two cosine similarity measures. The document which is getting the highest MR, $D_{MMR} = \arg \max_{D_i \in R \setminus S} MR(D_i)$ is then added to S. The procedure is continuing until a maximum number of documents are selected or a minimum relevance threshold is achieved. Generating snippets for search engines are the examples of query focused summarization [TTHW07; VH06].

Although this MMR technique is widely used in *query-focused* summarization, but in *extractive* summarization [XL08], the final score of i-th sentence, S_i is calculated by adapting Equation 22 as follows:

$$MMR(S_i) = \lambda \times Sim_1(S_i, D) - (1 - \lambda) \times Sim_2(S_i, Summ) \tag{23}$$

where D is the document vector, *Summ* represents the sentences that have been extracted into the summary, and λ is used to adjust the combined score to emphasize the relevance or to avoid redundancy. The two similarity functions (Sim_1 and Sim_2) represent the similarity of a sentence to the entire document and to the selected summary, respectively. The sentences with the highest MMR scores will be iteratively added into the summary until the summary reaches a predefined size.

Another work by Celikyilmaz et al. [CHT11] is presented for extracting sentences that focus on both issues: topically coherent and non-redundant. They have built a two-tiered hierarchical topic model to capture higher-level topics, which are multinomials over lower-level topics resulting in less redundant summaries. This two-tiered model is inspired by the hierarchical topic model, PAM proposed by Li and McCallum [LM06]. In the PAM model, the lower-level topic significance (TS) based on the higher-level topic is calculated as follows:

$$TS(z_k) = \frac{1}{D} \sum_{d \in D} \frac{1}{K_1} \sum_{k_1}^{K_1} p(z_{sub}^{k} | z_{sup}^{k_1}) \tag{24}$$

where z_{sub}^{k} is a lower-level topic $k = 1, \ldots, K_2$ and $z_{sup}^{k_1}$ is a higher-level topic k_1. The conditional probability of lower-level topic k given a higher-level topic k_1, $p(z_{sub}^{k} | z_{sup}^{k_1})$ explains the variation of that lower-level topic in relation to other higher-level topics. The higher the variation over the entire corpus, sentences from such topics will have higher importance for reducing redundancy:

$$score^{PAM}(s_i) = \frac{1}{K_2} \sum_{k}^{K_2} \prod_{w \in s_i} p(w | z_{sub}^{k}) * TS(z_k) \tag{25}$$

where the word w belongs to sentence s_i and each sentence score $score^{PAM}(s_i)$ is calculated by imposing each lower-level topic's significance additionally on vocabulary words.

Most extractive summarization approaches attempt to select a set of sentences from the original text based on the relevance of the main ideas expressed in the original text and then put them together in a coherent manner. The KL-divergence between two unigram word distributions P and Q is given by:

$$KL(P\|Q) = \sum_{w} P(w) \log \frac{P(w)}{Q(w)} \tag{26}$$

This measure is used for selecting sentences into a summary in several systems [LM09; HV09]. Since the problem of finding the subset of sentences from a cluster of documents that minimizes the KL divergence is NP-complete, a greedy algorithm is often used in practice. The HIERSUM [HV09] model finds the summary with sentences that minimizes the KL-divergence between the estimated content distribution ϕ_c and the summary word distribution P_S:

$$S^* = \min_{S:|S|\leqslant L} KL(\phi_c\|P_S) \tag{27}$$

where sentences are greedily added to the summary one at a time until the summary has reached the maximum word limit, L.

In contrast to the previous approach, Mason and Charniak [MC11] combine the KL-divergence of both content and document-specific word distributions linearly for penalizing sentences that contain document-specific topics:

$$S^* = \min_{S:|S|\leqslant L} KL(\phi_c\|P_S) - KL(\phi_d\|P_S) \tag{28}$$

Recently, Ouyang et al. [OLZLL13] proposed the subsuming relationship between sentences to define a conditional saliency measure of the sentences instead of the general saliency measures used in most existing methods.

2.3.2 Time-biased Approaches

After the derivation of the intermediate representation from the input text, the approaches discussed previously for scoring and selecting summary sentences in classical text summarization can be used in temporal text summarization without constraining time-biased features on those approaches. Although there are studies which are conducted to develop time-biased sentence ranking methods to cover more temporal concepts for summarization.

Some web pages change more rapidly than others. Including such web documents into the collection could be beneficial to produce better quality summaries. Periodically, a list of common weighted terms are taken from all changes. Jatowt et al. [JBI04] showed that for a web page (S_t), it is possible to describe the value of "commonness" of its dynamic content simply by summing weights of all terms and dividing the sum by the number of all terms in the text, N_d.

$$S_t = \frac{\sum_{j=1}^{N_d} S_{jd}}{N_d} \tag{29}$$

In addition to this, the number of times that the web page changed during the whole monitoring process is also incorporated by an up-to-date-ness function D. The motivation is to score higher the latest changes than the old ones.

$$D = \sum_{t=1}^{T} \frac{S_t}{(T - t + 1)} \tag{30}$$

At end, web pages are ranked according to the value of the function D.

Yan et al. [YWOKLZ11] proposed the framework for generating an evolutionary timeline which consists of a series of individual but correlated summaries, i.e. $I = \{I_1, I_2, \ldots, I_{|T|}\}$. Each individual summary I_i is scored by a function, which is based on the weighted combination of *relevance* (F_r), *coverage* (F_{cv}), *coherence* (F_{ch}) and *diversity* (F_d).

$$U(I_i) = w_1 F_r(I_i) + w_2 F_{cv}(I_i) + w_3 F_{ch}(I_i) + w_4 F_d(I_i) \tag{31}$$

According to the scores of timeline attributes such as *relevance* (F_r), *coverage* (F_{cv}), *coherence* (F_{ch}) and *diversity* (F_d), summaries are generated by ranking sentences in an optimized way through iterative substitution from a set of sentences to a subset of sentences under constraints. The optimization is a trade-off between the neighboring set locally and the global collection of sentences. For an user given query Q, a collection of sentences C are collected from query related documents. Then, the sentence clusters are formed into $\{C_1, C_2, \ldots, C_{|T|}\}$ associated with the published dates $T = \{t_1, t_2, \ldots, t_{|T|}\}$. So, the utility function for summary I_i is given as:

$$U(I_i) = \lambda . U(I_i)|C_i + (1 - \lambda) . U(I_i)|C \tag{32}$$

where $U(I_i)$ is generated using Equation 31.

2.4 EVALUATION

Conventional automatic summarization evaluation can be broadly classified into two categories [JG96]: intrinsic and extrinsic evaluation. The intrinsic evaluation refers to the evaluation that tests a summarization system on accomplishing its own purposes [Lin04; RJST04; TV04; NP04; HLZF06]. The extrinsic evaluation checks the summarization based on how it affects other tasks like document retrieval, and in particular relevance filtering [HDMS07; MHKHFS99]. In this study, we concentrate on the automatic intrinsic summarization evaluation. The approaches for existing evaluation frameworks and metrics are reviewed, as well as new proposed frameworks that have emerged are taken into account to evaluate the time-biased summaries. The main difference between classical and time-biased approaches with respect to the evaluation is building a framework that considers the important aspect, summaries related with time. While reviewing the intrinsic summarization evaluation to assess the informativeness and the quality of a summary, the approaches are categorized into automatic and user evaluation. Figure 2 provides an overview of the evaluation approaches for classical and time-biased summarization. There have emerged new metrics (the boxes marked without any background in Figure 2) specifically for the automatic evaluation of time-biased summaries. Grey background boxes highlight the approaches mainly used in classical summarization evaluation but they

could be used for time-biased summarization as well. In the latter case, the most crucial thing is to design an evaluation framework that can use the classical evaluation metrics inside whereas the framework tackles itself the time-bias.

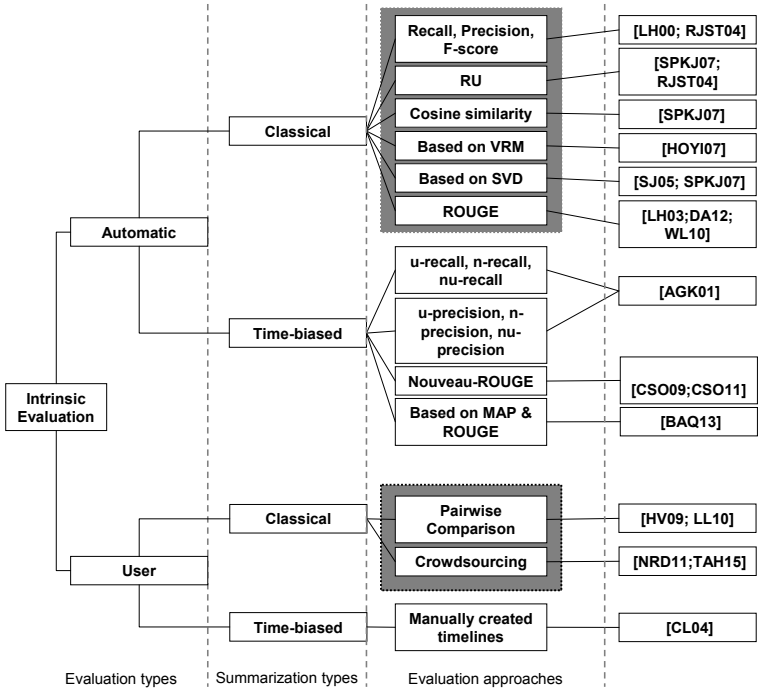

Figure 2: An overview of the evaluation approaches for classical and time-biased summarization. Grey background boxes highlight the approaches mainly used in classical summarization evaluation but they could be used for time-biased summarization as well. In the latter case, the most crucial thing is to design an evaluation framework that can tackle itself the time-bias.

2.4.1 Automatic Evaluation

In order to evaluate the effectiveness of the summary sentences by using any extractive summarization approach with human annotated model summaries, the measures recall (R), precision (P) and F-score (F) are computed [LHo0; RJST04]. They are defined by:

$$R = \frac{N_{me}}{N_m} \tag{33}$$

$$P = \frac{N_{me}}{N_e} \tag{34}$$

$$F = \frac{(1+\beta^2)PR}{\beta^2 P + R} \tag{35}$$

where N_{me} denotes the number of sentences extracted that also appear in the model summary, N_m denotes the number of sentences in the model

summary, N_e denotes the number of sentences extracted by the system and β represents the relative importance of R and P.

On the other hand, Allan et al. [AGK01] built an automatic evaluation framework that has similarities with the evaluation metrics recall and precision in IR but with the inclusion of temporal properties. Recall and precision only consider the relevance property for a retrieved document. But, the authors in their study defined recall and precision in terms of the following properties:

- Useful sentences, which have the potential to be included into the summary. The sentences, which discuss one or more of the events in the topic, are called *on-event* sentences whereas sentences that do not describe any of the events, are called *off-event* sentences. So, except *off-event* sentences, all *on-event* sentences are useful.
- Novel sentence is the first sentence about an event, whereas all following sentences discussing the same event are not.

They assumed that the entire set of sentences is divided into a set of useful sentences, U and a set of non-useful sentence, $\bar{U}$ and E was the set of v events, $E = \{e_1, e_2, \ldots, e_v\}$ whereas, S was the set of sentences $S = \{s_1, s_2, \ldots\}$. Moreover, they assumed that S_m is the subset of S, $S_m = \{s_1, \ldots, s_m\}$ and $C(X)$ is represented the set of events from E that are mentioned in the set of sentences X. All measures are taken after r sentences have been seen in the ranked list. $I(exp)$ is 1 if exp is true and 0 if not.

Recall and precision in terms of the 'useful' factor is defined as follows. The measure, u-recall is the proportion of *on-event* (useful) sentences that have been retrieved and u-precision is the proportion of retrieved sentences that are *on-event* for some event.

$$\text{u-recall} = \frac{|\, S_r \cap U \,|}{|\, U \,|} \tag{36}$$

$$\text{u-precision} = \frac{|\, S_r \cap U \,|}{|\, S_r \,|} \tag{37}$$

Recall and precision in terms of the 'novel' factor is defined as follows. A sentence is called novel if it covers one or more events that were not covered by any previous sentence. The measure, n-recall is the proportion of events that have been covered so far and n-precision determines whether the top ranked sentences are novel. The first part of Equation 39 checks if the top most sentence is novel or not, then the summation does the same checking for each following sentence.

$$\text{n-recall} = \frac{|\, C(S_r) \,|}{|\, E \,|} \tag{38}$$

$$\text{n-precision} = \frac{I(C(S_1) > 0) + \sum_{i=2}^{r} I(C(S_i) > C(S_{i-1}))}{|\, S_r \cap U \,|} \tag{39}$$

By combining both factors usefulness and novelty the measures become:

$$\text{nu-recall} = \frac{|\, C(S_r) \,|}{|\, E \,|} = \text{n-recall} \tag{40}$$

$$\text{nu-precision} = \frac{I(C(S_1) > 0) + \sum_{i=2}^{r} I(C(S_i) > C(S_{i-1}))}{|\, S_r \,|} \tag{41}$$

The main problem with recall (R) and precision (P) measures is that two equally good summaries may be judged very differently. For example, a manual summary contains the first and second sentences from a document and two systems produce two summaries separately in which one summary contains the first and second sentence from that document, and the other summary contains the first and third sentences respectively. Now, if the third sentence is an equally good alternative to the second sentence, the evaluation metrics precision and recall give much higher score to the summary which contains the first and second sentences compared to the other summary.

Steinberger et al. and Radev et al. [SPKJ07; RJST04] choose Relative Utility (RU) as an evaluation measure. With RU, a number of judges, $(N \geqslant 1)$ are asked to assign confidence values for their inclusion into the summary to all n sentences in a document. For example, a document with five sentences [1 2 3 4 5] is represented as [1/5 2/4 3/4 4/1 5/2]. The second number in each pair indicates the degree to which the given sentence should be part of the summary according to the judgment of a human. This number is called the utility of the sentence. In this example, the utility scores are 5, 4, 4, 1 and 2 for the first, second, third, fourth and fifth sentence, respectively. Now, back to the previous problem, with two system-generated summaries in which one summary contains the first and second sentence and the other summary contains the first and third sentence will get the same score. Because, both summaries carry the same utility score i.e., $(5 + 4)$. The RU metric is defined as:

$$RU = \frac{\sum_{j=1}^{n} \delta_j \sum_{i=1}^{N} u_{ij}}{\sum_{j=1}^{n} \epsilon_j \sum_{i=1}^{N} u_{ij}} \tag{42}$$

where u_{ij} is a utility score of sentence j from annotator i, ϵ_j is 1 for the top e sentences according to the sum of utility scores from all judges, otherwise is 0, δ_j is equal to 1 for the top e sentences extracted by the system, otherwise is 0.

Another evaluation metric used in literature [SPKJ07] is cosine similarity computed using the standard formula:

$$\cos(X, Y) = \frac{\sum_i x_i . y_i}{\sqrt{\sum_i (x_i)^2} . \sqrt{\sum_i (y_i)^2}} \tag{43}$$

where X and Y are representations of a system summary and its reference summary based on the vector space model.

An evaluation measurement based on first left singular vector similarity [SJ05; SPKJ07] compares the vectors correspond to the most salient word pattern in the original document and its summary. The measurement uses the angle between the first left singular vectors of the original document

(i.e. SVD performed on the original document) and the summary (i.e. SVD performed on the summary):

$$\cos\varphi = \sum_{i=1}^{n} ue_i \, . \, uf_i, \tag{44}$$

where uf is the first left singular vector of the full text SVD, ue is the first left singular vector of the summary SVD and n is a number of unique terms in the full text.

Hirao et al. [HOYI07] proposed a supervised automatic evaluation method based on a new regression model called the Voted regression model (VRM). The VRM model has two features: (i) model selection based on 'corrected AIC' to avoid multicollinearity, (ii) voting by the selected models to avoid overfitting problem. The VRM model is similar to Averaged Regression Model (ARM), which is proposed by Burnham and Anderson [BA02] but the difference is in the averaging strategy. Hirao et al. confirmed that the VRM model outperforms the ARM model on the selected datasets.

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [LH03] metrics, which were proposed as a variant of BLEU [PRWZ02] are widely used by the DUC and TAC for the evaluation purpose in update summarization task [DA12; WL10; LDS13; ZDXC09; SKLC08; SK08]. These metrics automatically measure the quality of a summary by counting the number of overlapping words between the system-generated summary and the summary created by an human (reference summary). Intuitively, a higher ROUGE score means the system-generated summary and the human-created summary are more similar. Moreover, according to the authors of the ROUGE toolkit [LH03], ROUGE-1 and ROUGE- 2 have high correlation with the human judgments. ROUGE has been accepted as a de facto standard automatic evaluation metric for summarization.

There are different ROUGE measures: ROUGE-N, ROUGE-L, ROUGE-W, ROUGE-S and ROUGE-SU. ROUGE-N is an n-gram recall between a candidate summary and a set of reference summaries. ROUGE-N is computed as follows:

$$\text{ROUGE-N} = \frac{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)}, \tag{45}$$

where n is the length of the n-gram, $gram_n$, $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in a system-generated summary and a set of reference summaries, and $Count(gram_n)$ is the number of n-grams in the reference summaries. ROUGE-1 and ROUGE-2 metrics of ROUGE-N are used here with the length of the n-gram as $n = 1$ and $n = 2$, respectively. The other ROUGE metric used is ROUGE-L, which measures the Longest Common Subsequence (LCS) between a system-generated summary and a reference summary. ROUGE-W is similar to ROUGE-L except it is based on weighted LCS where the weighting function is $f(L) = L^{weight}$, L indicates the length of LCS. Here, the input of the $weight$ is given as $weight = 1.2$ i.e., the metric ROUGE-W-1.2 is calculated. ROUGE-S measures the overlapping of skip-bigrams where the maximum gap length between two words is given as 4 i.e., ROUGE-S_4 is calculated. ROUGE-SU_4 is calculated here to perform an evaluation similar to ROUGE-S, where the maximum gap length between two words is given as 4 with the addition of unigram as a counting unit. It is clear that although ROUGE is a recall-oriented metric but it can be used as a precision-based measure by counting

the percentage of n-grams (in case of computing n-gram precision) in the system-generated summary overlapping with the references. In this way, each of the ROUGE metrics as stated above has three scores (recall, precision and F-score).

The Pyramid method [NP04] was proposed to identify information with the same meaning across different human-generated summaries, called Summarization Content Units (SCU). Each SCU has a weight depending on the number of human assessors that expressed the same information, and these weights follow a specific distribution in order to discriminate important information from less important one. However, the effort to annotate manually all the SCUs is a difficult task. Wang et al. [WLL08] proposed an approach for evaluating summaries based on n-gram co-occurrence statistics, but its main novelty is the use of HowNet for considering the synonyms of a word.

Cornoy et al. [CSO09; CSO11] used new metrics, called Nouveau-ROUGE which are experimented with different ROUGE-based ideas. The most important thing is that the Nouveau-ROUGE includes a measure of novelty in time-biased summarization, more specifically *update* summarization. These scores can be provided with ROUGE scores for sharper estimates with respect to manual evaluation metrics such as overall responsiveness or pyramid scores [NP04]. The Nouveau-ROUGE is proposed using two ROUGE scores, $R_i^{(AB)}$ and $R_i^{(BB)}$ $(i = 1, 2, SU4, \ldots)$, in a three parameter model in order to predict automatic scores for two manual evaluation metrics, pyramid and overall responsiveness. $R_i^{(AB)}$ compares each update summary to the human-generated summaries and $R_i^{(BB)}$ compares each update summary with the base summaries (i.e.the original summaries). It is defined as follows:

$$N_i = \alpha_{i,0} + \alpha_{i,1} R_i^{(AB)} + \alpha_{i,2} R_i^{(BB)} \tag{46}$$

where the α parameters are determined using robust linear regression on the given datasets for responsiveness and pyramid scores.

Recently, Tran et al. [BAQ13] designed another framework for evaluating the similarity of the generated timeline summaries with the manually created ones. The authors used Mean Average Precision (MAP) metric to evaluate the relevance of the dates which are determined by different date selection methods. Let the set of relevant dates for timeline summaries be $d_1, \ldots, d_n$ and R_k be the ranked lists of dates from d_q to d_k then the metric is defined as follows:

$$MAP = \frac{1}{n} \sum_{k=1}^{n} Precision(R_k) \tag{47}$$

After date selection, they used ROUGE metrics to measure the similarity of the generated timeline summaries with the manually created ones and in comparison with other methods.

2.4.2 User Evaluation

While an automatic evaluation can provide up to a certain level to estimate the informativeness of a generated summary, it does not consider some important aspects such as the coherence, readability or the overall responsiveness. To evaluate such aspects further, a manual evaluation is required. In

the literature, a quite standard approach for manual evaluation is made through pairwise comparison [HV09; LL10; CHT11; DA12].

In this approach, human evaluators are presented with three random summaries, a pair of summaries generated by two systems and a corresponding reference summary. Later, they are asked to mark the better summary between the given pair of system-generated summaries according to some criteria such as — *non-redundancy* (which summary repeats less the same information), *coherence* (which summary has more consistency among sentences), *focus* (which summary contains less irrelevant details), *overall responsiveness* (which summary is best overall both in terms of content and fluency), and so on. The evaluation judgments in frequencies will record that how many times 'system A' is better than 'system B', or 'system B' is better than 'system A', or there is a tie between 'system A' and 'system B'. From the records, the results can be shown that the summaries from 'system A' are better than 'system B' or vice versa based on some statistical tests (for example, a paired statistical t-test).

The other way of evaluating summaries with direct user feedback is to use the crowdsourcing [How06] in order to design an experiment and collect the user feedback. To improve the quality of the results, sometimes it is needed to set up a strategy either by defining some gold tasks as ground truth or requesting multiple judgments for each task [NRD11; TAH15]. At the end, the results can be shown in a similar way like in previous approach that one system is better than the other based on some statistical tests.

In order to evaluate the system generated timelines against manually constructed timelines, Chieu et al. [CL04] carried out user evaluation in three phases. They used person names, namely, the eight leaders of the countries in G8 from January to June 2002 from the English Gigaword corpus. In the first phase, four evaluators were asked to construct timelines of ten sentences for the queries assigned to them. Each evaluator was assigned 4 queries out of 8 queries, so that there are exactly 2 manually constructed timelines for each query. In the second phase, for each of the 8 queries the evaluators were given 4 timelines (2 manually constructed timelines and 2 system generated timelines) without telling which of the timelines were manually constructed and asked to rate the timeline on a scale of 1 to 6 based on the four criteria 'Representative of media coverage', 'Comprehensibility', 'Conciseness' and 'Importance'.

2.5 SUMMARY

In this chapter we have made an effort to give a comprehensive overview of the most prominent extractive methods for both temporal and classical text summarization. The aim of this chapter is to present the state-of-the-art in summarization techniques focusing, specially in the connection to classical approaches while contrasting time-biased approaches in terms of how they represent the input, score sentences and select the summary. In other words, the important part of this survey deals with the current state-of-the-art of classical summarization techniques, where it is shown how it has been adapted to the new requirements. We have also highlighted a study of the approaches for existing evaluation frameworks and metrics, as well as new proposed frameworks that have emerged concerning the automatic evaluation of time-biased summaries. Temporal summarization is an emerging area, where there is still a lot of room for improvement, specially

in deriving representation from the input text or taking into account contextual information that can help to determine sentence selection. Although the aforementioned evaluation frameworks and metrics really help to assess automatic summaries, there are shortcomings with respect to the quality evaluation that remain still unsolved.

3 SUMMARIZATION OF CHANGES USING LDA MODEL

The summarization of changes can be described as follows: given an evolving document collection and a temporal period, generate a summary of significant alterations made to the collection of documents during that period. This chapter proposes different approaches to generate the summaries of changes using extractive summarization techniques. First, individual terms are scored and then this information is used to rank and select sentences to produce the final summary. A system based on LDA is used to find the hidden topic structures of changes. The purpose of using the LDA model is to identify separate topics where the changed terms from each topic are likely to carry at least one significant change. The different approaches are then compared with the previous work in this area.

A collection of articles from Wikipedia, including their revision history, is used to evaluate the proposed system. For each article, a temporal interval and a reference summary from the article's content are selected manually. The articles and intervals in which a significant event occurred are carefully selected. The summaries produced by each of the approaches are evaluated comparatively to the manual summaries using ROUGE metrics. It is observed that the approach using the LDA-based approach outperforms all other approaches. Statistical tests reveal that the differences in ROUGE scores for the LDA-based approach is statistically significant at 99% over baseline.

Publication

This chapter is based on the following publication:

- Kar, M., Nunes, S., & Ribeiro, C. (2015). Summarization of changes in dynamic text collections using Latent Dirichlet Allocation model. Information Processing & Management, 51(6), 809–833. [KNR15]

3.1 INTRODUCTION

In the area of IR, it is recognized that retrieval from dynamic text collections on the web brings several new research challenges [ACMSa12]. Web pages are continually added, removed, or edited, resulting in active collections of documents that are always being modified. It is common to observe a high rate of changes as a consequence of the occurrence of real-world events. However, there are also modifications to documents which are generic, namely those resulting from minor revisions or additions/modifications of outdated information. The automatic summarization of changes in dynamic text collections gains relevance in this context. The goal is to obtain a summary that describes the most significant changes made to a document during a given period. In other words, the idea is to have a summary of the revisions made to a document over a specific period of time.

This study uses a collection of articles from Wikipedia where data is dynamic by nature. One of the most important challenges is the diversity of intentions of the users while updating an article. When an event draws attention to a given Wikipedia article, it is possible to verify two types of positive revisions: revisions related to the specific event and revisions to generally update the whole article [NRD08]. The system proposed here should present the significant changes as a final summary by filtering the general updates for a given time period. In general, there are two approaches to automatic summarization: extractive summarization and abstractive summarization [JM00; KM02]. This chapter focuses on extractive summarization, proposing different approaches for executing this task.

The rest of the paper is organized as follows: Section 3.2 describes the basic architecture of the proposed system. Section 3.3 defines the sentence-scoring measurements using different approaches and presents a simple similarity measurement to identify the unique sentences, discarding the redundant ones. Section 3.4 describes the experimental details and Section 3.5 presents the analysis of experimental results. Finally, Section 3.6 provides summarizes the chapter and adds some final remarks, suggesting future research directions.

3.2 SYSTEM ARCHITECTURE

This section describes the overall architecture and methodologies for the task of summarizing changes. Various notations which are used in the following sections are also introduced. When a user gives a time interval to an entity of interest, the time range actually determines how many versions of the document are used for detecting the changes. To incorporate the temporal dimension of documents, it is considered that for any article $\mathcal{A}$ there are T document versions represented as $A = \{\mathrm{rev}_1, \mathrm{rev}_2, \ldots, \mathrm{rev}_T\}$ in the given time range. rev_1 is the most recent document version and rev_T is the oldest document version in the given time frame for the article $\mathcal{A}$.

The conceptual architecture of our system is displayed in Figure 3. It shows the information flow at each step. The input of the diff process is a set of document versions for the article $\mathcal{A}$ and the outputs are a set of word-diff's and a set of block-diff's. The set of block-diff's is the input for the sentence extraction process, which is shown in the left hand side of the figure and the set of word-diff's is the input for the feature extraction process, shown in the right hand side. The feature extraction process generates the feature files for each approach separately and provides them for the purpose of generating the term scores. The set of sentences extracted with the sentence extraction process is used by each approach to generate the summary.

The system extracts the differences (diff) from the collection of T document versions by comparing the consecutive versions starting from rev_1. The last extracted difference is between rev_{T-1} and rev_T document versions. Therefore, at the end of the diff process, the total number of extracted differences is $T-1$ for T document versions. The set of $T-1$ differences is defined as $D = \{\mathrm{diff}_1, \mathrm{diff}_2, \ldots, \mathrm{diff}_{T-1}\}$.

In practice, the changes in document versions can be made in three ways: insertion, modification or deletion. Based on this, the set D is divided into two categories: the changes made by insertions and modifications are put into one category, and the changes made by deletions are put into another. In order to differentiate the two categories of changes, $D^{(ins)}$ is the sub-

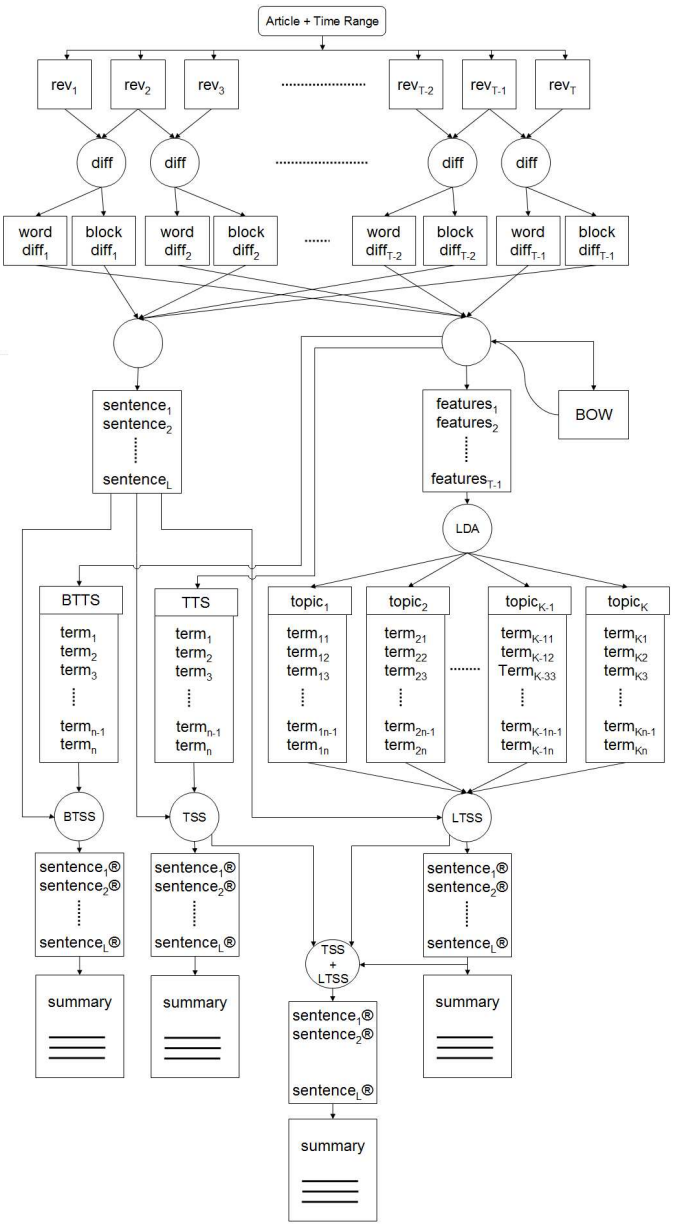

Figure 3: System architecture

set of D containing the differences which occurred due to insertions and modifications whereas $D^{(del)}$ is the subset of D containing the differences which occurred due to deletions. Therefore, the set D can be written as $D = D^{(ins)} \cup D^{(del)}$.

The set of differences caused by insertions and modifications ($D^{(ins)}$) and the set of differences caused by deletions ($D^{(del)}$) are processed further in two modes: word mode which is denoted as word-diff and block mode which is denoted as block-diff. In word-diff, the diff process extracts the changed words by comparing the consecutive document versions on the word basis [Gooa] while in block-diff, the diff process extracts the changed paragraphs [Jav]. $D^{(ins)}$ in word mode is represented as

$$D^{(ins)}_{word} = \{\text{word-diff}^{(ins)}_1, \text{word-diff}^{(ins)}_2, \ldots, \text{word-diff}^{(ins)}_{T-1}\}$$

and $D^{(ins)}$ in block mode is represented as

$$D^{(ins)}_{block} = \{\text{block-diff}^{(ins)}_1, \text{block-diff}^{(ins)}_2, \ldots, \text{block-diff}^{(ins)}_{T-1}\}.$$

Similarly, $D^{(del)}$ in word mode and $D^{(del)}$ in block mode are denoted as $D^{(del)}_{word}$ and $D^{(del)}_{block}$ respectively.

After obtaining the sets of all extracted differences, namely, $D^{(ins)}_{word}$, $D^{(ins)}_{block}$, $D^{(del)}_{word}$ and $D^{(del)}_{block}$, four main approaches are proposed for the task. The basic framework for all approaches starts by scoring the words/terms and then ranking the sentences on the basis of those words scores. In the sentence ranking process, a set of sentences is provided to each of the approaches as input. This set of sentences S is built through a sentence extraction process from all the $\text{block-diff}_i \in D^{(ins)}_{block}$. Then, the sentence ranking process assigns a score to each of the sentences in S. The sentence score is basically calculated by the sum of the scores of all its terms, divided by the total number of terms. Each of the approaches calculates the term's score differently. After the sentence ranking process, a set of sentences is obtained of the form $\{(sentence_i, sentence^{(s)}_i), sentence_i \in S\}$, where $sentence_i$ refers to the i-th sentence itself from the set S, whereas $sentence^{(s)}_i$ is the corresponding score of $sentence_i$. Then, all the sentences are ranked in descending order according to their scores. Finally, from each approach, the top ranked sentences are presented as a summary.

In the first approach, term scores are computed using the scoring function which is adapted from an existing work [JBI04], and then the score of each sentence is computed using those term scores. The sets A, $D^{(ins)}_{word}$ and $D^{(del)}_{word}$ are used to generate scores for terms, whereas $D^{(ins)}_{block}$ is used to build a set of sentences. This method is considered the baseline approach in the task proposed here. The score of a term generated by this approach is called baseline temporal term score (BTTS) and the score of a sentence using BTTS is called baseline temporal sentence score (BTSS).

A different scoring function is proposed in the second approach. Here, the focus is on the temporal aspects to be incorporated into the scoring function. The two sets $D^{(ins)}_{word}$ and $D^{(del)}_{word}$ are used to rank words while $D^{(ins)}_{block}$ is used to rank sentences. The basic idea behind this scoring function is that the word which occurs frequently in $D^{(ins)}_{word}$ will obtain a higher score but if the word occurs in $D^{(del)}_{word}$, it will get a lower score. This means that the score of a word is higher as it is inserted in the document versions more frequently; simultaneously the score decreases as the word is deleted from the document versions. The score of a term generated by this approach is called temporal term score (TTS) and the score of a sentence using TTS is called temporal sentence score (TSS).

In the third approach, words scores are generated via LDA model. The set, $D^{(ins)}_{word}$ is used to generate a feature file for the LDA model. The feature file has a total of $M = T-1$ feature vectors where each feature vector $\mathbf{w}_i$ consists of a sequence of words $(w_1, w_2, \ldots w_{j-1}, w_j, w_{j+1}, \ldots)$ generated from $\text{word-diff}_i \in D^{(ins)}_{word}$. Here, each word w_j in the sequence belongs to any of the words from a set of V distinct words $BOW = \{w^{(1)}, w^{(2)}, \ldots w^{(V)}\}$, which is directly created from $D^{(ins)}_{word}$ as well. This feature file is given as an input file in the LDA model. The LDA model generally tries to backtrack from the documents to find out a set of latent topics that consist of terms with certain probabilities. Here, the LDA model is used to figure out the im-

portant changed terms assigned to different latent topics. It is assumed that each latent topic corresponds to at least one significant change, and different changes can be interpreted by different latent topics. The words with the corresponding scores for each latent topic are generated as an output via the LDA model. Let there be K number of latent topics $\{z_i : i = 1, 2, \ldots, K\}$ and each topic is described by a set of terms, each word is associated with a score. Consider the set of terms of the form $\{(term_{ij}, term_{ij}^{(s)}), term_{ij} \in BOW\}$, where $term_{ij}$ is the j-th word of topic z_i and $term_{ij}^{(s)}$ is the corresponding score of $term_{ij}$, produced by the LDA model as an output. In general, any term is denoted as w_j but while describing the scoring function $term_{.j}$ is introduced instead of w_j to make it conventional. Apart from calculating the scores of sentences, the label is also assigned to each sentence based on the terms which belong to a particular topic. One sentence belongs to z_i be decided, if z_i gives the highest score to that sentence. Therefore, in this approach, the ranking process generates a set of sentences of the form $\{(sentence_i, sentence_i^{(s)}, z), sentence_i \in S\}$, where $sentence_i$ refers to the i-th sentence itself from the set S, $sentence_i^{(s)}$ is the corresponding score of $sentence_i$ and z is the label to identify to which topic the sentence belongs. The score of a term generated by the LDA model is called latent topic term score (LTTS) and the score of a sentence using LTTS is called latent topic sentence score (LTSS). The system ranks the sentences in descending order based on the sentences scores. The top ranked sentences are presented to convey the main changes in the defined period.

The goal is to develop a system that can provide a summary with two main characteristics: i) the summary should only describe the information that has been changed, and ii) at the same time, the summary should contain only the significant changes among all other general changes made within the given time period. Since, we consider $D_{word}^{(ins)}$ for computing the words/terms scores, by default, the first characteristic is incorporated into all the approaches. However, to find out the most significant changed words and to assign higher scores to them, different scoring functions are used in different approaches. The main difference of the third approach comparatively to the other two is that it can separate groups of related changed words, in which each group is likely to carry at least one significant change. In the third approach, however, only the changes caused by insertions and modifications ($D_{word}^{(ins)}$) are considered while scoring the terms. Nevertheless, the changes caused by deletions ($D_{word}^{(del)}$) can also play an important role in temporal aspects. In order to indirectly incorporate the changes caused by deletions in the LDA, another approach is introduced which is a combination of the second and third approaches. Here, the top ranked sentences generated by the LDA are re-ranked with a combination of LTSS and TSS. Figure 3 describes the outline of the four sentence scoring measurements in different approaches, where the first three sentence scoring measurements, namely BTSS, TSS and LTSS are independent whereas the fourth is a combination of the second and third (TSS and LTSS). Each sentence scoring measurement is described in detail in the following sections.

To evaluate the system, a collection of articles from Wikipedia, including their revision history, is used. For each article, a temporal interval and a reference summary are manually selected from the article's contents. The articles and intervals in which significant events occurred are carefully selected. To construct a reference summary for a given time range, a set of sentences are previously selected and extracted such that these sentences

can describe the exact significant change. The problem of having multiple reference summaries for that time period does not arise for two reasons. First, in general, for a reference summary the sentences are extracted from the latest version of that specified Wikipedia article in the given time period instead of writing a reference summary manually. Second, the significant change is so prominent, it is easy to select those sentences as a reference summary. The summaries produced by each of the approaches are evaluated comparatively to the reference summaries using ROUGE metrics. The most important sentences are selected manually for a reference summary and since this reference summary is provided for comparison against the system generated summaries, ROUGE scores can express whether the best sentences are peaked or not by different approaches. Intuitively, the higher the ROUGE scores, the better sentences are selected using that approach.

3.3 SENTENCE RANKING

Previous summarization tasks usually focused either on a single document or on a set of documents from a static collection on a given topic. However, document collections change dynamically when the topic evolves over time, as new documents are continuously added, modified or deleted. These changes usually bring the new information to the topic, which poses new challenges to the sentence ranking process when summarizing a dynamic collection of documents [ACMSa12].

The objective of the sentence ranking process is to calculate scores for all sentences so that they can be arranged in descending order of their scores. Usually, the scores for all sentences are calculated in such a way that the most significant sentences are likely to obtain higher scores. The two main steps associated with the sentence ranking process are: i) calculating each term's score in a sentence and ii) calculating each sentence's score using the scores of these terms. This section describes different sentence scoring measurements obtained with the proposed approaches.

3.3.1 Approach-I: Baseline Temporal Sentence Score (BTSS)

In the first approach, each term's score is generated using the following scoring function:

$$\begin{aligned} BTTS(term_j) = (1 + \frac{\sum_{r=1}^{N_{doc}} [\frac{n_{rc}}{N_{rc}+1} - \alpha \times \frac{n_{rs}}{N_{rs}+1}]}{N_{doc}}) \times \\ \exp(\frac{n_{jcp}}{N_{cdoc}+1} - \alpha \times \frac{n_{jsp}}{N_{sdoc}+1}) \end{aligned} \tag{48}$$

where $BTTS(term_j)$ is the baseline temporal term score (BTTS) for the j-th term in our system and the meanings of individual symbols are described in Table 2. This scoring function was introduced in the *ChangeSummarizer* system [JBI04], which periodically monitors a web collection in search for new changes and generates their summary related to a specific topic. To the best of our knowledge, the only explicit reference to the idea put forward here is the *ChangeSummarizer* system [JBI04]. However, contrarily to the theory presented here, the information addressed in the *ChangeSummarizer* system [JBI04] is limited to "recent, important changes". Hence, the scoring function used in the *ChangeSummarizer* system [JBI04] is adapted here as the

Table 2: Explanation of the symbols used in Equation 48 for both the systems

Symbols	Explanation of the symbols used in the system proposed	Explanation of the symbols used in the *ChangeSummarizer* system
N_{doc}	Number of versions of an article within a time period	Number of pages in the web collection
N_{sdoc}	Number of static versions of an article within a time period	Number of static pages in the web collection
N_{cdoc}	Number of changed versions of an article within a time period	Number of changed pages in the web collection
n_{jsp}	Number of versions of an article where static parts contain term j	Number of pages in the web collection where static parts contain term j
n_{jcp}	Number of versions of an article where changed parts contain term j	Number of pages in the web collection where changed parts contain term j
N_{rs}	Number of static terms in revision r of an article	Number of static terms in page r in the web collection
N_{rc}	Number of changed terms in revision r of an article	Number of changed terms in page r in the web collection
n_{js}	Number of term j in static part of revision r of an article	Number of term j in static part of page r in the web collection
n_{jc}	Number of term j in changed part of revision r of an article	Number of term j in changed part of page r in the web collection

baseline, but the meanings of the symbols used in Equation 48 are slightly changed. Table 2 describes the basic differences in the meaning of the symbols between the *ChangeSummarizer* system and the proposed system. The motivation of adapting Equation 48 as the baseline is to compare against a system which is already in the literature for a similar task.

The main difference between both systems is that we consider an article's different versions instead of using individual web pages devoted to a common topic. In the summarization of changes system, it is assumed that the number of static versions of an article within a time period, N_{sdoc} is equal to zero and the number of changed versions of an article within a time period, N_{cdoc} is equal to the total number of document versions made to an article within that time period. If there is a total T number of document versions of an article in the given time period, then the values for the variables are $N_{sdoc} = 0$ and $N_{cdoc} = T$. The changed and the static terms are figured out by comparing between consecutive versions of an article. For example, the changed and the static terms for rev_{T-1} version are obtained by comparing rev_{T-1} and rev_T consecutive versions for that article. After removing stop words obtained from that comparison, the fixed terms are considered the static terms for rev_{T-1} version whereas the terms which are either added, modified or deleted in rev_{T-1} version are the changed terms. The sets, A, $D^{(ins)}_{word}$ and $D^{(del)}_{word}$ are used to obtain the changed and static terms separately for all the versions of an article in the given time period.

The basic idea of this term scoring function is to give higher scores to the terms that appear more often in the changed parts of an article's different versions but occur rarely in the static parts of those versions. In Equation 48, the first part gives higher scores to the popular terms that occurred in the changed parts of the article's different versions, but not in the static parts. The second part of Equation 48 has another motivation. The terms appearing frequently in the changed parts (i.e. popular terms in changed parts) may have low semantic values for a particular topic. Therefore, the second part of Equation 48 tries to assign higher scores to those changed terms which are less common or typical to a specific domain. The parameter α is used to control the scoring of these changed uncommon terms. α ranges between 0 and 1.

It is stated earlier in Section 3.2 that the basic framework for all approaches starts by scoring the words/terms and then ranks the sentences on the basis of those words scores. In the sentence ranking process, a set of sentences is provided to each of the approaches as input. This set of sentences S is built through a sentence extraction process from all the $\text{block-diff}_i \in D^{(ins)}_{block}$. Then, the sentence ranking process assigns a score to each of the sentences in S. The sentence score is basically calculated by the sum of the scores of all its terms, divided by the total number of terms after excluding the stop words. Thus, the sentence ranking process generates a set of sentences of the form $\{(sentence_i, sentence_i^{(s)}), sentence_i \in S\}$, where $sentence_i$ refers to the i-th sentence from the set S and $sentence_i^{(s)}$ is the corresponding score of $sentence_i$. Here, $sentence_i^{(s)}$ is computed with the $BTSS(sentence_i)$ function, the baseline temporal sentence score (BTSS) for $sentence_i$. $BTSS(sentence_i)$ is defined as

$$BTSS(sentence_i) = \frac{\sum_j BTTS(term_j)}{N(sentence_i)}, term_j \in sentence_i \tag{49}$$

where $BTTS(term_j)$ for the j-th term is calculated using Equation 48 and $N(sentence_i)$ denotes the total number of terms in $sentence_i$ after excluding the stop words. Once the sentence ranking process assigns a score to each of the sentences, the sentences are ranked in a descending order of their scores. Finally the top ranked sentences are selected to produce as a summary.

3.3.2 Approach-II: Temporal Sentence Score (TSS)

In the second approach, a different scoring function is introduced. Here, the emphasis is on the temporal aspects to be incorporated into the scoring function. The aim is to extract the significant terms from a set of document versions which bring changed and novel information into an article. Research has been done on term weighting using the time impact on changes occurring either in the collection [Efr10] or in an individual document [NRD11]. Instead of directly taking into account the periods of time in which a term occurs throughout an article's revision history, the score of a term is calculated here by considering the joint probabilities of both the insertion and deletion events occurring in a set of document versions.

The basic idea behind the second approach is that a term which has been revised frequently in a set of document versions should be more important because it could reflect a significant change. However, simultaneously, if the term is deleted frequently in the set of document versions, the term might not be the important term to present any significant change and as a result that term should obtain less importance. In particular, a term which occurs frequently in $D^{(ins)}_{word}$ will obtain a higher score; consequently, if it occurs less frequently in $D^{(del)}_{word}$, its score will be lower. The probability of $term_j$ occurring as a result of insertions is calculated as

$$P^{(ins)}(term_j) = \frac{tf(term_j, D^{(ins)}_{word})}{\sum_j tf(term_j, D^{(ins)}_{word})} \tag{50}$$

where, $tf(term_j, D^{(ins)}_{word})$ is the frequency of j-th term ($term_j$) in the set $D^{(ins)}_{word}$. Similarly, the probability of $term_j$ occurring as a result of deletions is calculated as

$$P^{(del)}(term_j) = \frac{tf(term_j, D^{(del)}_{word})}{\sum_j tf(term_j, D^{(del)}_{word})} \tag{51}$$

where, $tf(term_j, D^{(del)}_{word})$ is the frequency of j-th term ($term_j$) in the set $D^{(del)}_{word}$. Finally, the temporal term score (TTS) for $term_j$ is calculated as

$$TTS(term_j) = P^{(ins)}(term_j) \times (1 - P^{(del)}(term_j)) \tag{52}$$

The sentence extraction process builds a set of sentences S, extracted from all the $\text{block-diff}_i \in D^{(ins)}_{block}$. The sentence ranking process assigns a score to each sentence, which is calculated with the sum of the scores of all terms, divided by the number of terms. Thus, the sentence ranking process generates a set of sentences of the form $\{(sentence_i, sentence^{(s)}_i), sentence_i \in S\}$, where $sentence^{(s)}_i$ is the score of $sentence_i$. Here, $sentence^{(s)}_i$ is computed with the $TSS(sentence_i)$ function, the temporal sentence score (TSS) for $sentence_i$. $TSS(sentence_i)$ is defined as

$$TSS(sentence_i) = \frac{\sum_j TTS(term_j)}{N(sentence_i)}, term_j \in sentence_i \tag{53}$$

where $TTS(term_j)$ is calculated for the j-th term using Equation 52 and $N(sentence_i)$ is the total number of terms in $sentence_i$ after excluding the stop words. Like in the previous approach, the top ranked sentences are selected at the final step to produce a summary.

3.3.3 Approach-III: Latent Topic Sentence Score (LTSS) using Latent Dirichlet Allocation Model

In this approach, the term score is generated using the LDA model. Before defining the latent topic sentence score (LTSS), it is necessary to describe the theoretical background of the LDA model. Although the basic concepts of the LDA model and its learning paradigm are already available in the literature, introducing briefly the theoretical part of the LDA model makes it easier to explain how LTSS can be computed using the LDA model.

Latent Dirichlet Allocation (LDA)

LDA is a statistical model that tries to capture the latent topics in a collection of documents. LDA was first introduced by David Blei [BNJ03]. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. One important assumption about the LDA generative model is that the number of topics is known in advance. Before describing the LDA model, formally, the following definitions are required:

1. A *word* is the basic unit of discrete data, defined to be an item from a vocabulary of size V denoted by $BOW = \{w^{(1)}, w^{(2)}, \ldots, w^{(V)}\}$.

2. A *document* (d_i) is a sequence of N words denoted by $\mathbf{w} = (w_1, w_2, \ldots, w_N)$, where each $w_j, j = 1, 2, \ldots N$ belongs to any of the V vocabulary words from the set BOW.

3. A *corpus* is a collection of M documents denoted by $A = \{d_1, d_2, \ldots, d_M\} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M\}$

In the LDA model, for each document (d_i), there is a multinomial distribution over K topics $\{z_i = j : j = 1, 2, \ldots, K\}$, with parameters $\theta^{(d_i)}$, so for a word in document d_i, $P(z_i = j) = Multinomial(\theta_j^{(d_i)})$. The j-th topic ($z_i = j$) is represented further by a multinomial distribution over the set of vocabulary words (BOW), with the parameters $\phi^{(j)}$, so $P(w_i|z_i = j) = Multinomial(\phi_{w_i}^{(j)})$. To make the predictions about new documents, it is assumed a prior distribution on the parameters $\theta^{(d_i)}$. It is well known that the Beta distribution is the conjugate prior of the Bernoulli distribution and the Dirichlet distribution is the conjugate prior of the multinomial distribution. Therefore, for $\theta^{(d_i)}$ a Dirichlet prior with parameters α, i.e., $\theta^{(d_i)} \sim Dir(\alpha)$ is chosen. Similarly for $\phi^{(j)}$, a Dirichlet with parameters β is chosen as prior i.e., $\phi^{(j)} \sim Dir(\beta)$. For K-dimensional Dirichlet random variable θ ($\theta_i \geqslant 0$, $\sum_{i=1}^{K} \theta_i = 1$), the probability density function is defined as

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{K} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \ldots \theta_K^{\alpha_K - 1} \tag{54}$$

where α is a K-vector with components $\alpha_i > 0$, K is the number of hidden topics and $\Gamma(x)$ is the Gamma function. Then, the distribution over words for any document is modeled as the mixture

$$P(w_i) = \sum_{j=1}^{k} P(w_i|z_i = j) P(z_i = j) \tag{55}$$

In the learning stage, the probability $P(z_i = j) = Multinomial(\theta_j^{(d_i)})$ is computed in terms of $\theta_{M \times K}$ matrix and $P(w_i|z_i = j) = Multinomial(\phi_{w_i}^{(j)})$ in terms of $\phi_{K \times V}$ matrix. In order to generate topic wise word score, $\phi_{K \times V}$ matrix is used and then the words are sorted in descending order of their scores for each topic.

Learning and Inference

So far motivations and intuitions for LDA have been described. One of the key challenges associated with LDA is the inference problem, in particular computing the posterior probabilities for the hidden variables given a document. Variational EM algorithm [BNJ03] is introduced for obtaining approximate maximum-likelihood estimates for $\phi^{(j)}$ and the hyper-parameters of the prior on $\theta^{(d_i)}$. Gibbs sampling [Gri02] is another method where a symmetric $Dir(\alpha)$ prior on $\theta^{(d_i)}$ for all documents, and a symmetric $Dir(\beta)$ prior on $\phi^{(j)}$ for all topics are considered in the model and Markov Chain Monte Carlo technique is used for the inference. In this paper, Gibbs sampling technique, which is comparatively faster than other existing algorithms is used to infer the model parameters for the given dataset. The plate diagram of the LDA model is shown in Figure 4.
In Gibbs sampling, the next state is reached by sequentially sampling all variables from their distribution depending on the current values of all other variables and the data. One advantage of Gibbs sampler is that it deals with

the subset of the words seen so far rather than the whole data. So, the conditional posterior distribution for j-th topic, $z_i = j$ is given by

$$P(z_i = j|\mathbf{z}_{-i}, \mathbf{w}) \propto P(w_i|z_i = j, \mathbf{z}_{-i}, \mathbf{w}_{-i})P(z_i = j|\mathbf{z}_{-i}) \tag{56}$$

where, $\mathbf{z}_{-i}$ is the assignment of all other topics except topic j. From Equation 56, the first term on the right hand side can be written as

$$P(w_i|z_i = j, \mathbf{z}_{-i}, \mathbf{w}_{-i}) = \int P(w_i|z_i = j, \phi^{(j)})P(\phi^{(j)}|\mathbf{z}_{-i}, \mathbf{w}_{-i})d\phi^{(j)} \tag{57}$$

For a multinomial-Dirichlet model, Equation 57 gives the predictive distribution when a new word appears. The first term $P(w_i|z_i = j, \phi^{(j)})$ of the integral is equal to $\phi^{(j)}_{w_i}$, the multinomial distribution over words associated with topic j. The second term of the integral can be written from Bayes' rule

$$P(\phi^{(j)}|\mathbf{z}_{-i}, \mathbf{w}_{-i}) \propto P(\mathbf{w}_{-i}|\phi^{(j)}, \mathbf{z}_{-i})P(\phi^{(j)}) \tag{58}$$

Since, $P(\phi^{(j)})$ follows $Dir(\beta)$ as prior and conjugate to $P(\mathbf{w}_{-i}|\phi^{(j)}, \mathbf{z}_{-i})$ multinomial, then according to the definition of conjugate, the posterior distribution $P(\phi^{(j)}|\mathbf{z}_{-i}, \mathbf{w}_{-i})$ will also follow $Dir(\beta + n^{(w_i)}_{-i,j})$, where $n^{(w_i)}_{-i,j}$ is the number of occurrences of word w_i assigned to topic j, not including the current observing word. So, the integral of Equation 57 comes as:

$$\begin{aligned} P(w_i|z_i = j, \mathbf{z}_{-i}, \mathbf{w}_{-i}) &= \int \phi^{(j)}_{w_i} P(\phi^{(j)}|\mathbf{z}_{-i}, \mathbf{w}_{-i})d\phi^{(j)} \\ &= E(\phi^{(j)}_{w_i}|z_i = j, \mathbf{z}_{-i}, \mathbf{w}_{-i}) \\ &= \frac{n^{(w_i)}_{-i,j} + \beta}{n^{(.)}_{-i,j} + V\beta} \end{aligned} \tag{59}$$

where, $n^{(.)}_{-i,j}$ is the total number of words assigned to topic j, not including the current observing word. Similarly, from Equation 56, the second term on the right hand side can be written as

$$\begin{aligned} P(z_i = j|\mathbf{z}_{-i}) &= \int P(z_i = j|\theta^{(d_i)})P(\theta^{(d_i)}|\mathbf{z}_{-i})d\theta^{(d_i)} \\ &= \frac{n^{(d_i)}_{-i,j} + \alpha}{n^{(d_i)}_{-i,.} + K\alpha} \end{aligned} \tag{60}$$

where, $n^{(d_i)}_{-i,j}$ is the number of words from document d_i assigned to topic j, not including the current observing word and $n^{(d_i)}_{-i,.}$ is the total number of words in document d_i, not including the current observing word. Combining the results of Equations 59 and 60 in Equation 56 leads to

$$P(z_i = j|\mathbf{z}_{-i}, \mathbf{w}) \propto \frac{n^{(w_i)}_{-i,j} + \beta}{n^{(.)}_{-i,j} + V\beta} \frac{n^{(d_i)}_{-i,j} + \alpha}{n^{(d_i)}_{-i,.} + K\alpha} \tag{61}$$

*Latent topic sentence score (*LTSS*)*

It has been previously mentioned that the LDA model returns a $\phi_{K\times V}$ matrix which describes the probability of terms (w_i) assuming that they belong to a specific topic ($z_i = j$) i.e., $P(w_i|z_i = j)$. The goal is to discover different significant changes in terms of different latent topics within a set of revisions on an article. This will make a cluster of related terms which reflect

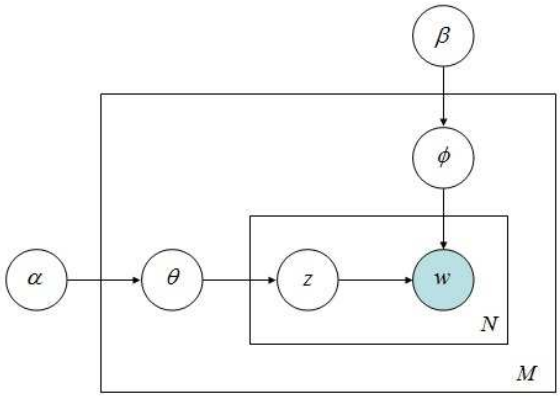

Figure 4: LDA plate diagram

the same kind of change. The number of latent topics, which corresponds to the number of different changes is selected beforehand depending on the choice of K. Even though the conventional notations $z_i = j$ and w_i are used in the theoretical descriptions, in order to make them consistent throughout the paper, $z_i = j$ to $z_i = i$ and w_i to w_j are flipped. Moreover, w_j is replaced with term_j while describing the scoring function. More specifically, when term_j belongs to a particular topic, say $z_i = i$, it is used as term_{ij}. Let be the set of terms of the form $\{(\mathrm{term}_{ij}, \mathrm{term}_{ij}^{(s)}), \mathrm{term}_{ij} \in BOW\}$, where $\mathrm{term}_{ij} \in BOW$ is the j-th word of topic z_i and $\mathrm{term}_{ij}^{(s)}$ is the corresponding score of term_{ij} which is generated via LDA i.e., $\mathrm{term}_{ij}^{(s)} \simeq \phi_{ij}$. The latent topic term score (LTTS) for $\mathrm{term}_{ij} \in z_i$ is computed as

$$\mathrm{LTTS}(\mathrm{term}_{ij}) = \mathrm{term}_{ij}^{(s)} \times \mathrm{itf}(\mathrm{term}_{ij}), \tag{62}$$
$$\mathrm{itf}(\mathrm{term}_{ij}) = \log \frac{(K+1)}{\mathrm{tf}(\mathrm{term}_{ij})}$$

where topic frequency (tf) returns the count of a term's presence in different topics and inverse topic frequency (itf) accounts for a higher weight to the terms which are not common in different topics. Inverse topic frequency (itf) follows a concept similar to the inverse document frequency (idf) which is well known in the literature. This happens because the terms appearing in almost all the topics may have low semantic values for an article. In Equation 62, the numerator of log function is increased by 1 i.e., from K to $(K+1)$, such that when the term appears in all K topics, at least a minimum weight is assigned instead of zero.

Similarly, the latent topic sentence score (LTSS) for $\mathrm{sentence}_{ik} \in z_i$ is calculated as

$$\mathrm{LTSS}(\mathrm{sentence}_{ik}) = \frac{\sum_j \mathrm{LTTS}(\mathrm{term}_{ij})}{N(\mathrm{sentence}_{ik})}, \mathrm{term}_{ij} \in \mathrm{sentence}_{ik} \tag{63}$$

Finally, regardless of latent topic z_i, the latent topic sentence score (LTSS) for $\mathrm{sentence}_k$ is calculated

$$\mathrm{LTSS}(\mathrm{sentence}_k) = \max_i \mathrm{LTSS}(\mathrm{sentence}_{ik}) \tag{64}$$

and the sentence is assigned with a topic label which will provide the maximum score to that sentence i.e., $z_i = \arg\max_i \mathrm{LTSS}(\mathrm{sentence}_{ik})$. Therefore,

in this approach, the ranking process generates a set of sentences of the form $\{(sentence_i, sentence_i^{(s)}, z), sentence_i \in S\}$, where $sentence_i^{(s)}$ is the corresponding score of $sentence_i$ and z is the label to identify which topic the sentence belongs to. Once, the sentence ranking process assigns a score to each sentence, then all the sentences are ranked in a descending order of their scores. Finally the top ranked sentences are chosen to produce as a summary.

3.3.4 Approach-IV: A Combination of Temporal Sentence Score (TSS) & Latent Topic Sentence Score (LTSS)

When generating score of a term using the third approach, only the changes caused by insertions and modifications ($D_{word}^{(ins)}$) are considered. However, the changes caused by deletions ($D_{word}^{(del)}$) can also play an important role in temporal aspects. In order to incorporate all kinds of changes, another approach is introduced which is a combination of the second and third approaches, where the top ranked sentences of the form

$$\{(sentence_i, sentence_i^{(s)}, z), sentence_i \in S\}$$

generated by LDA are re-ranked with a combination of the LTSS and TSS scores. In some situations, there are changes which are inserted and deleted multiple times in the document versions for different reasons. However, LTTS can capture those multiple insertions from the set, $D_{word}^{(ins)}$ but is unable to account for the deletions. Unlike LTTS, TSS uses both sets, $D_{word}^{(ins)}$ and $D_{word}^{(del)}$, and therefore those multiple insertions and deletions influence TSS. For that reason, using TSS, the highest ranked sentences selected by LTSS will receive lower scores if they are deleted frequently in the revisions. The motivation is that the changes which are frequently deleted in document versions are implied as non significant/general changes. Therefore, the sentence re-ranking process reconsiders the selected sentences so that the effects of non-significant changes are likely to be reduced in the final summary. The combined score is defined as

$$sentence_i^{(s)} = \lambda \times LTSS(sentence_i) + (1 - \lambda) \times TSS(sentence_i) \quad (65)$$

where the constant term $\lambda \in [0, 1]$ is a regulator parameter which provides flexibility to decide on the proportion of the LTSS and TSS scores to be considered in the final sentence score.

3.3.5 Equality Measurement between Two Sentences

In the sentence ranking process, we need to identify unique sentences by discarding the redundant ones for all proposed approaches. One of the properties expected from a summary of changes is that it should contain no redundant information. In other words, two similar sentences carrying the same information should not be chosen. In practice, it is observed that many sentences have a similar meaning although they contain different terms. These different terms can occur because they are replaced with their synonyms or they appear in various changed forms (e.g., as a result of stemming). However, it can also happen that the words of the two sentences are the same and yet they can carry different meaning. Therefore, it is always difficult to say whether two sentences are equal or not without analyzing

their semantics. There have been works on natural language and semantics-based metrics for the *Semantic Textual Similarity* (STS) task [ACDGA12] in the literature. Here, we will use a simple similarity measurement to address the non-redundancy requirement. The similarity measurement between two sentences, $sentence_i$ and $sentence_j$ is defined as:

$$\rho = \frac{2.0 \times \text{Count}(term : term \in sentence_i \wedge term \in sentence_j)}{\text{Count}(term : term \in sentence_i) + \text{Count}(term : term \in sentence_j)} \tag{66}$$

where $\text{Count}(.)$ returns the number of terms. In Equation 66, the numerator gives the number of common terms between two sentences and the denominator gives the sum of the number of terms in each sentence. Two sentences $sentence_i$ and $sentence_j$ are said to be equal if $\rho \geqslant \xi$, where $\xi \in [0, 1]$ is used as a threshold.

3.4 EXPERIMENTAL SETUP

The experiments to validate the proposed system were organised as follows. First the dataset was prepared with 54 distinct Wikipedia articles. Next, in the data pre-processing stage, an algorithm is proposed for filtering the articles versions with positive contributions. Then, a framework to automatically evaluate the system was used. In the sequel we describe these steps in detail and present a case study to show how the system is built using different approaches.

3.4.1 Dataset Preparation

Wikipedia, the collaboratively edited encyclopedia available on the web, is a major example of a dynamic text collection. Wikipedia is constantly updated by the supporting community to maintain the article's quality. Currently, Wikipedia has more than 30 million articles written in 287 languages and the English Wikipedia alone has more than 4 million articles [Met]. Wikipedia is a pertinent resource in the context of the summarization of changes task for two main reasons; first, the entire revision history of every web page is kept and these revisions can be accessed publicly through an API; second, because this is a publicly available resource, other people can easily reproduce someone's findings. To the best of our knowledge, there are three possible ways of accessing Wikipedia's revision history.

- Programmatically parsing the XML revision dumps [Wik] published by the Wikimedia Foundation on a regular basis. The English Wikipedia is dumped monthly and smaller projects are often dumped twice a month. Nevertheless, the huge size of the English Wikipedia dumps (terabytes in size) makes it impractical to work with this approach.
- RevisionMachine [Rev], a part of the Wikipedia Revision Toolkit [FZG11] provides an API for retrieving the data from the XML revision dumps and stores them into offline databases (MySQL) in a compressed format. According to the author's description [FZG11], "we achieve to reduce the demand for disk space for a recent English Wikipedia dump containing all article revisions from 5470 GB to only 96 GB, i.e. by

98%". Although the required storage space is much lower than its original size, initially it is still necessary to have a large amount of space. Since, in our context, it is not necessary to store all of the article's full revision dumps, it would be preferable to choose a sample set of articles for testing the proposed algorithms.

- Another possibility is using the MediaWiki API [Med], a web service which directly downloads live data from Wikipedia. This option has another advantage, it is flexible to download an up-to-date article of our own choice and if required, the results can be easily compared to other systems.

In our case, the download process is executed as follows. After selecting the sample of Wikipedia articles, the full revision history of each article is downloaded in XML format using the MediaWiki API. The content for each article version is parsed from the downloaded XML and stored as an individual flat file. A folder is created for each article gathering all the versions with the same article-ID. For ease of use, the filename for a document version follows a specific naming convention. The format is shown below.

sequence-ID_time-stamp_annonymous-flag_minor-flag.dmp

The first field is sequence-ID, which assigns 1 to the most recent revision of an article and is sequentially incremented as older revision files are added to the article-ID's directory. Therefore, the sequence-ID of the last revision file of a particular article-ID represents the oldest revision of that article and at the same time it identifies the total number of revisions made to that article. The second field is the time-stamp of the revision. The third field is the anonymous flag which indicates whether the revision was created by an anonymous user (an unregistered user) or not. The anonymous flag is set to true if the revision is made by an anonymous user, otherwise it is set to false. Similarly, the last field denotes whether it is a minor change or not.
There is plenty of meta information available for any revision of an article. Three meta fields (times-stamp, anonymous flag and minor flag) were used here while creating the name for a revision file. It is found that the other "metadata", such as user (who made the revision), user-id (id of the revision creator), size (the size of the revision texts in bytes) and comment (why the revision was made) might be useful for further processing. It is worth noting that for an anonymous user the meta field 'user' holds the IP address and the meta field 'user-id' holds zero. On the other hand, with a registered user the meta field 'user' holds the user name, and the meta field 'user-id' holds an id. All the four metadata items are saved in a different file with the same naming convention but in .inf format. The template is shown below.

sequence-ID_time-stamp_annonymous-flag_minor-flag.inf

The next step is converting each revision file from wiki markup to plain text format. Again each file is saved with the same naming convention but in .txt format. The template is shown below.

sequence-ID_time-stamp_annonymous-flag_minor-flag.txt

These plain text files are used directly for pre-processing.

3.4.2 Filtering Inserted Vandalism & Reverted Revisions

There is a lot of revisions where the changes are reverted back mainly due to vandalism issues for the article. Therefore, it is necessary to filter out the article's revisions during the pre-processing stage. The authors in *Wikipedia Event Reporter* [GKKNS13] simply discarded the updates made by anony-

mous users to avoid most suspicious edits. However, this assumption does not seem reasonable since an anonymous user can also make positive contributions to an article.

In a preliminary step during pre-processing, the *reverted* or *undid* revisions which are simply identified from the corresponding metadata are simply discarded. However, later it is found that discarding only the *reverted* or *undid* revisions is not enough because, at the same time it is necessary to discard the revisions where the vandalism texts actually have been inserted. For this purpose, an algorithm (see Algorithm 1) is proposed to filter out the bad revisions before the proposed methodology starts working. This is one of the possible ways of handling vandalized revisions, so that it is possible to focus on the valuable changes that were made to the revisions.

Table 3 presents the current overall statistics for the complete revision history for 49 selected Wikipedia articles, using each articles lifespan up to January 2015. The columns in Table 3 show the total number of versions made, the number of versions where the minor changes were made, the number of versions where the changes were made by the unregistered (anonymous) users and the number of identified bad revisions using the following proposed algorithm respectively.

Input: A set of revisions $R = \{R_1, R_2, \dots R_T\}$
A set of corresponding meta files: $M = \{M_1, M_2, \dots M_T\}$
Output: A set of filtered revisions $R' = \{R_1, R_2, \dots R_{T'}\}$
Generate users list: $\{U\}_{i=1}^{T} \leftarrow$ *generateUserList*(M)
for $i \leftarrow 1$ ***to*** T **do**
  if $M_i \in$ {"revert"or"undid"} **then**
    Mark R_i as vandalized revision
    VandalismUserName $\leftarrow$ *findVandalismUser*(M_i)
    RestoreUserName $\leftarrow$ *findRestoreUser*(M_i)
    if RestoreUserName $\neq$ NULL **then**
      k = findUser($\{U\}_{k=i+1}^{T}$, RestoreUserName)
      if valid(k) **then**
        Mark $\{R_{i+1}, R_{i+2}, \dots R_{k-1}\}$ as vandalized revisions
      end
    end
    else
      k = *findUser*($\{U\}_{k=i+1}^{T}$, VandalismUserName)
      if valid(k) **then**
        Mark R_k as vandalized revision
      end
    end
  end
end
$R' = \{R_1, R_2, \dots R_{T'}\} \leftarrow$ *filterVandalism*$(R = \{R_1, R_2, \dots R_T\})$
return $R' = \{R_1, R_2, \dots R_{T'}\}$

Algorithm 1: Filtering Inserted Vandalism and Reverted Revisions

3.4.3 Automatic Evaluation

The experiments are performed on 54 case studies for 49 distinct Wikipedia articles within different given time periods. These 54 case studies are selected based on two criteria: i) there can be exactly one significant change made to an article within the chosen time period; ii) the change should be known *a priori*. The second criterion makes it possible to build a framework for evaluating the proposed approaches. If the significant change to any arti-

Table 3: The current (up to January 2015) complete revision history statistics for the selected 49 distinct Wikipedia articles

#	Article ID	Article Name	Total Revisions	# Minor Revisions	# Anonymous Revisions	# Bad Revisions
1	227696	Luciano Pavarotti	2,276	572	859	515
2	19596391	Aníbal Cavaco Silva	744	188	297	110
3	444222	Narendra Modi	6,507	1,288	1,508	2,328
4	623737	Cristiano Ronaldo	10,806	2,638	1,869	2,779
5	186642	Steve Fossett	2,229	637	705	452
6	57570	Sachin Tendulkar	5,914	1,218	1,431	914
7	281337	Viswanathan Anand	2,046	494	720	430
8	329833	David Moyes	2,430	522	1,143	835
9	141833	Pete Seeger	2,263	633	754	256
10	62682	A. P. J. Abdul Kalam	3,785	772	1,626	1,220
11	7412236	Steve Jobs	9,275	2,564	2,630	2,718
12	278119	Charlie Sheen	4,210	1,029	1,396	1,465
13	22468	Osama bin Laden	13,333	3,996	3,389	4,361
14	2944	Ariel Sharon	5,690	1,593	1,866	1,546
15	5792809	Angelina Jolie	6,246	1,792	1,488	1,738
16	68335	Anna Nicole Smith	4,654	1,185	1,255	1,057
17	1687680	Pope Francis	4,645	1,032	49	285
18	1942372	Rituporno Ghosh	843	141	261	110
19	2847	Aung San Suu Kyi	3,829	1,116	1,357	1,196
20	53242	Robin Williams	6,187	1,546	1,857	1,510
21	39626432	Edward Snowden	6,941	1,738	509	851
22	313701	Paul Krugman	4,255	834	911	1,080
23	290474	Alice Munro	933	234	243	169
24	12047	Gaza Strip	4,525	1,033	1,718	1,093
25	419342	David Cameron	7,632	1,909	2,167	2,496
26	6437759	Sebastian Vettel	3,893	842	1,436	1,018
27	656933	2014 FIFA World Cup	7,001	1,098	1,822	1,620
28	2900585	2011 Cricket World Cup	3,418	648	965	829
29	534366	Barack Obama	24,000	6,310	1,669	8,086
30	20396	Michael Schumacher	8,937	1,961	3,244	2,173
31	13076	Gordon Brown	6,094	1,596	1,168	1,508
32	154099	Kim Jong-il	5,950	1,737	1,549	1,692
33	17391	Kosovo	11,230	2,638	1,795	2,256
34	19535	Mikhail Kalashnikov	1,055	305	456	252
35	19831	Margaret Thatcher	9,625	2,345	1,983	1,940
36	21492751	Nelson Mandela	9,002	2,492	2,791	2,643
37	2251390	Charlie Hebdo	994	243	204	137
38	26909	Silvio Berlusconi	7,175	1,512	3,184	1,347
39	27630477	Chelsea Manning	4,585	1,035	612	1,033
40	29490	Saddam Hussein	12,164	3,673	3,272	3,514
41	33983258	Malala Yousafzai	2,813	1,059	489	668
42	36627950	Mars Orbiter Mission	1,822	417	434	404
43	38481813	Hassan Rouhani	1,525	277	359	253
44	40817590	Ebola virus disease	8,288	2,295	2,894	2,887
45	42142305	Malaysia Airlines Flight 370	10,361	2,036	2,037	2,705
46	43326718	Malaysia Airlines Flight 17	5,220	1,107	68	542
47	43529715	Shooting of Michael Brown	6,084	1,610	253	738
48	53029	Muammar Gaddafi	8,733	2,042	2,012	2,031
49	72201	Prince William, Duke of Cambridge	6,453	1,653	2,315	1,926

cle within a time period is known beforehand, it is possible to make a corresponding reference summary. To construct a reference summary for a given time range, a set of sentences are previously selected and extracted to describe the exact significant change. As a result, a set of reference summaries were prepared for all the articles selected corresponding to their given time periods. It is worth noting that the first criterion helps to prepare the reference summaries without any ambiguity. In the summarization of changes system, when a user selects an article of interest for a given time period, the total number of revisions within that time period are counted in order to detect the changes. For a very long time period, in general, the article may have a large number of revisions, which can either be significant or general changes. Now the question is how to prepare a reference summary for that time period. If a very long time period is chosen for an article, the reference summaries of the article may vary from person to person because they can give different priorities to different significant changes. Suppose that there are four significant changes made in a very long time frame. While building a reference summary, one person can choose the first and second as being significant, whereas other may consider the third and fourth as the most significant. Therefore, it would be difficult to prepare a proper reference summary without any ambiguity. In order to avoid these complications initially, the focus is on those time intervals, which have one strong significant change besides the general changes and that change should be known *a priori*. This way, a non-ambiguous human-created summary can be attributed for every input period. The significant change is so prominent, there does not arise any doubt of having multiple reference summaries within the given time period. Moreover, for every reference summary, the sentences are extracted from the latest version of the Wikipedia article in the given time period instead of writing a reference summary manually. This also implies that multiple reference summaries are not required for an article within a time period. In the evaluation framework proposed here, these reference summaries are provided in comparison with the corresponding system-generated summaries.

In the evaluation, the results obtained by using different approaches (system-generated summaries) are compared with summaries created by humans (reference summaries) using ROUGE metrics [LH03] as they are widely used by the DUC and TAC for update summarization task [DA12; WL10; WFQY08; LDS13; ZDXC09; SK08; SKLC08]. These metrics automatically measure the quality of a summary by counting the number of overlapping words between the system-generated summary and a reference summary. When constructing a reference summary for a given time range, the best sentences are selected and extracted from the latest version of that specified Wikipedia article in the given time period. These manually created reference summaries are provided to compare against the system-generated summaries. Therefore, ROUGE scores can express whether the best sentences are picked or not by the proposed approaches. Intuitively, a higher ROUGE score means the system-generated summary using one of the proposed approaches and the human-created summary are more similar. Moreover, according to the authors of the ROUGE toolkit [LH03], ROUGE-1 and ROUGE- 2 have high correlation with the human judgments.

There are different ROUGE measures: ROUGE-N, ROUGE-L, ROUGE-W, ROUGE-S and ROUGE-SU. ROUGE-N is an n-gram recall between a candi-

date summary and a set of reference summaries. ROUGE-N is computed as follows:

$$\text{ROUGE-N} = \frac{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)}, \quad (67)$$

where n is the length of the n-gram, $gram_n$, $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in a system-generated summary and a set of reference summaries, and $Count(gram_n)$ is the number of n-grams in the reference summaries. ROUGE-1 and ROUGE-2 metrics of ROUGE-N are used here with the length of the n-gram as $n = 1$ and $n = 2$, respectively. The other ROUGE metric used is ROUGE-L, which measures the LCS between a system-generated summary and a reference summary. ROUGE-W is similar to ROUGE-L except it is based on weighted LCS where the weighting function is $f(L) = L^{weight}$, L indicates the length of LCS. Here, the input of the *weight* is given as $weight = 1.2$ i.e., the metric ROUGE-W-1.2 is calculated. ROUGE-S measures the overlapping of skip-bigrams where the maximum gap length between two words is given as 4 i.e., ROUGE-S4 is calculated. ROUGE-SU4 is calculated here to perform an evaluation similar to ROUGE-S, where the maximum gap length between two words is given as 4 with the addition of unigram as a counting unit.
Although each of these ROUGE metrics has three scores (recall, precision and F-measure), there is similar conclusion in terms of any of them. For simplicity, in this paper, the average F-measure (the harmonic mean of precision and recall) scores are reported as generated by ROUGE-1, ROUGE-2, ROUGE-L, ROUGE-W-1.2, ROUGE-S4 and ROUGE-SU4 to compare the proposed approaches.

3.4.4 A Detailed Case Study

The experiments are performed on 54 different case studies for a set of Wikipedia articles. However, in order to easily understand the overall framework, a case study is conducted on a Wikipedia article where the flow is described from beginning to end and the intermediate results obtained with the different proposed approaches in the summarization of changes system are demonstrated.

For this case study, the Wikipedia article on *Narendra Modi* with article id 444222 is chosen. The number of revisions made to this article over time is plotted in Figure 5 using the WikiChanges system [NRD08]. The WikiChanges system is a web-based application designed to plot the distribution of the revisions made to an article on a monthly/daily basis. Figure 5 shows that there are 515 revisions made in May, 2014 alone. In these 515 revisions, the significant change is "*Narendra Modi* was elected as Prime Minister during the month of May, 2014". However, most of these edits were not related to this main reason, but instead were general edits. Basically, the number of revisions in an article grows significantly while some important events are taking place, because of the upcoming new information and also due to the update of general information with the growing popularity of the article. The challenge in our system proposed here is picking solely the change that reflects the main reason for edits.

Because the entire month of May, 2014 is selected for the article on *Narendra Modi*, 515 revisions are considered initially. In the pre-processing step,

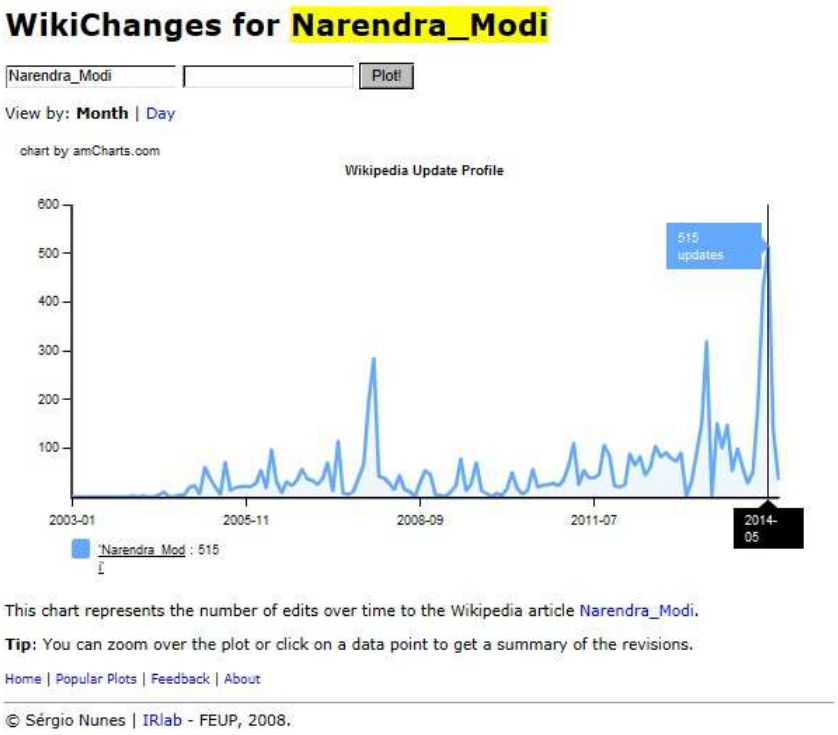

Figure 5: WikiChanges system [NRD08] showing the number of revisions edited in monthly basis for the Wikipedia article on *Narendra Modi*

the vandalized revisions are filtered out from the 515 revisions using Algorithm 1 discussed in Section 3.4.2. After filtering, the number of revisions is reduced to 441. These revisions are supposed to contain the changes with positive contributions made to the article. These 441 revisions are defined as $A = \{\mathrm{rev}_1, \mathrm{rev}_2, \ldots, \mathrm{rev}_{441}\}$ (discussed in Section 3.2). It should also be noted here that rev_1 is the latest revision and rev_{441} is the oldest revision in that given time period.

In the following step, the system extracts the two sets of changes by comparing consecutive revisions. In practice, the changes in revisions can be made in three ways: insertion, modification and deletion. The changes which are caused by insertions or modifications in revisions form the first set, $D^{(ins)}$ whereas the second set, $D^{(del)}$ consists of the changes caused by deletions. Therefore, the two sets of changes can be written together as $D = D^{(ins)} \cup D^{(del)}$. Both sets of changes $D^{(ins)}$ and $D^{(del)}$ are processed further in two modes: word mode and block mode. In the word mode, the changed words are taken from the consecutive revisions by comparing on a word basis while in block mode, the changed information are excerpted in paragraph basis. After this process, the four sets of changes $D^{(ins)}_{word}$, $D^{(del)}_{word}$, $D^{(ins)}_{block}$ and $D^{(del)}_{block}$ are generated. The maximum cardinality for each of these four sets is 440.

In the following step for sentence ranking, different term scoring measurements are used in different approaches. In the first approach, baseline temporal term score (BTTS) for each term is generated using Equation 48 with the sets A, $D^{(ins)}_{word}$ and $D^{(del)}_{word}$. In Equation 48, the range of parameter α varies between 0 and 1. According to the author's [JBI04] explanation, increasing the value of parameter α allows higher relative scores to be given

to the rare or specific terms rather than terms with a general meaning. Because in this step the goal is to preferably find the significant terms, the value of parameter α is set to 1.

In the second approach, the temporal term score (TTS) is calculated using Equation 52 with the two sets of changes, $D^{(ins)}_{word}$ and $D^{(del)}_{word}$. In the third approach, a feature file is created using the set $D^{(ins)}_{word}$. The feature file consists of 440 feature vectors. This feature file is given as an input file in the LDA model and the output from the LDA model using Equation 62 represents a set of words/terms in which each word is associated with a score for each of the K latent topics. To facilitate the system's evaluation, the time period with one significant change is chosen. For this reason, the number of latent topics is always given as $K = 2$, where one topic is supposed to correspond to the significant change and the other one reflects general changes. Section 3.3.3 mentions that Gibbs sampling [Gri02] and Markov Chain Monte Carlo technique are used to infer the LDA model parameters. Here, Gibbs sampling is run for 1000 iterations and for all runs, a symmetric Dirichlet prior on θ with $\alpha = 0.5$ and a symmetric Dirichlet prior on ϕ with $\beta = 0.1$ are used. The terms with corresponding scores are generated using the LDA model for $K = 2$ latent topics.

A first exploratory examination of the proposed approaches is performed and the approaches are compared by looking at the illustrative case study presented in Table 4. This table lists the 30 best scoring terms obtained with the three main approaches, baseline temporal term score (BTTS), temporal term score (TTS) and latent topic term score (LTTS) for the Wikipedia article on *Narendra Modi* for the month of May, 2014. LTTS is shown separately in the last two columns for topic 1 (LTTS: Topic 1) and for topic 2 (LTTS: Topic 2). It is possible to confirm that there are clear differences between each pair of columns, even when the top 30 terms are considered. The terms which are directly related to the significant change are marked in bold in each column of Table 4. In the first column, it is obvious that the top 30 terms selected by BTTS do not clearly indicate the significant change. Among the 30 terms, only the terms *elections2014* and *26th* are relevant when using BTTS. TTS selects a higher number of relevant terms than BTTS. Basically, TTS assigns higher scores to the terms which have been inserted more times and deleted less times in the revisions for a given time period. If, within a period, there haven't been many general changes besides the significant change, then TTS can pick the relevant terms to describe the significant change. However, if there are general changes within that period, individually TTS is not sufficient to pick and assign the relevant terms with higher scores due to conflict with the terms in the general changes. This problem is handled using LTTS. The third column (LTTS: Topic 1) provides the relevant terms, which are more specifically related to the significant change. Relevant terms such as *prime*, *minister*, *2014*, *india*, *bjp*, *election*, *indian*, *born* and *general* are selected in both TTS and LTTS: Topic 1. Besides these relevant terms, terms such as *victory*, *lok*, *sabha*, *chief*, *president* are more specific to describe the significant change are further captured in LTTS: Topic 1. Moreover, there are no terms related to the significant change in LTTS: Topic 2, whereas the terms that describe other common changes are reflected in LTTS: Topic 2.

An observable point to be noted in Table 4 that there are the terms such as *gujarat*, *narendra*, *modi*, which are common in both columns, LTTS: Topic 1 and LTTS: Topic 2. This is the reason why the concept of inverse topic frequency (itf) is used in Equation 62 to calculate the latent topic term score

Table 4: Top 30 scoring terms obtained using different approaches on the Wikipedia article on *Narendra Modi* for the month of May, 2014

BTTS	TTS	LTTS: Topic 1	LTTS: Topic 2
president1	modi	modi	modi
governor2	**minister**	**prime**	gujarat
206	gujarat	**minister**	riots
indepedent	court	**2014**	narendra
interviews	**prime**	**india**	media
trial	**bjp**	**bjp**	term
raised	**india**	14th	sit
1994	sit	**election**	2002
07	riots	**15th**	modi's
kejrival	2002	party	court
incendiary	**election**	narendra	state
www.hindustantimes.com	supreme	**indian**	truth
categories	report	**victory**	government
elections2014	**2014**	leaders	report
says-gujarat-government	media	current	news
eastern	**born**	**elections**	police
www.dnaindia.com	**general**	**president**	supreme
differing	narendra	office	allegations
conclusion	state	**lok**	stated
triumphs	party	**general**	2012
counsel	modi's	**chief**	reporting
indepedence	government	date	statement
support	evidence	**sabha**	allegedly
rising	2012	article	case
25,375,63	rss	times	evidence
positively	case	gujarat	projects
responded	**elections**	hindu	gift
26th	march	vadodara	barkha
substance	**indian**	**born**	political
prosecutable	2010	varanasi	rajdharma

(LTTS). In Equation 62, a weight which is assigned by LDA model to each term in a latent topic is further multiplied by the inverse topic frequency (itf) of that term. Therefore, the common terms in both topics obtain lower scores with the motivation that they may convey less information to the significant change.

The following step in sentence ranking is calculating a score for each sentence so that the sentences with the higher scores are presented as a summary. The sentence score in both the first and second approaches, i.e. baseline temporal sentence score (BTSS) using Equation 49 and temporal sentence score (TSS) using Equation 53 are simply calculated based on the sum of the scores of all terms divided by the total number of terms (excluding stop words) the sentence contains. In the third approach, latent topic sentence score (LTSS) using Equation 64 is calculated in a similar way, specified in the first and second approaches. However, the only difference in LTSS is that each sentence's score needs to be calculated for each latent topic. That score is finally decided when a topic gives the maximum score to that sentence. LTSS generates a set of sentences of the form $\{(sentence_i, sentence_i^{(s)}, z), sentence_i \in S\}$, where $sentence_i^{(s)}$ is the corresponding score of $sentence_i$ and z is the label to identify which topic the sentence belongs to. The last three approaches are a combination of LTSS and TSS approaches. In these approaches, basically the top ranked sentences of the form $\{(sentence_i, sentence_i^{(s)}, z), sentence_i \in S\}$ generated by LTSS are re-ranked with a combined score of LTSS and TSS. The regulator parameter $\lambda \in [0, 1]$ with values $\lambda = 0.75$, $\lambda = 0.50$ and $\lambda = 0.25$ are given in Equation 65. The top 5 scoring sentences with different approaches, regarding the Wikipedia article on *Narendra Modi* for the month of May, 2014 are

shown in Table 5. The table shows that no redundant sentence is chosen in 6 different summaries generated by different approaches. Non-redundancy is one of the objectives of the summarization of changes task. The threshold, $\xi = 0.70$ is used here to measure the similarity between two sentences using Equation 66.

The final step is automatically evaluating the summaries generated by the system using different approaches. Different ROUGE metrics are used to measure the quality of a summary comparing a system-generated summary using one of the proposed approaches, and a summary created by a human. The higher the ROUGE scores, the more similar are the summary created with the proposed approach and the summary created by a human. Table 6 represents an example of a human-generated summary used as reference summary for the Wikipedia article on *Narendra Modi*. Table 7 shows the performances of the different approaches after evaluating Wikipedia article on *Narendra Modi* for the month of May, 2014. Table 7 shows that the approach using the combination of LTSS and TSS ($\lambda = 0.75$) provides the best summary for this case study.

3.5 RESULTS AND DISCUSSION

The different approaches proposed are compared in 54 case studies on 49 distinct Wikipedia articles. Table 8 demonstrates each article's given time period, the number of revisions made during that period and the actual number of revisions considered within that period after filtering with Algorithm 1. Some articles are chosen more than once to make different case studies by selecting several time periods. For example, the fifth, sixth and seventh rows in Table 8 are selected for the same article, *Steve Fossett* with article ID 186642, but with different time periods. The overall performances of all the approaches based on 6 ROUGE metrics are shown in Table 9. From the comparison results, the following observations are noticed:

- BTSS chooses mainly the sentences which have uncommon/typical words on a specific domain. This may be advantageous in order to detect the significant changes that reflect new information. However, in practice the changes are not always made using typical words. That is why the overall ROUGE scores for BTSS are the lowest.
- TSS chooses the sentences-whose terms are inserted more frequently and simultaneously deleted less times in the revisions. A term which has been revised frequently in a set of document revisions should be more important because it could reflect a significant change. Therefore, TSS performs better than the implemented baseline approach, BTSS. However, again it is not always possible to capture the main change because the terms are revised both by adding new information and by updating the general information of the article.
- LTSS selects the sentences from different topics where each topic is a cluster of terms which reflects the same kinds of changes. This way, all the words related to any change are likely to be grouped. The sentences which contain more words related to a particular topic have higher weights. Moreover, it is important to notice that if a topic brings a significant change, the terms associated to this topic have higher weights than the terms associated to another topic. This is why LTSS outperforms BTSS and TSS.

Table 5: Sentences selected by the summarization of changes system using different approaches on the Wikipedia article on *Narendra Modi*

BTSS			
#	$sentence_i^{(s)}$	z	$sentence_i$
1	0.067046	-	Modi raised terrorism issues with Pakistan PM including 26/11 trial.
2	0.04097	-	Modi reacted to this in words "Truth alone triumphs!".
3	0.021695	-	He will be the first holder of this office born after India became independent of the British.
4	0.017615	-	In 1978, Modi graduated with an extramural degree through Distance Education in political science from Delhi University.
5	0.015653	-	The stock market responded positively to the election result with the BSE SENSEX rising more than 6 per cent to a record high of 25,375,63.

TSS			
#	$sentence_i^{(s)}$	z	$sentence_i$
1	0.011325	-	Instead he acknowledged that Rajdharma was followed by Modi and his administration.
2	0.009588	-	This is where the witch hunt of Narendra Modi started.
3	0.009295	-	Critics of Modi have used this statement to argue that Modi wanted to curb free speech at that time.
4	0.008989	-	He is now in the course of being the 14th Prime Minister of India.
5	0.008707	-	Narendra Modi was sworn in as Prime Minister on 26 May 2014 at the Rastrapati Bhavan.

LTSS			
#	$sentence_i^{(s)}$	z	$sentence_i$
1	0.00528	1	Modi is India's first prime minister born after the country's independence.
2	0.003946	1	Narendra Modi was sworn in as Prime Minister on 26 May 2014 at the Rastrapati Bhavan.
3	0.003909	1	He led the BJP in the AprilMay 2014 general election, which resulted in a majority for the BJP in the Lok Sabha, first time any party has done so since 1984.
4	0.002974	1	Fourth term (20122014) After being elected as Prime Minister, Modi resigned from the post of chief minister on 21 May 2014, and his MLA seat from the Maninagar constituency, after delivering a leaving speech described as emotional.
5	0.002733	1	In the oath ceremony for prime minister post Modi invited leaders of SAARC countries to strengthen relationship and increase business.

LTSS + TSS ($\lambda = 0.75$)			
#	$sentence_i^{(s)}$	z	$sentence_i$
1	0.006734	1	Modi is India's first prime minister born after the country's independence.
2	0.005114	1	Narendra Modi was sworn in as Prime Minister on 26 May 2014 at the Rastrapati Bhavan.
3	0.003961	1	He led the BJP in the AprilMay 2014 general election, which resulted in a majority for the BJP in the Lok Sabha, first time any party has done so since 1984.
4	0.003361	1	Fourth term (20122014) After being elected as Prime Minister, Modi resigned from the post of chief minister on 21 May 2014, and his MLA seat from the Maninagar constituency, after delivering a leaving speech described as emotional.
5	0.002834	1	On 9 June 2013, Modi was appointed Chairman of the BJP's Central Election Campaign Committee for the 2014 general election, at the national level executive meeting of BJP.

LTSS + TSS ($\lambda = 0.50$)			
#	$sentence_i^{(s)}$	z	$sentence_i$
1	0.008187	1	Modi is India's first prime minister born after the country's independence.
2	0.006282	1	Narendra Modi was sworn in as Prime Minister on 26 May 2014 at the Rastrapati Bhavan.
3	0.004013	1	He led the BJP in the April-May 2014 general election , which resulted in a majority for the BJP in the Lok Sabha , first time any party has done so since 1984.
4	0.003996	2	The SIT questioned Modi in March 2010, and in May 2010 presented its report before the Court, stating that it found no evidence to substantiate the allegations.
5	0.003748	1	Fourth term (2012-2014) After being elected as Prime Minister, Modi resigned from the post of chief minister on 21 May 2014, and his MLA seat from the Maninagar constituency, after delivering a leaving speech described as emotional.

LTSS + TSS ($\lambda = 0.25$)			
#	$sentence_i^{(s)}$	z	$sentence_i$
1	0.009641	1	Modi is India's first prime minister born after the country's independence.
2	0.007449	1	Narendra Modi was sworn in as Prime Minister on 26 May 2014 at the Rastrapati Bhavan.
3	0.005742	1	Narendra Modi face first FIR against him in entire life.
4	0.005337	2	The SIT questioned Modi in March 2010, and in May 2010 presented its report before the Court, stating that it found no evidence to substantiate the allegations.
5	0.004383	2	One of such websites, Gujarat Riots , attempts to ""bring out the TRUTH, THE WHOLE TRUTH, AND NOTHING BUT THE TRUTH"" of the Gujarat riots of 2002; including the truth of "myths" like "the Gujarat police turned a blind eye to the rioting", "the Gujarat government was involved in the riots", that Narendra Modi said:"Every action has equal and opposite reaction", "Narendra Modi gave free hand to rioters for 3 days", "no one was brought to justice for the riots" etc.

Table 6: An example of a human-generated summary as reference summary on the Wikipedia article on *Narendra Modi*

#	Sentences are selected for a reference summary
1	Narendra Modi was sworn in as prime minister on 26 May 2014 at the Rastrapati Bhavan.
2	He is India's first prime minister born after the country's independence.
3	In September 2013, BJP announced Modi as their prime ministerial candidate for the 2014 Lok Sabha election.
4	He led the BJP in the 2014 general election, which resulted in an outright majority for the BJP in the Lok Sabha (the lower house of the Indian parliament) the last time that any party had secured an outright majority in the Lok Sabha was in 1984.
5	After being elected as Prime Minister, Modi resigned from the post of chief minister on 21 May 2014, and his MLA seat from the Maninagar constituency, after delivering a leaving speech described as emotional.

Table 7: ROUGE scores using all the proposed approaches on the Wikipedia article on *Narendra Modi* for the month of May, 2014

Proposed approaches	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-W-1.2	ROUGE-S4	ROUGE-SU4
BTSS	0.22115	0.01942	0.21154	0.08044	0.02970	0.06250
TSS	0.36538	0.16505	0.34615	0.14900	0.14455	0.18092
LTSS	0.73438	0.60630	0.71094	0.35325	0.52960	0.56383
LTSS + TSS ($\lambda = 0.75$)	**0.73962**	**0.60836**	**0.72453**	**0.35951**	**0.53127**	**0.56611**
LTSS + TSS ($\lambda = 0.50$)	0.72243	0.58237	0.69962	0.34459	0.51829	0.55240
LTSS + TSS ($\lambda = 0.25$)	0.39298	0.19081	0.36491	0.17699	0.17204	0.20977

- In the last approach, the sentences produced by LTSS are re-ranked by combining LTSS and TSS. The goal is to incorporate the deletion effects into LTSS. However, the overall performance of the last approach is not improved comparatively to LTSS, which is shown in Table 9. This happens because, as TSS itself does not outperform LTSS, the re-ranking approach using a linear combination of TSS does not perform better or the selected articles do not have the effects of deletions in LTSS.

The statistical distributions of ROUGE scores for all approaches are shown using boxplots in Figure 8. It is observed that the performances of different approaches are arranged from lower to higher order as BTSS, TSS, the proportions of LTSS and individual LTSS. The statistical tests are further performed to see if the differences in ROUGE scores for different proposed approaches are significant or not. A Friedman test [Fri40] is used because the samples (ROUGE scores for all case studies) are not normally distributed. This test reveals for all ROUGE metrics, p-value is lower than 0.001. This indicates that for each ROUGE metric, there is at least one statistically significant difference between two of the approaches. To identify these cases, a post-hoc analysis using Nemenyi tests [Nem63] is conducted. Tables 10 and 11 show that there are significant differences between BTSS and TSS ($p < 0.005$), between BTSS and LTSS ($p < 0.001$), between BTSS and LTSS + TSS ($\lambda = 0.75$) ($p < 0.001$), between BTSS and LTSS + TSS ($\lambda = 0.50$) ($p < 0.001$), and between BTSS and LTSS + TSS ($\lambda = 0.25$) ($p < 0.001$) for ROUGE-1 and ROUGE-L scores. Similar results are found for other ROUGE metrics as well.

Table 8: Number of edits made to the different Wikipedia articles for 54 selected time periods

Article #	Article ID	Time Period (YYYY-MM)	# Edits	# Edits after Filtering
1	227696	2007-09	626	535
2	19596391	2006-01	89	73
3	444222	2014-05	515	441
4	623737	2014-01	169	154
5	186642	2007-09	462	409
6	186642	2008-02	87	69
7	186642	2008-10	389	320
8	57570	2012-03	86	78
9	281337	2010-05	117	96
10	329833	2013-05	271	177
11	329833	2014-04	64	62
12	141833	2014-01	190	178
13	62682	2012-06	132	84
14	7412236	2011-10	1,431	1,252
15	278119	2011-03	335	293
16	22468	2011-05	1,542	1342
17	2944	2006-01	1,048	793
18	5792809	2006-01	195	138
19	5792809	2006-06	280	186
20	68335	2006-09	299	276
21	68335	2007-02	1,656	1,357
22	1687680	2013-03	2,853	2,707
23	1942372	2013-05	296	267
24	2847	2010-11	268	214
25	53242	2014-08	1,603	1,467
26	39626432	2013-06	2,086	1,850
27	313701	2009-08	541	487
28	290474	2013-10	276	241
29	12047	2014-07	344	275
30	419342	2010-05	556	478
31	6437759	2010-11	278	200
32	656933	2014-07	554	519
33	2900585	2011-04	209	163
34	534366	2008-11	1,434	1,033
35	20396	2006-10	891	818
36	13076	2007-06	748	598
37	154099	2011-12	427	387
38	17391	2008-02	1,575	1,071
39	19535	2013-12	101	90
40	19831	2013-04	768	663
41	21492751	2013-12	689	600
42	2251390	2015-01	518	424
43	26909	2008-05	114	78
44	27630477	2013-08	558	443
45	29490	2006-12	1,278	1,192
46	33983258	2012-10	1,253	1,104
47	36627950	2014-09	433	311
48	38481813	2013-06	683	599
49	40817590	2014-10	1,072	1,019
50	42142305	2014-03	7,667	5,677
51	43326718	2014-07	4,001	3,742
52	43529715	2014-09	712	564
53	53029	2011-06	906	732
54	72201	2011-04	356	277

Table 9: Overall ROUGE scores using all proposed approaches for 54 different case studies on 49 distinct Wikipedia articles

Proposed approaches	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-W-1.2	ROUGE-S4	ROUGE-SU4
BTSS	0.28673	0.09421	0.25832	0.13063	0.08668	0.12073
TSS	0.38563	0.19322	0.35645	0.17489	0.17123	0.20748
LTSS	**0.45978**	**0.27684**	**0.43172**	**0.22362**	**0.24573**	**0.28169**
LTSS + TSS ($\lambda = 0.75$)	0.44490	0.26174	0.41533	0.21115	0.22920	0.26561
LTSS + TSS ($\lambda = 0.50$)	0.42922	0.25217	0.40004	0.20326	0.21979	0.25527
LTSS + TSS ($\lambda = 0.25$)	0.41502	0.23469	0.38427	0.19563	0.20465	0.24020

Table 10: Pairwise comparisons of **ROUGE-1** scores for all proposed approaches using Nemenyi post-hoc test. There are 54 different case studies on 49 distinct Wikipedia articles.

	BTSS	TSS	LTSS	LTSS + TSS ($\lambda = 0.75$)	LTSS + TSS ($\lambda = 0.50$)
TSS	**4.2e-05**	-	-	-	-
LTSS	**2.4e-12**	0.075	-	-	-
LTSS + TSS ($\lambda = 0.75$)	**1.7e-10**	0.283	0.992	-	-
LTSS + TSS ($\lambda = 0.50$)	**4.4e-09**	0.585	0.889	0.997	-
LTSS + TSS ($\lambda = 0.25$)	**4.1e-07**	0.953	0.449	0.820	0.976

We have also studied the same article in different time periods. As an example, three different time periods are chosen for the Wikipedia article on late USA adventurer *Steve Fossett* with article ID 186642. Table 8 shows the details of these three records. The first time period is the month of September, 2007 where the main changes are related to the fact that Fossett was reported missing; the second is the month of February, 2008, where the main changes are related to the fact that Fossett was declared dead and the third one is the month of October, 2008, where the main changes are related to the identification of Fossett's airplane wreckage and other personal items, which were found near Mammoth Lakes, California. The purpose is to test whether the system is able to capture those significant changes for the same article but for different time periods. Table 12 shows the best summaries given by either LTSS or with the combination approach. Table 12 shows that the three different significant changes do not overlap, and the summarization of changes system is able to detect the changes for the different time periods.

Table 11: Pairwise comparisons of **ROUGE-L** scores for all proposed approaches using Nemenyi post-hoc test. There are 54 different case studies on 49 distinct Wikipedia articles.

	BTSS	TSS	LTSS	LTSS + TSS ($\lambda = 0.75$)	LTSS + TSS ($\lambda = 0.50$)
TSS	**0.00036**	-	-	-	-
LTSS	**2.4e-12**	**0.01945**	-	-	-
LTSS + TSS ($\lambda = 0.75$)	**4.0e-11**	0.06109	0.99890	-	-
LTSS + TSS ($\lambda = 0.50$)	**8.6e-09**	0.35366	0.84518	0.96807	-
LTSS + TSS ($\lambda = 0.25$)	**1.7e-06**	0.88948	0.29607	0.53350	0.94658

Table 12: Sentences selected for different time periods by the summarization of changes system from the Wikipedia article on *Steve Fossett*

2007-09 (YYYY-MM)	2008-02 (YYYY-MM)	2008-10 (YYYY-MM)
Fossett has been reported as missing since 3 September 2007.	A Cook County, Illinois probate judge today declared wealthy Chicago adventurer Steve Fossett legally dead on 15 February 2008, five months after his plane disappeared.	No plane wreckage found.
He was last seen flying a single engine private aircraft, a Citabria , south of Smith Valley, Nevada.	That's where I thrived.	On October 2nd, 2008, human remains were purportedly found near the wreckage site.
About two dozen aircraft were involved in the search.	On November 26 , 2007 , Fossett's wife requested that Fossett be declared legally dead.	He said it was unclear if it was human - and added that he did not know of any confirmed human remains being found
Disappearance expand On September 4th, 2007, the Record-Courier , reported that Steve Fossett had disappeared.	On November 2 , 2007 , Peggy Fossett and Dick Rutan accepted the Spread Wings Award in Steve Fossetts behalf at the 2007 Spreading Wings Gala, Wings Over the Rockies Air and Space Museum , Denver, Colorado.	October 2nd 2008 Authorities have found the plane Steve Fossett was flying when he disappeared last year, but they have not found the millionaire adventurer's body, the Madera County, California, sheriff said Thursday.

This paper also studies the effects on the ROUGE scores of the increasing number of revisions made to the different articles for the given time periods. Generally, when the number of revisions increases, more conflicts are likely to occur while choosing the sentences with the significant change vs other changes. Because a similar conclusion is obtained for different ROUGE scores, ROUGE-1 and ROUGE-2 results are shown in Figures 6 and 7. It is observed from the figures that the system performs well even when the number of revisions is within the higher range of $[1500, 5750]$.

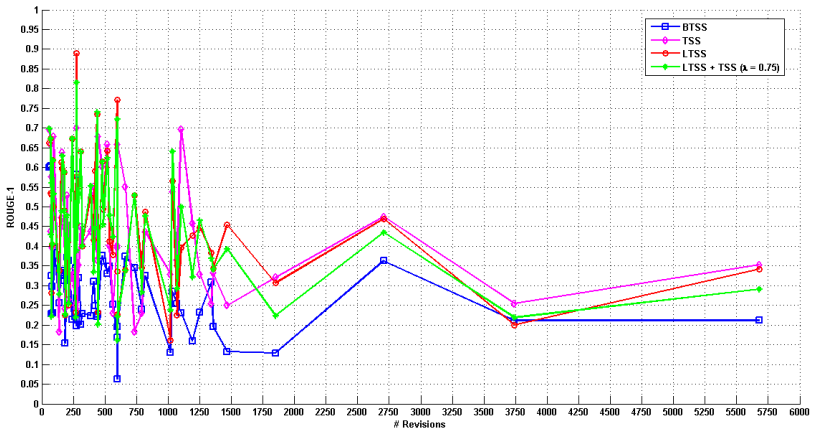

Figure 6: Effects on ROUGE-1 scores of the increasing number of revisions in the selected articles using different approaches. The total of 54 case studies on 49 distinct Wikipedia articles are arranged in ascending order according to the number of revisions within the given time periods.

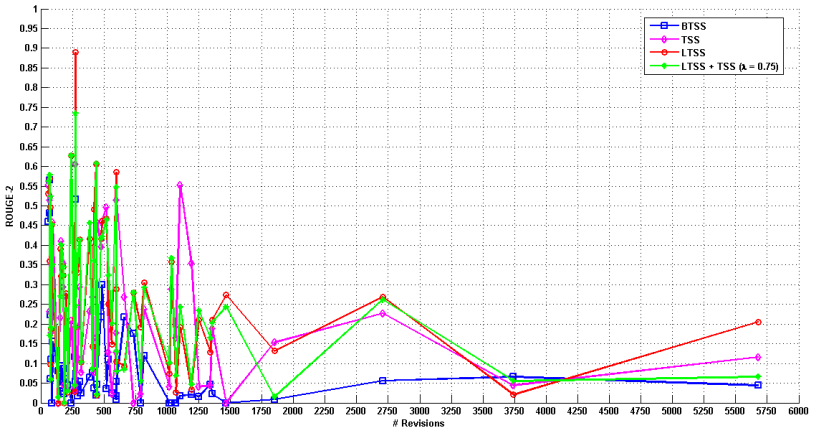

Figure 7: Effects on ROUGE-2 scores of the increasing number of revisions in the selected articles using different approaches. The total of 54 case studies on 49 distinct Wikipedia articles are arranged in ascending order according to the number of revisions within the given time periods.

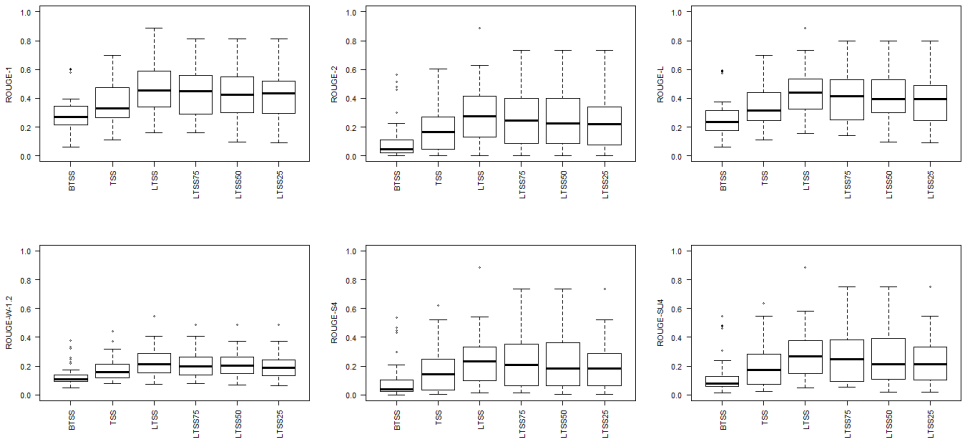

Figure 8: Boxplots on all ROUGE scores for different approaches. There are 54 different case studies on 49 distinct Wikipedia articles. Here, the labels, LTSS75, LTSS50 and LTSS25 refer to LTSS + TSS ($\lambda = 0.75$), LTSS + TSS ($\lambda = 0.50$) and LTSS + TSS ($\lambda = 0.25$), respectively.

3.6 SUMMARY

The summarization of changes focuses on the generation of abridged and non-redundant accounts of document modifications in dynamic text collections. This research introduces a new framework for summarizing changes from a set of revisions made to a Wikipedia article during a given time period. Four different approaches are proposed for the summarization of changes. The first approach provides a baseline that is adapted from an existing related work [JBI04], which periodically monitors a web collection in search for recent changes and generates their summary with respect to a specific topic. The summarization of changes differs from this task, as it addresses the changes and generates their summary in dynamic text collections within any user-defined period. In the second approach, each term's temporal aspect is investigated by considering the joint probabilities of both the insertion and deletion events over a set of document versions within the given period. The third approach is based on the LDA model for finding hidden/latent topic structures of changes. The fourth approach is a combination of the previous two approaches in which the top ranked sentences generated from the third approach are re-ranked with a combined score from the second and third approaches.

The four approaches are used initially to estimate the term scores, and then to rank the sentences based on those scores. Finally, for generating a summary, a few top ranked sentences are chosen independently for each of the approaches. All of them are evaluated using ROUGE metrics by comparing the system-generated summaries and the human-created reference summaries. It is observed that the third approach based on the LDA model outperforms the others.

Although a set of articles from Wikipedia have been used with their full revision histories as a document collection, these approaches can be used in other time-dependent collections. For example, any of them can be used to generate a change summary on the history of a single web page from a web archive.

In this study, a simple metric is used for equality measurement between two sentences. Future work is expected to investigate richer equality measurement metrics, such as the ones used in NLP for the Semantic Textual Similarity task. This will improve the identification of redundant sentences. In this work, a framework was set up and used for automatic evaluation. The summaries produced by each of the approaches are evaluated comparatively to the manual summaries using ROUGE metrics. An extrinsic evaluation, considering human feedback in this task is expected as future work.

To evaluate the proposed system, we took into consideration the time periods where exactly one significant change has occurred in any article. For this reason, the number of latent topics is considered as $K = 2$. However, when there is more than one significant change within a given time range, then it is necessary to increase the value of K. In those cases, finding an optimum K is an interesting research challenge to address in the future.

4 MULTI-LEVEL CHANGES SUMMARIZATION

In this chapter, the exploration of the LDA model proceeds to detect multiple significant changes for a wider temporal interval. It tries to uncover different latent changes in terms of different topics. The number of latent changes for LDA model is estimated using Bayesian model selection. The number of estimated latent changes is thereafter assumed as the number of different categories of candidate changes, which are likely to include both significant and non-significant ones.

For each category of candidate changes, a burst region is identified. A burst is defined as a succession of changes belonging to the same topic that occurred in spatial proximity of each other within a short enough period of time. A burst region is the spatial region where the burst occurred. The importance of each category of candidate changes (i.e. topic) is assessed by analyzing its burst region, as well as the topic ratios, in order to filter out the non-significant ones. A set of sentences is then selected from the burst region to present a meaningful and coherent summary, for each significant topic. These summaries are generated hierarchically: a summary is presented for each significant topic in an intermediate level and, at the top-level, a single summary is generated in order to consolidate the most significant changes. The novelty of this work is that (i) it can produce multiple summaries of changes within a given time range, facilitating content discovery and navigation in different levels depending on a user's interest, (ii) the system can effectively produce multi-level summaries without the constraint of specifying a default model selection criterion *a priori* and (iii) the produced summaries are focused and coherent.

4.1 INTRODUCTION

The volume of online documents has been increasing frequently and significantly over time. A high rate of changes in textual contents is observed in dynamic text collections. While users have access to an increasing amount of evolving information, they may find it hard to determine which were the most important changes made. This has induced a new IR task to provide an automatic summarization of changes in dynamic text collections [JBI04; NRD08; KNR15]. The task consists in obtaining a summary of the revisions made to a document over a specific period of time.

The stated task can be explained more precisely through an example using the Wikipedia article on *Steve Fossett*. Upon studying this article, it is found that most of the edits (additions, modifications or deletions) occurred within the time range from September, 2007 to October, 2008, due to three main facts: (i) Fossett was reported missing on September, 2007; (ii) Fossett was declared dead on February, 2008; and (iii) Fossett's airplane wreckage and other personal items were found on October, 2008. In this perspective, we can say that three types of significant changes occurred. The aim of the task is to generate a summary that reflects how many major changes occurred, what type of changes were made, and when they occurred within

this time period. A system proposed for this task would provide users the following information:

- The number of different changes that were made throughout the revisions of a document is determined for a given time interval. If there are multiple prominent changes, they are likely to be distinguished.
- Summaries of changes are presented at multiple levels, in order to provide information to users depending on their requirements. At an intermediate level, a separate summary for each important change is generated; however, at the top-level, a single summary consolidates the most significant changes. It is worth to mention that each intermediate summary contains more detailed information for a particular category of changes. For example, if users have interest in exploring the information regarding the disappearance of Steve Fossett (and not his death or the discovery of his airplane's wreckage), they can view the corresponding intermediate summary.
- The burst regions are associated with for each significant category of changes. For example, if there were three important categories of changes within a given time period, not only they should be correctly detected as three but also be mapped with their corresponding burst regions. The advantage of finding the burst region for each significant category is that it can improve summary coherence. Instead of including all sentences that contain alterations, only the sentences within the burst region are passed on to the sentence ranking step. The sub-changes related to any significant change tend to co-occur within a temporal proximity. In this work, this intuition is explored to determine whether or not it achieves coherence among the selected sentences for the intermediate summaries.

The core technique of temporal summarization research is the automatic generation of summaries by extracting the key sentences from a set of texts updated over time. One of the drawbacks of such approaches is that they do not guarantee the coherence among the selected sentences in the summaries, with the help of only sentence ranking algorithms. Due to the nature of textual data, it is hard to select the right sentences from a large set and still maintain continuity among them. In fact, this is a challenging problem in the NLP community for the task of automatic text summarization [BL08; BKLB12]. In this context, our basic intuition is that the updates related to the same category of changes generally occur together. Thus, if it is possible to determine the burst region for any particular category of changes, the summary generated from this identified region is likely to be coherent. The novelty of this approach is to propose a way of generating multiple summaries of changes at different levels in the context of temporal summarization. Haghighi et al. proposed HIERSUM [HV09], a hierarchical Bayesian approach to produce multiple 'topical summaries', but for the task of multi-document summarization. However, multi-document summarization techniques do not handle the links between the sequential temporal connections and the updated texts.

Jatowt et al. [JBI04] first introduced the task of changes summarization in web collections within a limited scope, where only the 'recent, important' changes are considered. Later, Nunes et al. [NRD08] addressed this task in a wider temporal aspect, presenting changes summarization for any user-defined time period. Following this problem, we initially developed

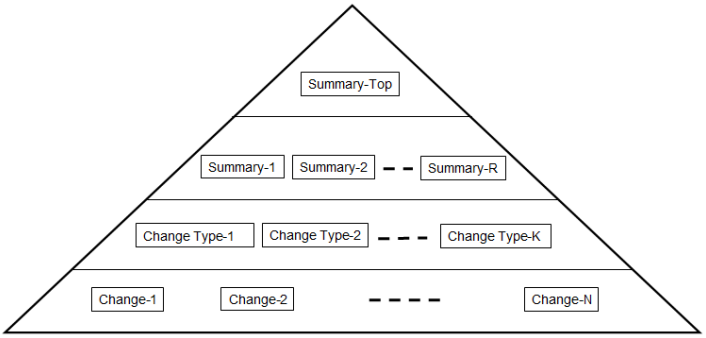

Figure 9: Hierarchical structure for multi-level changes summarization

a system based on Latent Dirichlet Allocation model (LDA) to categorize the changes into significant and non-significant [KNR15]. However, the limitation of this previous system was the strict assumption of the number of latent topics in the LDA as always two—one topic for capturing the significant changes and another for general changes. The shortcomings of this assumption are evident in scenarios where there can be multiple change categories within a given time range. In those cases, all of them are put into a single topic, which is not adequate for capturing each change category explicitly. Hence, it is necessary to detect the number of different changes made throughout the revisions of a document for the given time period *a priori*.

In this study, we have developed a multi-level changes summarization system that can automatically detect the multiple significant change categories present in a collection of revised documents, within a user-defined time period. The overall system is based on a hierarchy of levels as shown in Figure 9. The hierarchical levels from the bottom to the top are: (i) a set of changes (differences) are collected by comparing the consecutive document versions within a given time period, (ii) a set of latent change categories are identified by the LDA model, (iii) a coherent summary is produced for each significant change category, and (iv) a top summary consolidating the most significant changes is presented.

The rest of this chapter is organized as follows: Section 4.2 presents the basic architecture of the proposed system. There are six major stages followed in the proposed system. The algorithms used in each stage, as well as its corresponding results are described in detail. Section 4.3 presents a framework for automatically evaluating multiple summaries obtained from different categories of changes with respect to time. Section 4.4 reports the experimental details and Section 4.5 presents the analysis of experimental results. Finally, we summarize this chapter in Section 4.6.

4.2 THE MULTISUMMAR SYSTEM

This work presents a new multi-level changes summarization system, called *MultiSummar* for the task of summarizing changes, which generates a summary of significant alterations made to a sequential collaborative text collection within a given time period. The overall architecture of the *MultiSummar* system is shown in Figure 10.

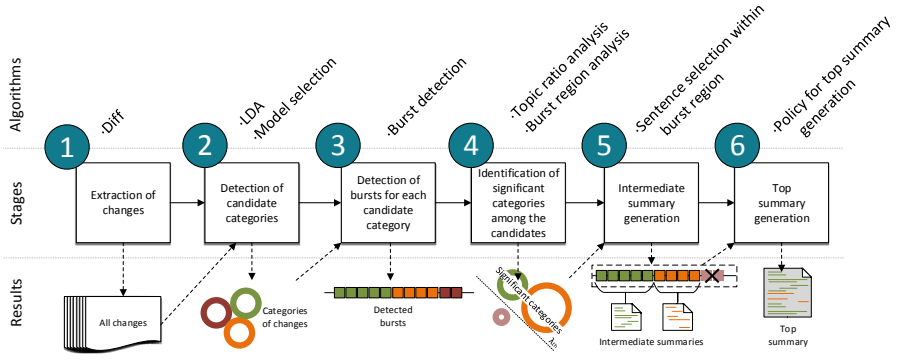

Figure 10: Schematic diagram for multi-level changes summarization system (*MultiSummar*)

This figure depicts the six major stages followed in the proposed system. The algorithms used in each stage, as well as the corresponding results are presented. In stage **1**, the system extracts changes from each pair of consecutive versions. In stage **2**, the system automatically finds hidden/latent topic structures of changes. To accomplish this, the LDA model is used and the number of latent topics, which are the candidate categories of changes, is determined using model selection criteria. In stage **3**, the system locates the burst region for each individual category of changes (topic). But the different categories of changes determined by the LDA model selection criteria may not be all significant. In order to discard non-significant ones, in stage **4** the system analyzes both burst regions and topic ratios for each category of changes in order to determine their importance. The significant categories of changes are only those whose topic ratios are above a specified threshold (λ_{th}) and their burst regions are found as well. In stage **5**, only the significant categories of changes are considered for generating the intermediate summaries. These summaries are another level of changed information in concise form which becomes available to the user. In stage **6**, the system considers all intermediate summaries and generates a top summary on the basis of the defined policy.

4.2.1 Extraction of Changes

Let us assume that for any article $\mathcal{A}$ and a given time range, there are T document versions defined as $A = \{rev_1, rev_2, \ldots, rev_T\}$. Here, rev_1 is the latest document version and rev_T is the oldest document version in the given time frame for $\mathcal{A}$. All changes are extracted by comparing each pair of consecutive versions in two modes, word-diff and block-diff [KNR15], yielding $T-1$ diffs for each mode. The set of changed words obtained by the word-diff process is used to generate a feature file for LDA model. This feature file has a total of $M = T-1$ feature vectors where each feature vector $\mathbf{w}_i$ consists of a sequence of words say, $(w_1, w_2, \ldots, w_{j-1}, w_j, w_{j+1}, \ldots)$. Each word w_j in the sequence belongs to any of the words from a set of V distinct words $BOW = \{w^{(1)}, w^{(2)}, \ldots, w^{(V)}\}$, which is directly created from the set of changed words.

4.2.2 Detection of Candidate Categories of Changes

In this section, we describe how to determine the number of candidate categories of changes in terms of latent topics in the LDA model using model selection criteria.

Detection the Number of Candidate Categories

The theoretical part of the LDA model is already discussed in Section 3.3.3. Given the values of α and β hyper-parameters, the problem of choosing the appropriate value for K topics in the LDA model is a problem of model selection, which is addressed by using a standard method in Bayesian statistics [GS04]. According to Bayesian statistics, the key constituent for choosing a model among a set of statistical models is to compute the posterior probability of that set of models given the observed data. This posterior probability is the likelihood of the data given the model, integrating over all parameters in the model. In this problem, the data are the changed words in the revisions of an article, $\mathbf{w}$, and the model is specified by the number of topics, K, so it needs to compute the likelihood of $P(\mathbf{w}|K)$. This likelihood can be computed from Equation 56 and Equation 59, where the first term in Equation 56 on the right hand side is a likelihood and the second is a prior. Now, the logarithm of the likelihood, $P(\mathbf{w}|K)$ requires summing over all possible assignments of words over K latent topics.

$$\begin{aligned}
\log P(\mathbf{w}|K) &= \sum_{i=1}^{V}\sum_{j=1}^{K} \log(P(w_i|z_i = j, \mathbf{z}_{-i}, \mathbf{w}_{-i})) \\
&= \sum_{i=1}^{V}\sum_{j=1}^{K} \log\left(\frac{n_{-i,j}^{(w_i)} + \beta}{n_{-i,j}^{(.)} + V\beta}\right) \\
&= \sum_{i=1}^{V}\sum_{j=1}^{K} \log\left((\phi_{ji})_{K\times V}\right) \qquad (68)
\end{aligned}$$

Estimates of $\log P(\mathbf{w}|K)$ are computed based on all changed words across all revisions of an article by changing the number of topics K. The model is accounted for the best when it is rich enough to fit the information available in the data. Alternatively, the value of K for which $\log P(\mathbf{w}|K)$ produces the highest value is considered as an appropriate value for K. However, there is a problem of computing $\log$ when entries of $\phi_{K\times V}$ are zero. Therefore, in practice, only non-zero entries are considered and the value of $\log P(\mathbf{w}|K)$ is further divided by the total number of non-zero entries to normalize it. Hence, the model selection criteria for determining K is defined in normalized form as:

$$\log P(\mathbf{w}|K) = \frac{1}{\mathcal{N}}\sum_{i=1}^{V}\sum_{j=1}^{K} \log\left((\phi_{ji})_{K\times V}\right), \forall \phi_{ji} > 0, \qquad (69)$$

where $\mathcal{N}$ is the total number of non-zero entries in $\phi_{K\times V}$. As an example, Figure 11 shows the $\log P(\mathbf{w}|K)$ (in Equation 69) values for different values of K on the Wikipedia article for *Narendra Modi* and it gives the maximum value at $K = 3$.

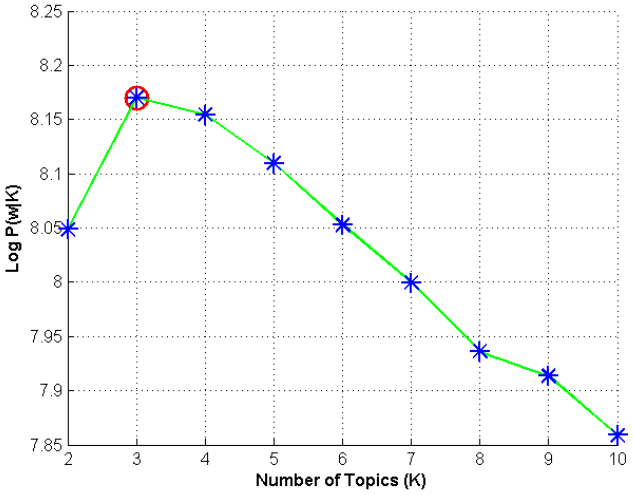

Figure 11: Model selection results showing the log-likelihood of the data for different values of K, for the Wikipedia article on *Narendra Modi* for the month of May, 2014

4.2.3 Burst Detection

In general, burst detection finds the elevated occurrence of activities over time. Burst detection algorithms have been defined in different ways in various domains [Mur99; Kle03; LBK09; HP10]. For example, in the context of collaborative editing of Wikipedia articles [GPKZSN13], burst is defined as an indicator of where the number of edits suddenly increases within a short period of time and can be detected using an elastic burst detection algorithm [ZS03]. In our study, instead of considering all categories of updates, a burst is defined on the basis of the two following propositions:

1. A burst region is a region of diffs where the changes of a same category are concentrated.
2. A high number of changes of the same category observed within a short time span indicates a burst.

In order to incorporate these two propositions, our burst detection algorithm is proposed as follows:

Step 1:

The number of different categories of changes (K) is detected using the LDA model selection criteria, as described in Section 4.2.2.

Step 2:

Each diff ($diff_t : t = 1, 2, \ldots, T-1$) produced by block-diff is marked with a specific category of changes using cosine similarity. Let, $topic_k = \{w_{k1}, w_{k2}, \ldots, w_{kn}\}$ be a set of words associated with the k^{th} topic produced by the LDA model. Let $F = (f_1, f_2, \ldots, f_n)$ be the feature vector corresponding to $topic_k$. The dimension of the feature vector F is the number of words associated with $topic_k$. It is important to mention that the top-n

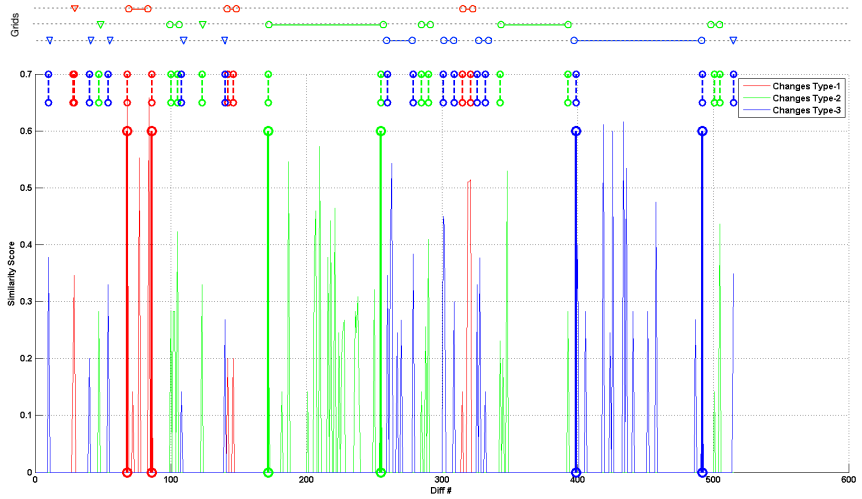

Figure 12: Three different change categories (K = 3) are marked with colors red, green and blue for the Wikipedia article on *Narendra Modi* for the month of May, 2014. There are a total of 21 grids which are shown with the vertical dotted lines at top. The most significant grids from each change category are shown with the solid vertical lines at the bottom.

words are considered for each topic to make the vector dimension equal. For an input word w_i, the component f_i of F is defined as:

$$f_i = \begin{cases} 1 & \text{if } w_i \in topic_k \\ 0 & \text{otherwise.} \end{cases} \tag{70}$$

Obviously, $\forall i,\ f_i = 1$ when the feature vector is constructed for a topic. However, if a word w_i in a diff belongs to the topic then f_i is equal to the number of occurrences of the word w_i in the diff. If F and F′ are the feature vectors for a topic and a diff, respectively, the cosine similarity between the two vectors is defined as:

$$\cos(F, F') = \frac{F.F'}{||F||||F'||} \tag{71}$$

Each diff will get a similarity score from each topic; the diff is then marked with the topic (or category of changes) for which it gets the highest score. It is important to note that, in order to find the highest score for each diff, instead of inspecting a single point ($diff_t$) from each topic, the score is obtained with a certain number of preceding and following points. A Gaussian filter with size 5 is used for this purpose. So, the score ($diff_t{}^{(s)}$) of a point ($diff_t$) is the Gaussian weighted sum of the scores of the observed point, the two preceding and the two following points. The importance of using a Gaussian filter is to remove some additional noise. The distribution of the diffs marked with different categories of changes is shown in Figure 12 for the Wikipedia article on *Narendra Modi*. In this figure, three different categories of changes (K = 3) are marked with three colors: red, green and blue. The number of categories of changes (K) is detected automatically in **Step 1**, using the LDA model selection criteria.

Step 3:

The whole time range is divided into a number of disjoint grids. A grid is a sequence of diffs marked with a particular category of changes. Let us assume that there is a set of grids, denoted by $R = \{\bigcup_{k=1}^{K} R_k\}$, where $R_k = \{R_{k1}, R_{k2}, \ldots, R_{kp_k}\}$ is a subset of grids in which each grid contains only diffs of the k^{th} category. The total number of grids across all categories of changes K is $\sum_{k=1}^{K} p_k$. For each grid, a score is calculated by summing the cosine similarity scores of all diffs that belong to that particular grid. In order to normalize the score of a grid, each grid's score is divided by the maximum score obtained from all grids. Denoting $R_{ki}^{(s)}$ as the score of the grid R_{ki}, then $R_{ki}^{(s)} = \frac{\sum_t diff_t^{(s)}}{\max_{k,i} R_{ki}^{(s)}}$.

Figure 12 illustrates the first three steps of the burst region detection, using the Wikipedia article on *Narendra Modi*. The diffs are represented on the X-axis, while the cosine similarity values for each diff are presented along the Y-axis. Each diff is colored according to its change category; in this case, there are three change categories ($K = 3$). At the top of the figure, there are several horizontal lines representing the detected grids for each change category. A triangle represents a grid containing a single diff, while two circles connected by a line represent a grid containing several diffs of the same category. Thus, the total number of grids is $\sum_{k=1}^{3} p_k = (4+7+10) = 21$. The grids with the highest score for each change category (i.e. the most significant) are highlighted with solid vertical lines.

Step 4:

In this step, the second proposition is incorporated in order to give more importance to the region of diffs where more diffs occurred within a short period of time. Let R_k be the subset of all grids for the k-th category (referring to the example on Figure 12, R_1 would be the set of 4 grids of Change Type-1). For every grid in R_k, where $R_k = \{R_{k1}, R_{k2}, \ldots, R_{kp_k}\}$, the coordinates of its center point are denoted as $CP_{ki} = (x_{ki}, y_{ki})$; x_{ki} corresponds to the diff which is the middle point of grid R_{ki}, and $y_{ki} = R_{ki}^{(s)}$.

So far, no temporal information has been used. In order to incorporate the second proposition, another set of center points and their corresponding scores are calculated. For this purpose, a temporal window is fitted over each grid R_{ki}, centered at CP_{ki}. For example, if the temporal window spanning across, say, 5 days, then it starts 2 days before the day of the diff at x_{ki}, includes the day of that diff, and extends to the 2 days after that day. The similarity scores of the diffs contained in this temporal window that belong to k-th category are then summed to yield a temporal score for R_{ki}. This score is then normalized by the maximum score obtained from all grids. Let $CP'_{ki} = (x'_{ki}, y'_{ki})$ be the temporal center point for grid R_{ki}. The value of x'_{ki} is that of the diff located at the middle point of the temporal window of grid R_{ki}, while y'_{ki} is equal to the normalized temporal score for R_{ki}.

So far, the two sets of center points of all grids have been calculated. In order to generate the scores for all other points, in which each point corresponds to an individual diff, a polynomial of degree 1 is fitted successively to each pair of consecutive center points. This fitting process is performed for the two sets of center points calculated earlier. Let P_k be the piecewise

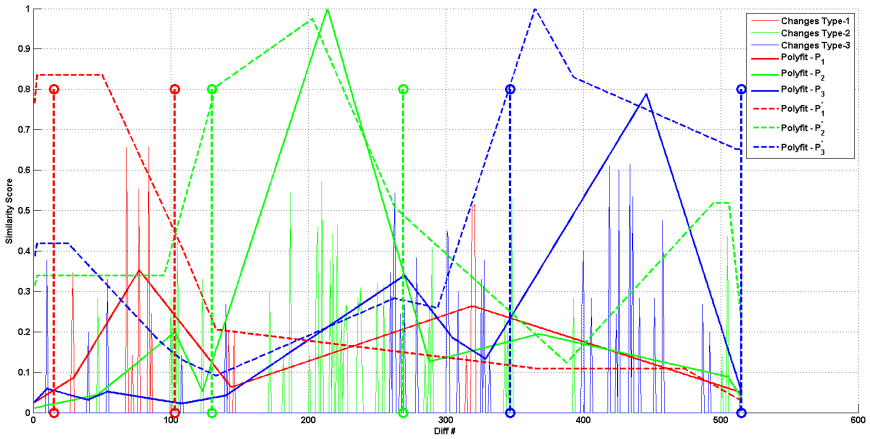

Figure 13: Three different change categories ($K = 3$) are marked with colors red, blue and green for the Wikipedia article on *Narendra Modi* for the month of May, 2014. Overlaying the curves for P_k (solid) and $P_k^{'}$ (dashed) for the three different change categories ($K = 3$) over the cosine similarity scores for each diff. For each change category, the detected burst region is shown in the figure within the dashed vertical lines; the detected bursts are $R_1^{'} = [15, 103]$, $R_2^{'} = [130, 269]$ and $R_3^{'} = [347, 515]$ for red, green and blue categories respectively.

polynomial which is obtained after concatenating all individual polynomials of degree 1. Therefore, P_k represents the scores of all points throughout the set of grids R_k. Similarly, $P_k^{'}$ is another piecewise polynomial for representing the temporal scores for all points throughout the same set of grids. This process is repeated for all change categories. It is important to note that, if the polynomial P_k is not defined for the entire range specifically the points before x_{k1} or the points after x_{kp_k} then, for those points the intermediate values are generated by

$$\frac{y}{1 + \log(|x - x^{'}| + 1)} \tag{72}$$

where $x^{'} = x_{k1}$ and $y = y_{k1}$ for any intermediate point x before x_{k1} and similarly, $x^{'} = x_{kp_k}$ and $y = y_{kp_k}$ for any intermediate point x after x_{kp_k}.

Figure 13 shows the results of this process for the Wikipedia article on *Narendra Modi*. In this example, there are three categories of changes ($K = 3$). For every category of changes, there are two curves, one in a solid line and one in a dashed line. The solid line represents P_k and the dashed line represents $P_k^{'}$.

Step 5:

This step uses the two series of piecewise polynomials P_k and $P_k^{'}$ ($k = 1, 2, \ldots, K$) which are obtained in the previous step. We assume that all the dominant polynomial segments are candidates for the identification of burst regions. In order to find these segments, we consider a masking array $[a_k]$

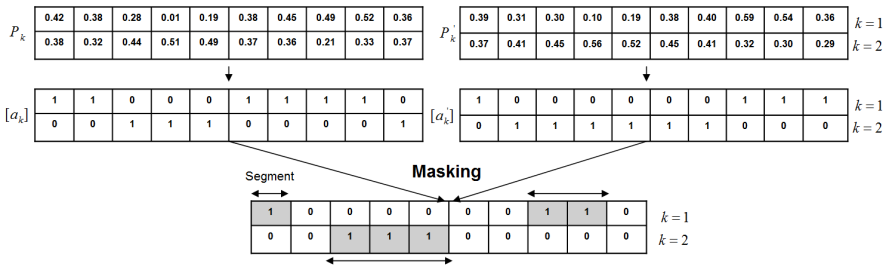

Figure 14: A prototype example illustrating the detection of candidate burst segments. There are two series of polynomials P_k and P'_k ($k = 1, 2$) are shown. The values of $[a_k]$ are generated for $k = 1$ and $k = 2$ using Equation 73 from P_k polynomials. Similarly, the values of $[a'_k]$ are generated for $k = 1$ and $k = 2$ using Equation 73 from P'_k polynomials. The masking technique is then applied between $[a_k]$ and $[a'_k]$ and as an output the resultant array is shown with two candidate burst segments for $k = 1$ and one candidate burst segment for $k = 2$.

of size $T-1$ or explicitly, $[a_{kt}]$ ($t = 1, 2, \ldots, T-1$) for each k. Now, for a particular k, we assign the value of each cell in the array as

$$a_{kt} = \begin{cases} 1 & \text{if } k = \max\limits_{k} P_k(\mathit{diff}_t) \\ 0 & \text{otherwise.} \end{cases} \tag{73}$$

In the array, a series of consecutive 1's immediately trailed and followed by zeroes is considered as a single segment. For ease of computation, we assume that there are two extra 0's before the beginning and after the ending of the arrays. Similarly, for a particular k, we compute another masking array $[a'_k]$ of size $T-1$ using the piecewise polynomial P'_k. To find the common segments a masking technique is used between $[a_k]$ and $[a'_k]$, and the segments are then obtained by observing a series of 1's immediately trailed and followed by zeroes in the resulting array. These segments are considered as the candidate burst segments for a change category k. The above procedure is applied for all change categories ($k = 1, 2, \ldots, K$). Figure 14 illustrates how the candidate burst segments are obtained for two change categories ($K = 2$) with a prototype example.

Let us assume that there is a set of candidate burst segments obtained after the end of masking procedure, denoted by $R' = \{\bigcup_{k=1}^{K} R'_k\}$, where $R'_k = \{R'_{k1}, R'_{k2}, \ldots, R'_{kp_{k'}}\}$ is a subset of candidate burst segments where each one is associated with a particular change category k. To compute the score of each candidate burst segment (which can be seen as a grid), a procedure similar to **Step 3** is applied, but the score of an individual diff is obtained from the polynomial $P'_k(\mathit{diff}_t)$. If $R'^{(s)}_{ki}$ is the corresponding score of the segment R'_{ki}, then $R'^{(s)}_{ki} = \sum_t P'_k(\mathit{diff}_t)$. Finally, the segment with the maximum score for each change category is chosen as its burst region. In general, there are K bursts for K different change categories, but there can be cases where no burst region is found for a category. The K bursts are denoted as $B_k = R'^{(s)}_k = \max\limits_i R'^{(s)}_{ki}$ ($k = 1, 2, \ldots, K$). Moreover, to define a complete burst, additional information such as time information ($\mathcal{T}$) and the

valid diffs ($\mathcal{D}$) within the time range is included. Any non-empty diff is considered as a valid diff. Therefore, if there is a burst region for a particular k, it is defined as $B_k = (R'_k, R_k^{'(s)}, \mathcal{D}_k, \mathcal{T}_k)$.

4.2.4 Identification of Significant Categories of Changes

The different change categories/topics determined by LDA model selection criteria may not necessarily be all significant. The reason is that when there is substantially less evidence for a topic than for other topics, the former is considered non-significant. Consequently, there is a possibility of not finding any burst region for this topic as it is dominated by other topics with stronger evidence. We therefore define the following propositions in order to determine whether a topic is significant or not.

1. If no burst region is found for a topic, it is considered as a non-significant topic.
2. If there is a burst region for a topic but the corresponding evidence is substantially less than the evidence for the others, it is considered as a non-significant topic.

In order to identify whether the evidence of a topic is substantially less or not, the topic proportion probabilities are analyzed in terms of topic ratios.

Topic Ratios

In Gibbs LDA, the topic proportion probabilities are calculated by counting the number of words from a particular document (in our case, a particular diff) assigned to a particular topic as:

$$P(z_i = j|\mathbf{z}_{-i}) = \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,.}^{(d_i)} + K\alpha} \tag{74}$$

where, $n_{-i,j}^{(d_i)}$ is the number of words from document d_i assigned to topic j, not including the current one, and $n_{-i,.}^{(d_i)}$ is the total number of words in document d_i, not including the current one and $\mathbf{z}_{-i}$ is the assignment of all other topics except topic j. However, the probability of a particular topic is computed in terms of the $\theta_{(M\times K)}$ matrix as follows:

$$P(z_i = j) = \frac{\sum_{d=1}^{M} \theta(d,j)}{\sum_{j=1}^{K}\sum_{d=1}^{M} \theta(d,j)} \tag{75}$$

A higher topic proportion probability value indicates a more significant topic. For each topic, its topic ratio is computed by dividing its own topic proportion probability by the highest one among all topic proportion probabilities:

$$\lambda_j = \frac{P(z_i = j)}{P(z_i = j)|_{max}}, j = 1, 2, \ldots, K, \lambda_j \in [0, 1] \tag{76}$$

The topic ratio value for the most significant topic is always equal to 1. A topic is considered significant if $\lambda_j \geqslant \lambda_{th}$ i.e., the value of λ_j is larger than a threshold called topic ratio constant.

4.2.5 Intermediate Summary Generation

Let us assume that $\{z_i^{(R)} : i = 1, 2, \ldots, c; c \leqslant K\}$ is a set of significant topics in ranked order that satisfy two criteria: (i) each topic ratio λ_i corresponding to $z_i^{(R)}$ satisfies $\lambda_i \geqslant \lambda_{th}$ and (ii) for each $z_i^{(R)}$, there exists a non-empty burst $B_i = (R_i', R_i^{'(s)}, \mathcal{D}_i, \mathcal{T}_i)$. For all significant topics, an intermediate summary is generated. To generate the i-th intermediate summary for the topic $z_i^{(R)}$, each sentence from each diff in $\mathcal{D}_i$ is scored using the $\phi_{K\times V}$ matrix from LDA model. The matching terms between each sentence and the significant topic, $z_i^{(R)}$ will get a score from the value of ϕ_{ij}, where ϕ_{ij} is the j-th term of topic $z_i^{(R)}$ in the $\phi_{K\times V}$ matrix. The scores of the non-matching terms are zero. The sentence score is now calculated by summing the scores of all terms and dividing by the number of terms. All sentences are ranked in descending order of their scores. Finally, the top ranked sentences are presented as an intermediate summary for topic $z_i^{(R)}$ with time range $\mathcal{T}_i$. The sentence selection considers the burst region of diffs $\mathcal{D}_i$ within a short time range $\mathcal{T}_i$ for the generation of an intermediate summary of topic $z_i^{(R)}$. Thus, it is likely to pick up the sentences which are closely related to each other, instead of selecting sentences from a large set. By selecting sentences from the burst region of diffs, the resulting summary is likely to be coherent since it avoids unnecessary noise.

4.2.6 Top Summary Generation

The aim is to generate a single summary at the top-level in order to consolidate the most significant changes within the given time period. The top summary is created based on a weighted linear combination of all significant topics. Let us assume that $(w_1, w_2, \ldots, w_c)$ are the weights of c significant topics, where w_i is associated with topic $z_i^{(R)}$ and $\sum_{i=1}^{c} w_i = 1$. The value of w_i is computed on the basis of a certain policy: (i) if a topic $(z_i^{(R)})$ is identified as the most significant topic according to both topic ratio (λ_i) and the highest burst region score $(R_i^{'(s)})$, this topic will be considered for generating the top summary. In this scenario, the weight corresponding to this topic is 1 and the weights for rest of them are 0. (ii) Otherwise, the weights are computed from their corresponding topic proportion probabilities. If p_i is the topic proportion probability of $z_i^{(R)}$ then $w_i = \frac{p_i}{\sum_{i=1}^{c} p_i}$. Each significant topic $z_i^{(R)}$ gives a score to a sentence as described in Section 4.2.5 and the score is further multiplied by the corresponding weight w_i. Finally the sentence is assigned with the maximum score. A certain number (depending on the intended size of the top-level summary) of top-ranked sentences obtained from each intermediate summary are then selected for ranking. The few highest ranked sentences are presented as a top summary.

4.3 EVALUATION FRAMEWORK

It is a common practice to evaluate a system generated summary against a reference summary using an evaluation metric. Though ROUGE met-

rics [LH03] give the flexibility to the user in order to provide multiple reference summaries against a system-generated summary however, in those cases, an automatic mapping between system and reference summaries is not required, as all summaries refer to the same topic. We therefore propose a framework that can deal with multiple system summaries generated from different topics. This is usually a difficult task as, in practice, a one to one mapping between the reference and system summaries is not provided a priori. An automatic mapping technique called best match mapping (BMM) is proposed in our framework. In BMM, a reference summary is mapped with the system summary that gives the highest score.

Let $S^{(R)} = \{S_i^{(R)} : i = 1, 2, \ldots, m\}$ and $S^{(A)} = \{S_j^{(A)} : j = 1, 2, \ldots, n\}$ be the set of reference and system summaries, respectively. The best match $BMM(S_i^{(R)}, S^{(A)})$ for a reference summary $S_i^{(R)}$ in a set of system generated summaries $S^{(A)}$ is defined as:

$$BMM(S_i^{(R)}, S^{(A)}) = \max_j \{\mathcal{M}(S_i^{(R)}, S_j^{(A)})\} \tag{77}$$

where $\mathcal{M}$ denotes any evaluation metric. The match between $S^{(R)}$ and $S^{(A)}$ is defined as:

$$BMM(S^{(R)}, S^{(A)}) = \frac{1}{m} \sum_{i=1}^{m} BMM(S_i^{(R)}, S^{(A)}) \tag{78}$$

The advantage of this framework is that any existing evaluation metric can be used to find the appropriate mapping. At the same time, this matching score is used for the final evaluation. In this study, ROUGE [LH03] and normalized cosine similarity evaluation metrics are used.

4.3.1 ROUGE

In the evaluation, the system-generated summaries obtained for the different topics are compared with human-created summaries (reference summaries) using ROUGE metrics [LH03]. ROUGE metrics are widely used by DUC and TAC. These metrics automatically measure the quality of a summary by counting the number of overlapping words between the system-generated summary and a reference summary. Intuitively, higher ROUGE scores indicate that the system-generated summary and the human created summary are more similar. There are different ROUGE measures: ROUGE-N, ROUGE-L, ROUGE-W, ROUGE-S and ROUGE-SU. ROUGE-N is an n-gram recall between a candidate summary and a set of reference summaries. Let $S^{(R)}$ be a set of reference summaries; the ROUGE-N score of a system generated summary is computed as follows:

$$\text{ROUGE-N} = \frac{\sum_{S \in \{S^{(R)}\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{S^{(R)}\}} \sum_{gram_n \in S} Count(gram_n)} \tag{79}$$

where $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in a system generated summary and a reference summary and $Count(gram_n)$ is the number of n-grams in the reference summary. ROUGE-1 and ROUGE-2 metrics of ROUGE-N are used with the length of n-gram as $n = 1$ and $n = 2$, respectively. There are other ROUGE metrics used in our evaluation. ROUGE-L, which measures the LCS between a system-generated summary and a reference summary. ROUGE-W, which is similar to ROUGE-L

except that it is based on weighted LCS where the weighting function is $f(L) = L^{weight}$ and L indicates the length of the LCS; in this evaluation the *weight* parameter is given as $weight = 1.2$ i.e., the metric ROUGE-W-1.2 is calculated. ROUGE-S measures the overlapping of skip-bigrams, given a maximum gap length between two words; in this evaluation, the maximum gap length is given as 4 i.e., ROUGE-S4 is calculated. ROUGE-SU4 is similar to ROUGE-S but with the addition of unigram as a counting unit; in this evaluation, the maximum gap length between two words is given as 4. Each of the ROUGE metrics has three scores (recall, precision and F-measure); in this evaluation, F-measure (the harmonic mean of precision and recall) scores are reported for ROUGE-1, ROUGE-2, ROUGE-L, ROUGE-W-1.2, ROUGE-S4 and ROUGE-SU4.

4.3.2 Normalized Cosine Similarity

Let us assume that $S_i^{(R)}$ and $S_j^{(A)}$ are the reference summary and system-generated summary, respectively. In order to calculate the cosine similarity between the two summaries, a feature vector is created from each of them. The size of each feature vector is the total number of unique words present in both summaries excluding the stop words. The value of a component of a feature vector is the frequency of corresponding word in that particular summary. Let $F^{(R)}$ and $F^{(A)}$ be the corresponding feature vectors for the summary $S_i^{(R)}$ and $S_j^{(A)}$ respectively. If $L^{(R)}$ and $L^{(A)}$ are the number of words, excluding the stop words, present in $S_i^{(R)}$ and $S_j^{(A)}$, respectively, the normalized cosine similarity score is defined as:

$$n\text{-}cos(S_i^{(R)}, S_j^{(A)}) = \frac{min(L^{(R)}, L^{(A)})}{max(L^{(R)}, L^{(A)})} \cdot \frac{F^{(R)}.F^{(A)}}{||F^{(R)}||||F^{(A)}||} \tag{80}$$

The min-max ratio is multiplied with the cosine similarity score in order to handle the differences in length for the summaries.

4.4 EXPERIMENTAL SETUP

The experiments to validate the *MultiSummar* system were organized as follows. The dataset consists of 54 case studies on the revision histories of 49 distinct Wikipedia articles within different time periods. Some articles are chosen more than once by selecting several time periods. The articles, along with the time periods, are selected in a way such that (i) the valuable changes that were made to the revisions of an article should be present within the chosen time period and (ii) the reasons for the occurrence of those changes are known to us. A detailed description of the dataset was reported in Table 3 & Table 8.

4.4.1 Validating K in the Context of Summarizing Changes

Finding the appropriate number of topics (K) for an optimum model is still an open issue. To the best of our knowledge, there is no standard practice to estimate the appropriate number of topics in LDA model. As mentioned in Section 4.2.2, the value of K is decided when the model is defined as the best to fit the information available in the data. The value of K depends on the

choice of α and β and is also affected by the inclusion of specific datasets. However, K is determined in a setting where a symmetric Dirichlet prior on θ with $\alpha = 0.5$ and a symmetric Dirichlet prior on ϕ with $\beta = 0.1$ are used. In this study, the objective is not only finding the appropriate value of K for LDA model, but also building a model using that K, which can produce a better summary of changes within a time range for the chosen article. The experiments are performed on 54 case studies for 49 distinct Wikipedia articles within different time periods. These 54 case studies are selected in such a way that there can be exactly one significant change made to an article within the chosen time period. Based on this assumption, it is expected that the appropriate number of latent topics can be found in between 2 and 10 i.e., $K \in [2, 10]$. Therefore, we computed an estimate of $\log P(\mathbf{w}|K)$ (using Equation 69) for K values from 2 to 10 topics for each selected article. Figure 11 illustrates the log-likelihood values against the number of topics for the Wikipedia article on *Narendra Modi* for the month of May, 2014. For this article, $\log P(\mathbf{w}|K)$ initially increases as a function of K, reaches a maximum at $K = 3$ and then it decreases. Therefore, the model with 3 topics is likely to be the best fitted model for this article.

An analysis is carried out to determine whether the best fitted model eventually produces a better summary. However, for this kind of application it is difficult to assess which summary is better when compared to other summaries produced by different LDA models. In this study, the ROUGE metrics in particular ROUGE-1, is used to compare different summaries. Let $K^{(R)}$ be the expected value of K, for which the model gives the maximum ROUGE score, and let $K^{(L)}$ be the value of K for the best-fitted model which is obtained by the log-likelihood criteria (see Equation 69). Now, our goal of generating better summaries will be achieved ideally if both $K^{(L)}$ and $K^{(R)}$ are the same for all articles. However, from the experiments it is observed that in many cases the two values are not the same, the value of $K^{(L)}$ is either larger or smaller than the expected value of $K^{(R)}$. Our experimental results are shown in Table 13. We assume that $K^{(L)}$ and $K^{(R)}$ are the same when $|\mathrm{ROUGE}(K^{(L)}) - \mathrm{ROUGE}(K^{(R)})| < \epsilon$, where ϵ is a threshold. In this condition, $\mathrm{ROUGE}(K^{(L)})$ is the ROUGE-1 score between a system-generated summary obtained by the LDA model for $K = K^{(L)}$ topics and the corresponding reference summary, and $\mathrm{ROUGE}(K^{(R)})$ is the same but the system-generated summary is obtained by the LDA model for $K = K^{(R)}$ topics. In Table 13, for $\epsilon = 0.05$, 11 articles are found where $\mathrm{ROUGE}(K^{(L)})$ is less than $\mathrm{ROUGE}(K^{(R)})$, $\mathrm{ROUGE}(K^{(L)})$ and $\mathrm{ROUGE}(K^{(R)})$ are the same for 31 articles, and 12 articles are found where $\mathrm{ROUGE}(K^{(L)})$ is greater than $\mathrm{ROUGE}(K^{(R)})$. The statistics are also made for $\epsilon = 0.06$ and $\epsilon = 0.07$. This statistics indicates that the likelihood value $K^{(L)}$ is one of the good choices for finding the appropriate K for producing better summaries.

Table 13: Statistics of expected K ($K^{(R)}$) and log-likelihood K ($K^{(L)}$) on 54 case studies for 49 distinct Wikipedia articles

	$K^{(L)} < K^{(R)}$	$K^{(L)} = K^{(R)}$	$K^{(L)} > K^{(R)}$
Count ($\epsilon = 0.05$)	11	31	12
Average ROUGE ($\epsilon = 0.05$)	0.371312	0.486322	0.369312
Count ($\epsilon = 0.06$)	10	33	11
Average ROUGE ($\epsilon = 0.06$)	0.338648	0.490196	0.366643
Count ($\epsilon = 0.07$)	8	36	10
Average ROUGE ($\epsilon = 0.07$)	0.359460	0.475901	0.357361

4.4.2 Finding the Topic Ratio Constant (λ_{th})

Since we are looking for the major changes which occurred in the given time periods, it is necessary to filter out the changes which are beyond interest. It is already assumed that each topic is most likely to carry one particular type of change. Hence, we need to determine which ones are the most interest topics. In this study, the value of λ_{th} (specified in Section 4.2.4) is chosen depending on how many top ranked topics need to be incorporated in the generation of the summaries. For this purpose, we examine the overall ROUGE scores on the same 54 case studies for different values of λ_{th}, which is shown in Figure 15. In this figure, it is observed that the maximum overall ROUGE scores is attained at $\lambda_{th} = 0.6$ and the ROUGE scores do not increase further for the values of λ_{th} which are greater than 0.6. This indicates that the topics which are included as extra for $\lambda_{th} > 0.6$ do not contribute to the generation of better summaries. Thus, the value of the topic ratio constant, λ_{th} is considered as 0.6.

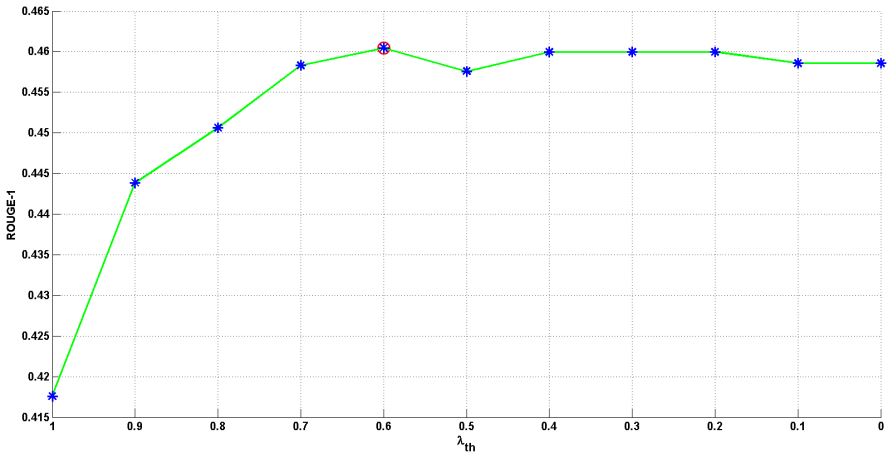

Figure 15: λ_{th} selection on 54 case studies for 49 different Wikipedia articles

4.4.3 Top Summary Evaluation

The aim of top summary generation is to provide a single summary that consolidates the most significant changes within the given time period. In general, it seems like the highest ranked topic (based on the topic ratios) may carry the most significant changes. However, in practice, this is not always true, and the significant changes can come from other than the highest ranked topic. This statement can best be substantiated with the following experiments, performed with the previously described dataset. Due to other surrounding changes of the main type of changes, the number of topics is not considered by default as 2, but rather it is detected (using log-likelihood criteria) in between 2 and 4. The summaries are thus generated for each individual topic in order to compare them.

Table 14 shows that 37 cases give the highest ROUGE scores for the summaries which are built with the highest ranked topic. Similarly, the second highest topic gives the highest ROUGE scores in 9 cases, whereas the other topics give it in 8 cases. In spite of the datasets being chosen with the assumption of having one major type of changes, for $(54 - 37) = 17$ cases the

summaries are the best regardless of other top ranked topics. The reason is that if the changes are not very strong, they might be distributed among multiple topics. Moreover, in the cases where multiple types of significant changes occurred in a given time range, it is required that we consider multiple topics instead of a single one. Thus, generating a summary by considering only the highest ranked topic is not always a good idea; more than one top ranked topic (may be, the second or third highest ranked topic) needs to be incorporated as well. Therefore, we defined a policy for generating

Table 14: Statistics of ranked topics ($z_i^{(R)}$) which give the maximum ROUGE scores ($z_i^{(ROUGE_{max})}$) to the summaries for 54 case studies on 49 different Wikipedia articles

# case studies	$z_1^{(R)}$	$z_2^{(R)}$	$z_{i>2}^{(R)}$
54	37	9	8

the top summary (see Section 4.2.6) which is a combination of multiple topics. For each top summary, we build a corresponding reference summary in which the sentences are extracted from the latest version within the given time period of the Wikipedia article instead of writing it manually. Each reference summary is prepared without any ambiguity as it carries one significant change. The overall experimental results for the top summary with two metrics, cosine similarity (in Section 4.3.2) and ROUGE (in Section 4.3.1), are shown in Table 15.

4.4.4 Intermediate Summary Evaluation

To evaluate the intermediate summaries, ideally we need to provide the same number of reference summaries as the number of significant topics detected by the system. However, in practice it is difficult to provide such a set of reference summaries; in the cases where multiple types of significant changes occurred in a given time range it might happen that one type of significant changes overwhelms the others. These types of significant changes are very difficult to separate by the users beforehand. Due to this difficulty, we provide only one reference summary corresponding to the main type of significant changes recognized a priori by the user. This reference summary is automatically mapped with one of the intermediate summaries according to our proposed evaluation framework, as specified in Section 4.3. If the system can appropriately separate one type of significant changes, other significant changes are also likely to be well separated by the system. Consequently, if the system can properly evaluate an intermediate summary, it can indirectly evaluate other intermediate summaries without the explicit use of a set of corresponding reference summaries. The overall evaluation results for the intermediate summaries using two metrics, cosine similarity (in Section 4.3.2) and ROUGE (in Section 4.3.1), are shown in Table 15.

4.5 RESULTS AND DISCUSSION

Our *MultiSummar* system focuses on the generation of abridged and non-redundant accounts of textual changes made to a set of document versions in a hierarchical way. For a given time interval, this system can detect automatically how many categories of changes were made on the document

Table 15: Overall ROUGE scores using both default K and detected K value by the given short time ranges for 54 case studies

System	# case studies	Summary Level	Evaluation Metric						
			n-cos	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-W-1.2	ROUGE-S4	ROUGE-SU4
Default ($LDA_{K=2}$)	54	—	0.47478	0.45978	0.27684	0.43172	0.22362	0.24573	0.28169
Multi-level (LDA_K)	54	Top	0.48395	0.46298	0.27923	0.43316	0.22429	0.24687	0.28319
Multi-level (LDA_K)	54	Intermediate	0.51109	0.47827	0.29319	0.44429	0.22969	0.25539	0.29258

versions. If there are multiple prominent changes, they are likely to be separated by the system. In our previous work [KNR15], different approaches were proposed for summarizing changes and the approach based on LDA model outperformed others. However, in that approach the LDA model was built on the assumption that there were always two topics ($K = 2$), one topic representing the significant changes and another for other associated changes. Our previous system, denoted as Default($LDA_{K=2}$) fits the situation where the number of significant changes is only one for a given interval. For the purpose of evaluation, we chose the dataset in such a way that for a given interval one significant type of changes occurred. Therefore, the results of the Default($LDA_{K=2}$) system is used here as a benchmark result. In this study, we build *MultiSummar* system which we refer as Multi-level(LDA_K) without the constraint of $K = 2$. The effectiveness of this generalized system can be established if it meets the benchmark results on the same dataset. The performance of both Default($LDA_{K=2}$) and Multi-level(LDA_K) with two evaluation metric, namely normalized cosine similarity and ROUGE are shown in Table 15. From this table it is observed that Multi-level(LDA_K) produces a similar result for the top summary. In fact, it is slightly better than the previous one, whereas the performance of the intermediate summaries is improved marginally.

In order to evaluate the burst detection algorithm we could look at the intersection of the detected and manually annotated interval. However, as an alternative way, the evaluation of the intermediate summaries can also prove the efficiency of the burst detection algorithm. Since the performance of the intermediate summaries crosses the benchmark results, it can be said that the proposed burst detection algorithm works satisfactorily. Moreover, the statistics in Table 16 show the effect of burst detection in the use of diffs in sentence selection. It is found that the use of the number of diffs covered by all detected bursts is reduced to 57.84% and the number of diffs for the best match summary is reduced to 88.13 %.

Table 16: Statistics for the use of diffs due to an effect of burst. The statistics are made on 54 case studies on 49 different Wikipedia articles.

# case studies	# diffs	# non-empty diffs	# diffs associated in all detected burst	# diffs associated in best match summary
54	38549	6884	2902	817

While the ROUGE metrics provide an arguable estimate of the similarity between a generated summary and a reference summary, they do not account for other important aspects such as focus (sentences should only contain information that is related to the rest of the summary) or coherence (consistency among sentences) drawn from the DUC manual evalua-

tion guidelines[1]. To evaluate these aspects, we carried out a simple user study through pairwise comparison [CHT11]. In this approach, the human evaluators are presented with randomly-selected pairs of summaries generated by the two systems: Default($LDA_{K=2}$) and Multi-level(LDA_K), as well as a corresponding reference summary. Then, they are asked to mark the better summary in the given pair of system-generated summaries on the basis of focus and coherence. We asked 5 annotators to rate 54 summary pairs for Default($LDA_{K=2}$) vs. Multi-level(LDA_K). The evaluation results in frequencies are shown in Table 17. The annotators rated Multi-level(LDA_K) generated summaries more coherent and focused compared to Default($LDA_{K=2}$), where the results are statistically significant based on paired t-test on 95% confidence level.

Table 17: Frequency results of manual user evaluation through pairwise comparison. Tie indicates evaluations where two summaries are rated equal.

Aspect	Default(**$LDA_{K=2}$**)	Multi-level(**LDA_K**)	Tie
Focus	14	30	10
Coherence	12	36	6

Table 15 shows that the results produced by the Multi-level(LDA_K) system are not significantly improved compared to the Default($LDA_{K=2}$) system. At first, it may seem that the Default($LDA_{K=2}$) system is a better choice with respect to the complexity of the Multi-level(LDA_K) system. This may be true when the number of significant changes is 1 for a given time interval, but the proposed system is developed to deal with the cases where multiple significant changes occurred. However, it is not easy to create a database with a large number of such examples. In the following case study, we use such an example to demonstrate how the proposed system works in order to highlight the advantages of the Multi-level(LDA_K) system.

4.5.1 A Case Study

Three different short time periods, each spanning one month, are chosen beforehand for the Wikipedia article on the late USA adventurer *Steve Fossett* with article ID 186642. The first time period is the month of September, 2007 where the important changes are related to the fact that Fossett was reported missing; the second is the month of February, 2008, where the important changes related to Fossett being declared as dead; the third is the month of October, 2008, where they concerned the identification of the wreckage of the airplane where Fossett travelled, as well as of other personal items found near Mammoth Lakes, California. We choose the whole time period from September, 2007 to October, 2008 as the given time range instead of choosing three short intervals separately. This case study helps us understand the scenarios where the detection of the number of topics is required. The first scenario is to present five top ranked sentences while the number of topics is assumed as 2 by default: one topic for all important changes and another for the non-important ones. The second scenario is to present five top ranked sentences while the number of topics is automatically detected as 4.

Table 18 presents top five sentences generated by our previous system (Default($LDA_{K=2}$)) from the Wikipedia article on *Steve Fossett* between September, 2007 and October, 2008. From the experiments, it is observed that one topic reflects the wreckage related changes and another topic represents the

1 http://duc.nist.gov/duc2007/quality-questions.txt

changes other than wreckage related. However, when we consider both top-

Table 18: Sentences are selected by different topic-ID's from the Wikipedia article on *Steve Fossett* between September, 2007 and October, 2008 by giving the default number of topics as 2

topic-ID 1 & topic-ID 2 ($z_i = 1$ & $z_i = 2$)			
#	score	topic (z_i)	sentence
1	0.0058	2	No plane wreckage found.
2	0.0055	2	On October 2nd, 2008, human remains were purportedly found near the wreckage site.
3	0.0034	2	On October 2, ground searchers found human remains, but they have not yet been confirmed to be Fossett's.
4	0.0033	2	On September 30, 2008, hikers found personal items suspected of belonging to Fossett near Mammoth Lakes, California.
5	0.0033	2	No human remains were found and officials doubt anyone would be able to walk away from the crash.

ics for sentence ranking, the sentences from the dominant topic take place in the top positions. Since the wreckage related topic appears as a dominant topic, the summary in Table 18 covers most of the wreckage related sentences. As a result, the information about the other two events is missed in the final summary. This shows the fact that considering the default number of topics as $K = 2$ is not always a good choice. This observation becomes stronger in the following scenarios.

The algorithm for detecting the number of different candidate topics K (described in Section 4.2.2) yielded the maximum log-likelihood value of the data in $K = 4$. Therefore, for this example, we consider that there are four different types of candidate changes in between September, 2007 and October, 2008. In order to find the significant topics from all different candidate topics (described in Section 4.2.4), at first we arrange the topic proportion probabilities, $P(z_i = j)$ in descending order. For this example, the topic proportion probabilities are $\{0.2927, 0.1970, 0.2787, 0.2316\}$ for corresponding topic-ID 1, topic-ID 2, topic-ID 3 and topic-ID 4, respectively. The topic-ID's are then arranged in descending order, becoming $\{1, 3, 4, 2\}$ based on their topic proportion probabilities. The topic ratios $\{\lambda_1, \lambda_3, \lambda_4, \lambda_2\}$ are then computed, where $\lambda_1 = \frac{0.2927}{0.2927} = 1$, $\lambda_3 = \frac{0.2787}{0.2927} = 0.9522$, $\lambda_4 = \frac{0.2316}{0.2927} = 0.7913$ and $\lambda_2 = \frac{0.1970}{0.2927} = 0.6730$. The value of λ_{th} used is 0.6. Since the values for all topic ratios are greater than λ_{th}, all topics are considered significant at this stage. Next, we detect the burst regions, $B_k = (R'_k, R_k^{'(s)}, \mathcal{D}_k, \mathcal{T}_k)$ for each corresponding topic where $k = 1, 2, 3, 4$ (see details in Table 19). We obtain two strong burst regions B_1 ($R_1^{'(s)} = 70.55$) and B_3 ($R_3^{'(s)} = 184.23$) and one weak burst region B_4 ($R_4^{'(s)} = 36.03$). However, there is no burst region found for the topic ID 2, i.e., $B_2 = \Phi$. Therefore, topic-ID 2 is considered at this stage as a non-significant topic though it was initially selected as a significant topic on the basis of topic ratio. The time range detected for the bursts B_1, B_3 and B_4 are $\mathcal{T}_1 = [04-09-2007, 04-09-2007]$, $\mathcal{T}_3 = [29-09-2008, 03-10-2008]$ and $\mathcal{T}_4 = [03-10-2007, 26-02-2008]$ respectively. Similarly, the number of diffs obtained for the bursts B_1, B_3 and B_4 are $N(\mathcal{D}_1) = 24$, $N(\mathcal{D}_3) = 47$ and $N(\mathcal{D}_4) = 15$, respectively.

Table 20 presents top five sentences produced by Multi-level(LDA_K) system for the Wikipedia article on *Steve Fossett* between September, 2007 and

Table 19: Detected Burst details are shown for the Wikipedia article on *Steve Fossett* between September, 2007 and October, 2008

B_k	$P(z_k)$	R'_k	$R'^{(s)}_k$	$\mathcal{D}_k$, $N(\mathcal{D}_k)$	$\mathcal{T}_k$
B_1	0.2927	[1155, 1295]	70.5546	[1743, 1852], 24	[04 – 09 – 2007, 04 – 09 – 2007]
B_2	0.1970	Φ			
B_3	0.2787	[115, 390]	184.2298	[673, 940], 47	[29 – 09 – 2008, 03 – 10 – 2008]
B_4	0.2316	[491, 817]	36.0307	[1127, 1349], 15	[03 – 10 – 2007, 26 – 02 – 2008]

October, 2008. At the intermediate level, three summaries are generated for topics $z_i = 1, 3, 4$ separately.

From the table, it is observed that topic-ID 1 ($z_i = 1$) includes the sentences related to the fact that Fossett was reported missing, whereas topic-ID 3 ($z_i = 3$) and topic-ID 4 ($z_i = 4$) describe the changes of identification of the wreckage of the airplane where he travelled and the declaration of his death, respectively. Topic-ID 4 is not related to a single event since not many substantial changes related to Fossett's death occurred, and hence this topic forms a cluster with mixed types of changes. This is one of the limitations of using the LDA model.

For generating the top summary, all three topics topic-ID 1, topic-ID 3 and topic-ID 4 are used on the basis of our policy (Section 4.2.6) and five top ranked sentences are presented in Table 20. It is observed from the table that four sentences are related to topic-ID 3 whereas one sentence is related to topic-ID 1, but there is no sentence from topic-ID 4 placed at the top . This is due to the fact that both topic-ID 1 ($\lambda_1 = 1, R'^{(s)}_1 = 70.55$) and topic-ID 3 ($\lambda_3 = 0.9522, R'^{(s)}_3 = 184.23$) are more significant than topic-ID 4 ($\lambda_4 = 0.7913, R'^{(s)}_4 = 34.89$).

Earlier, we have used a single reference summary against multiple intermediate summaries for evaluation. In this case study, we provide three reference summaries which are created from three different events: news of Fossett's disappearance, the declaration of Fossett's death and the discovery of wreckage items. Each reference summary is mapped automatically with one of the intermediate summaries in our evaluation framework. All intermediate mapping scores determined by Equation 77 for the three reference summaries are shown in first three rows, whereas the final evaluation scores calculated by Equation 78 are shown in the last row of Table 21 using different metrics.

4.6 SUMMARY

We describe the *MultiSummar* system that can automatically detect multiple significant changes in hierarchical levels within a user-defined time period. The LDA model is used to identify different latent changes in terms of different topics. At the top-level, a single summary is produced that consolidates the most significant changes, whereas each intermediate summary contains the changes of each significant type in detail. Thus, the novelty of this system is that it facilitates the exploration of information at different levels so that a user can use them on the basis of their interest (i.e more generic or more specific).

We also propose a burst detection algorithm that identifies a potential region for each type of changes. Unlike conventional approaches for burst detection, the proposed algorithm is focused only on changes of a similar

Table 20: Sentences are selected by different topic-ID's from the Wikipedia article on *Steve Fossett* between September, 2007 and October, 2008 by the automatic detection of number of topics as 4

topic-ID 1 ($z_i = 1$), Detected time frame by Burst: 04-09-2007 – 04-09-2007, # Diffs: 24		
#	score	sentence
1	0.0071	Steve Fossett was reported missing on September 3, 2007 after taking off in a small plane from an airport near Reno, NV.
2	0.0043	He was last seen taking off in his single-engine plane from the Hilton Ranch just south of Smith Valley.
3	0.0043	At 1:40pm EDT on September 4th 2007 CNN cited the Courier's report that Steve Fossett is missing and search teams are looking for him in the Nevada desert.
4	0.0040	There are currently more than eight aircraft searching for Fossett including aircraft from the Civil Air Patrol and California Highway patrol.
5	0.0035	Fossett took off from the private aircraft strip on the morning of September 3, 2007 and headed south.
topic-ID 3 ($z_i = 3$), Detected time frame by Burst: 29-09-2008 – 03-10-2008, # Diffs: 47		
#	score	sentence
1	0.0130	No plane wreckage found.
2	0.0116	On October 2nd, 2008, human remains were purportedly found near the wreckage site.
3	0.0074	On September 30, 2008, hikers found personal items suspected of belonging to Fossett near Mammoth Lakes, California.
4	0.0073	On October 2nd 2008 a wreckage was found near the town of Mammoth Lakes in California, it was later confirmed to be the wreckage of Steve Fossett's Bellanca Super Decathlon, no body was recovered.
5	0.0073	No human remains were found and officials doubt anyone would be able to walk away from the crash.
topic-ID 4 ($z_i = 4$), Detected time frame by Burst: 30-09-2007 – 16-02-2008, # Diffs: 17		
#	score	sentence
1	0.0020	On November 26, 2007, Fossett's wife requested that Fossett be declared legally dead.
2	0.0019	The search is going to continue.
3	0.0018	Fossett's friend and explorer, Sir Richard Branson has publicly made similar statements.
4	0.0016	As of September 10 , search crews had found eight previously uncharted crash sites, some decades old, but none related to Fossett's disappearance.
5	0.0012	A Cook County, Illinois probate judge today declared wealthy Chicago adventurer Steve Fossett legally dead on 15 February 2008, five months after his plane disappeared.
topic-ID 1 & topic-ID 3 & topic-ID 4 ($z_i = 1$ & $z_i = 3$ & $z_i = 4$)		
#	score	sentence
1	0.0036	No plane wreckage found.
2	0.0032	On October 2nd, 2008, human remains were purportedly found near the wreckage site.
3	0.0021	Steve Fossett was reported missing on September 3, 2007 after taking off in a small plane from an airport near Reno, NV.
4	0.0021	On September 30, 2008, hikers found personal items suspected of belonging to Fossett near Mammoth Lakes, California.
5	0.0020	On October 2nd 2008 a wreckage was found near the town of Mammoth Lakes in California, it was later confirmed to be the wreckage of Steve Fossettś Bellanca Super Decathlon, no body was recovered.

Table 21: Intermediate summaries evaluation using BMM for the Wikipedia article on *Steve Fossett* between September, 2007 and October, 2008

# Reference Summary	Evaluation Metric						
	n-cos	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**ROUGE-W-1.2**	**ROUGE-S4**	**ROUGE-SU4**
1	0.5515	0.50718	0.26087	0.45933	0.22560	0.17143	0.22913
2	0.3552	0.38168	0.17054	0.35114	0.18034	0.12800	0.16976
3	0.3452	0.41490	0.15054	0.40426	0.16455	0.11868	0.16971
BMM score	**0.4173**	**0.43459**	**0.19398**	**0.40491**	**0.19016**	**0.13937**	**0.18953**

kind instead of detecting the burst considering all types of changes together. From the experiments, it is found that the sentence selection range for a particular type of changes drastically narrows down. Since the sentence ranking process is concentrated on a very short time range, the summary is likely to be more focused and coherent. This hypothesis is supported by the experimental results.

We propose an evaluation framework that can deal with multiple intermediate summaries generated from different topics using the BMM metric. The advantage of this metric is that it can use any of the standard evaluation metrics inside. In order to show the effectiveness of the proposed system, the results are compared with the benchmark results. The comparison shows that, although the evaluation scores for the top summary are similar, the performance for the intermediate summaries is improved marginally. The results are not significantly improved for the chosen datasets due to the fact that the benchmark result is obtained for a specific constraint where the number of significant changes is assumed as 1 for a short interval. However, the other aspects of the summaries (focus and coherence) are assessed through pairwise comparison between two systems, proving that the summaries generated from the proposed system are preferred.

5 CONCLUSIONS

Time-biased summarization is an emerging area which brings new challenges to the task of building automatic summaries with respect to time. This has resulted in new types of summaries, and new scenarios in which summaries play different roles. In this thesis, we are focusing on the task of summarization of changes. The goal of this task is to generate abridged and non-redundant accounts of document modifications when dealing with dynamic collections, such as wikis, collaborative documents, or even collections of messages shared in a social network. Previous work on this particular task is relatively scarce. Jatowt et al. [JBI04] first introduced the task of changes summarization in web collections within a limited scope, where only the 'recent, important' changes are considered. Later, Nunes et al. [NRD08] addressed this task in a wider temporal aspect, presenting changes summarization for any user-defined time period. Specifically, we followed the same problem.

The main research question of this thesis is the development of different approaches to produce the significant changes that have occurred in a collection of documents between two dates, as a temporal summary. This main research question is divided into a set of sub-questions which are discussed in Chapter 1. An important challenge for summarizing changes lies in the fact that the significant changes need to be identified for the given period. Here the word 'significant' carries the meaning that this kind of changes should have the potential to be the main reasons for the updates.

5.1 EXTRACTION OF CHANGES

A temporal summary should have all properties (saliency, relevance and non-redundancy) a good summary is supposed to have [LDS13]. Besides those, the inclusion of temporality adds a new requirement. The summary should present the new information, excluding static contents. This is the reason why the information extracted from the dynamic text collection includes only the changes. This led us to formulate the first research question:

- **Q1:** Can we extract information from text such that it includes only the changes made to the text collection within a specified time period?

We have answered this question by extracting the differences from a set of document versions made to an article by comparing the consecutive versions for a given temporal period. In this way, we are able to concentrate only on the changes in the collection of such articles.

5.2 INTERMEDIATE REPRESENTATION OF THE CHANGES

There are three relatively independent stages performed virtually by all classic summarizers [NM12]. As discussed in Chapter 2, the first stage is to derive an intermediate representation which captures the key aspects of the

input text. The four different approaches are proposed, answering the second research question.

- **Q2:** Is it possible to fit a model that can derive an intermediate representation capturing the key aspects of the extracted information?

Each approach shows a different way to derive the intermediate representation that helps further to identify the important content in a summary. Both the first and second approaches follow a frequency-based technique. Whereas, the third approach is a Bayesian topic model-based one, more specifically the LDA model-based. The fourth one is the combination of the third and second. The results show that the LDA model-based approach outperforms the frequency-based approach. The LDA model is used to identify different latent changes in terms of different topics. One of the constraints of using the LDA model is the need to specify the number of latent changes *a priori*. This is overcome by using a standard method in Bayesian statistics [GS04] (Chapter 4).

5.3 SCORE SENTENCES & SELECT SUMMARY SENTENCES

To identify important content, the score of each sentence is determined based on the previously derived intermediate representation. Usually, the scores for all sentences are calculated in such a way that the important sentences are likely to obtain higher scores. Finally, the summarizer has to select the best combination of important sentences to present a meaningful summary. These two stages (as discussed in Chapter 2) led us to address the third and fourth research questions.

- **Q3:** Does the measurement for scoring a sentence identify the significant changes?

- **Q4:** Do other factors help in determining if summary sentences are focused (sentences should only contain information that is related to the rest of the summary) and coherent (there is consistency among sentences)?

We use a simple sentence score measurement which is basically the sum of the scores of all its terms, divided by the total number of terms after excluding the stop words. The summaries obtained by using different approaches are compared with summaries created by humans. When constructing a reference summary for a given time range, the best sentences are selected and extracted from the latest version of the article within the given time period. These manually created reference summaries are provided to compare against the system-generated summaries using ROUGE metrics. The evaluation results can express whether the important sentences are picked or not by the proposed approaches. Intuitively, a higher evaluation scores means the generated and the human-created summaries are more similar. Statistical tests reveal that the differences in ROUGE scores for the LDA-based approach is statistically significant at 99% over the frequency-based approach. A summary of changes is also supposed to be synthetic and therefore avoid redundant information. We use a simple similarity measurement to address this non-redundancy requirement (Chapter 3).

The sentence ranking algorithms do not guarantee the coherence among the selected sentences in a summary. In many cases, the summarizer has additional materials available that can improve the quality of a summary. In this context, a burst detection algorithm is proposed to identify a potential region for each type of changes. From the experiments, it is found that, with the help of burst detection, the sentence selection range for a particular category of changes drastically narrows down with respect to the whole range. Since the sentence ranking process is concentrated on a very short time range, the summary is likely to be more focused and coherent. This hypothesis is supported by a simple user study through pairwise comparison (Chapter 4).

5.4 EVALUATION

It is a common practice to evaluate a system generated summary against a reference summary using an evaluation metric. Though ROUGE metrics [LH03] are flexible enough to allow the input of multiple reference summaries to be tested against a system-generated summary, they do not generate an automatic mapping between system and reference summaries, as all summaries are assumed to refer to the same topic. To deal with multiple topics, we have formulated the fifth question.

- **Q5:** Is it possible to build an evaluation framework that evaluates multiple summaries generated from different topics?

We propose an evaluation framework that can deal with multiple intermediate summaries generated from different topics using the *BMM* metric. The advantage of this framework is that any existing evaluation metric can be used to find the appropriate mapping. At the same time, this matching score is used further as an evaluation score.

5.5 FUTURE RESEARCH

Although a set of articles from Wikipedia has been used with their full revision histories as a document collection, the proposed approaches can be used in other time-dependent collections. For example, any of them can be used to generate a summary of changes for the history of a single web page from a web archive. However, further analysis will be required as we continue research on this topic, while dealing with different kinds of dynamic text collections, such as wikis, collaborative documents, or even collections of messages shared in a social network.

We have proposed a way to automatically detect the number of topics in the LDA model in order to produce the summaries of changes for a wider time interval. However, due to the complex nature of text documents, it might happen that multiple significant changes in the different time spans share the same vocabulary list. In that case the LDA model cannot separate well among these different significant changes. This is one of the limitations of working with the LDA model. In this situation we can divide the larger time range into smaller ones and we can deploy our proposed model to each and every interval.

We have used the burst detection algorithm in order to provide coherent summaries without using sophisticated sentence ranking methods. Addi-

tionally, we can explore other contextual information (for example in case of Wikipedia, DBpedia) to produce higher-quality relevant summaries.

Another limitation of using the LDA model is that when the number of changes (revisions) is not sufficient, the LDA model may fail to identify significant topics appropriately and as a result important terms cannot obtain higher scores. In our proposed approaches, we have extracted and used the diff information. However, in cases where the sequential revisions do not exist we need to use appropriate temporal features for the LDA model or else build the LDA model in such a way that the model itself can capture the temporal changes without an explicit use of differences.

BIBLIOGRAPHY

[ABYG09] O. Alonso, R. Baeza-Yates, and M. Gertz. "Effectiveness of temporal snippets". In: *WSSP Workshop at the World Wide Web Conference—WWW*. Vol. 9. 2009 (cit. on pp. 13, 33).

[ACDGA12] E. Agirre, D Cer, M Diab, and A Gonzalez-Agirre. "Semeval-2012 task 6: A pilot on semantic textual similarity". In: *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012), in conjunction with the First Joint Conference on Lexical and Computational Semantics (SEM 2012)*. 2012, pp. 385–393 (cit. on p. 58).

[ACDYY98] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. "Topic Detection and Tracking Pilot Study: Final Report". In: *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*. 1998, pp. 194–218 (cit. on p. 23).

[ACMSa12] J. Allan, B. Croft, A. Moffat, M. Sanderson, and J. Aslam et al. "Frontiers, Challenges, and Opportunities for Information Retrieval: Report from SWIRL 2012 the Second Strategic Workshop on Information Retrieval in Lorne". In: *ACM SIGIR Forum* 46.1 (2012), pp. 2–32 (cit. on pp. 45, 50).

[AGBY11] O. Alonso, M. Gertz, and R. Baeza-Yates. "Enhancing document snippets using temporal information". In: *String Processing and Information Retrieval*. Springer, 2011, pp. 26–31 (cit. on pp. 13, 33).

[AGK01] J. Allan, R. Gupta, and V. Khandelwal. "Temporal Summaries of New Topics". In: *Proceedings of the 24th ACM SIGIR Conference on Research and Development in Information Retrieval* (2001), pp. 10–18 (cit. on pp. 2, 23, 39).

[AKDZ05] L. Agnihotri, J. R. Kender, N. Dimitrova, and J. Zimmerman. "User study for generating personalized summary profiles". In: *Proceedings of the IEEE International Conference on Multimedia and Expo, 2005 (ICME 2005)*. IEEE, 2005, pp. 1094–1097 (cit. on p. 13).

[AKR09] A. Arampatzis, J. Kamps, and S. Robertson. "Where to stop reading a ranked list?: threshold optimization using truncated score distributions". In: *Proceedings of the 32nd ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2009, pp. 524–531 (cit. on p. 27).

[AMM11] E. Aramaki, S. Maskawa, and M. Morita. "Twitter catches the flu: detecting influenza epidemics using Twitter". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 1568–1576 (cit. on p. 23).

[ASBYG11] O. Alonso, J. Strötgen, R. A. Baeza-Yates, and M. Gertz. "Temporal Information Retrieval: Challenges and Opportunities." In: *Proceedings of the 1st International Temporal Web Analytics Workshop (TWAW)*. Vol. 11. 2011, pp. 1–8 (cit. on p. 11).

[AVCB11] B. G. Ahn, B. Van Durme, and C. Callison-Burch. "WikiTopics: what is popular on Wikipedia and why". In: *Proceedings of the Workshop on Automatic Summarization for Different Genres, Media, and Languages*. Association for Computational Linguistics, 2011, pp. 33–40 (cit. on p. 24).

[AWB03] J. Allan, C. Wade, and A. Bolivar. "Retrieval and novelty detection at the sentence level". In: *Proceedings of the 26th ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2003, pp. 314–321 (cit. on p. 28).

[All02] J Allan. *Topic detection and tracking: event-based information organization*. Kluwer Academic Publishers, 2002 (cit. on p. 23).

[BA02] K. P. Burnham and D. R. Anderson. *Model selection and multimodel inference: a practical information-theoretic approach*. Springer Science & Business Media, 2002 (cit. on p. 41).

[BAQ13] G. Binh Tran, M. Alrifai, and D. Quoc Nguyen. "Predicting relevant news events for timeline summaries". In: *Proceedings of the 22nd International Conference on World Wide Web*. Rio de Janeiro, Brazil: International World Wide Web Conferences Steering Committee, 2013, pp. 91–92 (cit. on pp. 32, 42).

[BBZ08] S. Berkovsky, T. Baldwin, and I. Zukerman. "Aspect-based personalized text summarization". In: *Adaptive Hypermedia and Adaptive Web-Based Systems*. Springer, 2008, pp. 267–270 (cit. on p. 13).

[BGP08] A. Bossard, M. Généreux, and T. Poibeau. "Description of the LIPN System at TAC 2008: Summarizing Information and Opinions". In: *Proceedings of the 2008 Text Analysis Conference (TAC 2008)*. National Institute of Standards and Technology, 2008, pp. 282–291 (cit. on p. 13).

[BING12] H. Becker, D. Iter, M. Naaman, and L. Gravano. "Identifying content for planned events across social media sites". In: *Proceedings of the 5th ACM international conference on Web search and data mining*. ACM, 2012, pp. 533–542 (cit. on p. 25).

[BKLB12] S. R. K. Branavan, N. Kushman, T. Lei, and R. Barzilay. "Learning high-level planning from text". In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012, pp. 126–135 (cit. on p. 76).

[BL04] R. Barzilay and L. Lee. "Catching the drift: Probabilistic content models, with applications to generation and summarization". In: *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics 2004 (HLT-NAACL 2004)*. Boston, Massachusetts, 2004 (cit. on p. 18).

[BL08] R. Barzilay and M. Lapata. "Modeling local coherence: An entity-based approach". In: *Computational Linguistics* 34.1 (2008), pp. 1–34 (cit. on p. 76).

[BME99] R. Barzilay, K. R. McKeown, and M. Elhadad. "Information fusion in the context of multi-document summarization". In: *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*. Association for Computational Linguistics, 1999, pp. 550–557 (cit. on p. 19).

[BNG10] H. Becker, M. Naaman, and L. Gravano. "Learning similarity metrics for event identification in social media". In: *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 2010, pp. 291–300 (cit. on p. 24).

[BNJ03] M. D. Blei, Y. A. Ng, and I. M. Jordan. "Latent dirichlet allocation". In: *The Journal of Machine Learning Research* 3.March 2003 (Mar. 2003), pp. 993–1022 (cit. on pp. 17, 26, 53, 54).

[CB15] D. P. Costenaro and J. Brown. *Changes to documents are automatically summarized in electronic messages*. U.S. Patent No. 8,965,983. February 24, 2015 (cit. on p. 31).

[CDJ11] R. Campos, G. Dias, and A. M. Jorge. "What is the Temporal Value of Web Snippets?" In: *Proceedings of the 1st International Temporal Web Analytics Workshop (TWAW)*. 2011, pp. 9–16 (cit. on p. 33).

[CDJJ14] R. Campos, G. Dias, A. M. Jorge, and A. Jatowt. "Survey of temporal information retrieval and related applications". In: *ACM Computing Surveys (CSUR)* 47.2 (2014), p. 15 (cit. on p. 11).

[CG98] J. Carbonell and J. Goldstein. "The use of MMR, diversity-based reranking for reordering documents and producing summaries". In: *Proceedings of the 21st ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 1998, pp. 335–336 (cit. on pp. 33, 35).

[CHT10] A. Celikyilmaz and D. Hakkani-Tür. "A hybrid hierarchical model for multi-document summarization". In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 815–824 (cit. on p. 19).

[CHT11] A. Celikyilmaz and D. Hakkani-Tür. "Discovery of topically coherent sentences for extractive summarization". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 491–499 (cit. on pp. 19, 35, 43, 93).

[CJ08] Y. Chali and S. R. Joty. "Improving the performance of the random walk model for answering complex questions". In: *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*. Association for Computational Linguistics, 2008, pp. 9–12 (cit. on p. 22).

[CL04] H. Chieu and Y. Lee. "Query based event extraction along a timeline". In: *Proceedings of the 27th ACM SIGIR conference on Research and Development in Information Retrieval* (2004), p. 425 (cit. on pp. 13, 32, 43).

[CSO06] J. M. Conroy, J. D. Schlesinger, and D. P. O'Leary. "Topic-Focused Multi-Document Summarization Using an Approximate Oracle Score". In: *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*. Sydney: Association for Computational Linguistics, 2006, pp. 152–159 (cit. on pp. 16, 34).

[CSO09] J. M. Conroy, J. D. Schlesinger, and D. P. O'leary. "Classy 2009: summarization and metrics". In: *Proceedings of the text analysis conference (TAC)*. Citeseer, 2009 (cit. on p. 42).

[CSO11] J. M. Conroy, J. D. Schlesinger, and D. P. O'Leary. "Nouveau-ROUGE: A novelty metric for update summarization". In: *Computational Linguistics* 37.1 (2011), pp. 1–8 (cit. on p. 42).

[CSS07] C. Chemudugunta, P. Smyth, and M. Steyvers. "Modeling general and specific aspects of documents with a probabilistic topic model". In: *NIPS*. 2007, p. 241 (cit. on pp. 17, 18, 26).

[Cam13] R. Campos. "Disambiguating Implicit Temporal Queries for Temporal Information Retrieval Applications". PhD Thesis. University of Porto, 2013. URL: http://www.ccc.ipt.pt/~ricardo/publications.html (cit. on p. 3).

[Cg10] M. Ciglan and K. Nø rvå g. "WikiPop: personalized event detection system based on Wikipedia page view statistics". In: *Proceedings of the 19th ACM international Conference on Information and Knowledge Management (CIKM)*. ACM. Toronto, ON, Canada: ACM, 2010, pp. 1931–1932 (cit. on p. 24).

[Cha] *Changedetect: A web page monitoring service*. URL: http://www.changedetect.com/ (visited on 12/19/2015) (cit. on pp. 2, 30).

[DA12] J.-Y. Delort and E. Alfonseca. "DualSum: a Topic-Model based approach for update summarization". In: *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2012, pp. 214–223 (cit. on pp. 13, 26, 41, 43, 62).

[DDLFH90] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. "Indexing by latent semantic analysis". In: *Journal of the American Society for Information Science* 41.6 (1990), pp. 391–407 (cit. on p. 17).

[DGZC10] P. Du, J. Guo, J. Zhang, and X. Cheng. "Manifold ranking with sink points for update summarization". In: *Proceedings of the 19th ACM international Conference on Information and Knowledge Management*. ACM, 2010, pp. 1757–1760 (cit. on pp. 28, 33).

[DIIM04] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. "Locality-sensitive hashing scheme based on p-stable distributions". In: *Proceedings of the 20th annual symposium on Computational Geometry*. ACM, 2004, pp. 253–262 (cit. on p. 24).

[DM06] H. Daumé III and D. Marcu. "Bayesian query-focused summarization". In: *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2006, pp. 305–312 (cit. on p. 18).

[Dun93] T. Dunning. "Accurate Methods for the Statistics of Surprise and Coincidence". In: *Computational Linguistics* 19 (1993), pp. 61–74 (cit. on p. 17).

[ER04a] G. Erkan and D. R. Radev. "LexPageRank: Prestige in Multi-Document Text Summarization". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing 2004*. Barcelona, Spain: Association for Computational Linguistics, 2004 (cit. on p. 20).

[ER04b] G. Erkan and D. R. Radev. "LexRank: graph-based lexical centrality as salience in text summarization". In: *Journal of Artificial Intelligence Research* 22.1 (2004), pp. 457–479 (cit. on p. 20).

[Efr10] M. Efron. "Linear time series models for term weighting in information retrieval". In: *Journal of the American Society for Information Science and Technology* 61.7 (2010), pp. 1299–1312 (cit. on p. 52).

[FH04] E. Filatova and V. Hatzivassiloglou. "Event-based extractive summarization". In: *Proceedings of the Association for Computational Linguistics Workshop on Summarization*. 2004, pp. 104–111 (cit. on p. 16).

[FSTM13] F. Fukumoto, Y. Suzuki, A. Takasu, and S. Matsuyoshi. "Multi-document summarization based on event and topic detection". In: *Proceedings of the 6th Language and Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*. 2013, pp. 117–121 (cit. on p. 30).

[FZG11] O. Ferschke, T. Zesch, and I. Gurevych. "Wikipedia revision toolkit: efficiently accessing Wikipedia's edit history". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Systems Demonstrations*. Association for Computational Linguistics, 2011, pp. 97–102 (cit. on p. 58).

[Fis87] D. H. Fisher. "Knowledge acquisition via incremental conceptual clustering". In: *Machine learning* 2.2 (1987), pp. 139–172 (cit. on p. 27).

[Fri40] M. Friedman. "A comparison of alternative tests of significance for the problem of m rankings". In: *The Annals of Mathematical Statistics* 11.1 (1940), pp. 86–92 (cit. on p. 69).

[GG04] C. Gupta and R. L. Grossman. "GenIc: A Single-Pass Generalized Incremental Algorithm for Clustering." In: *Proceedings of the 2004 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, 2004, pp. 147–153 (cit. on p. 27).

[GKKNS13] M. Georgescu, N. Kanhabua, D. Krause, W. Nejdl, and S. Siersdorfer. "Extracting event-related information from article updates in wikipedia". In: *Advances in Information Retrieval*. Springer, 2013, pp. 254–266 (cit. on pp. 2, 26, 59).

[GL01] Y. Gong and X. Liu. "Generic text summarization using relevance measure and latent semantic analysis". In: *Proceedings of the 24th ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2001, pp. 19–25 (cit. on p. 17).

[GLF89] J. H. Gennari, P. Langley, and D. Fisher. "Models of incremental concept formation". In: *Artificial intelligence* 40.1 (1989), pp. 11–61 (cit. on p. 27).

[GLNWL07] F. Gotti, G. Lapalme, L. Nerima, E. Wehrli, and T. du Langage. "GOFAIsum: a symbolic summarizer for DUC". In: *Proceedings of the Document Understanding Conference 2007*. Vol. 7. Rochester: National Institute of Standards and Technology, 2007 (cit. on p. 16).

[GPKZSN13] M. Georgescu, D. D. Pham, N. Kanhabua, S. Zerr, S. Siersdorfer, and W. Nejdl. "Temporal summarization of event-related updates in wikipedia". In: *Proceedings of the 22nd International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2013, pp. 281–284 (cit. on pp. 26, 29, 80).

[GS04] T. L. Griffiths and M. Steyvers. "Finding scientific topics". In: *Proceedings of the National Academy of Sciences* 101.suppl 1 (2004), pp. 5228–5235 (cit. on pp. 79, 100).

[GT04] D. Griffiths and M. Tenenbaum. "Hierarchical topic models and the nested Chinese restaurant process". In: *Advances in neural information processing systems* 16 (2004), p. 17 (cit. on pp. 18, 19).

[Gooa] *Diff, Match and Patch libraries for Plain Text*. URL: http://code.google.com/p/google-diff-match-patch/ (visited on 09/22/2014) (cit. on p. 47).

[Goob] *Google News Timeline*. URL: https://news.google.com/ (visited on 12/19/2015) (cit. on p. 32).

[Gri02] T Griffiths. *Gibbs sampling in the generative model of Latent Dirichlet Allocation*. Tech. rep. 2002 (cit. on pp. 54, 65).

[Gru69] F. E. Grubbs. "Procedures for detecting outlying observations in samples". In: *Technometrics* 11.1 (1969), pp. 1–21 (cit. on p. 25).

[HCGL08] T. He, J. Chen, Z. Gui, and F. Li. "CCNU at TAC 2008: Proceeding on Using Semantic Method for Automated Summarization Yield". In: *Proceedings of the Text Analysis Conference (TAC) 2008*. National Institute of Standards and Technology, 2008 (cit. on p. 13).

[HDMS07] S. P. Hobson, B. J. Dorr, C. Monz, and R. Schwartz. "Task-based evaluation of text summarization using relevance prediction". In: *Information Processing & Management* 43.6 (2007), pp. 1482–1499 (cit. on p. 37).

[HKHBKM01] V. Hatzivassiloglou, J. L. Klavans, M. L. Holcombe, R. Barzilay, M.-Y. Kan, and K. McKeown. "SIMFINDER: A Flexible Clustering Tool for Summarization". In: *Proceedings of the NAACL Workshop on Automatic Summarization*. 2001 (cit. on p. 20).

[HL05] S. Harabagiu and F. Lacatusu. "Topic themes for multi-document summarization". In: *Proceedings of the 28th ACM SIGIR conference on Research and Development in Information Retrieval*. Santiago de Chile: ACM, 2005, pp. 202–209 (cit. on pp. 16, 17, 21, 34).

[HLZF06] E. Hovy, C.-Y. Lin, L. Zhou, and J. Fukumoto. "Automated summarization evaluation with basic elements". In: *Proceedings of the Fifth Conference on Language Resources and Evaluation (LREC 2006)*. 2006, pp. 604–611 (cit. on p. 37).

[HMR06] B. Hachey, G. Murray, and D. Reitter. "Dimensionality reduction aids term co-occurrence based multi-document summarization". In: *Proceedings of the workshop on task-focused summarization and question answering*. Association for Computational Linguistics, 2006, pp. 1–7 (cit. on p. 17).

[HOYI07] T. Hirao, M. Okumura, N. Yasuda, and H. Isozaki. "Supervised automatic evaluation for summarization with voted regression model". In: *Information Processing & Management* 43.6 (2007), pp. 1521–1535 (cit. on p. 41).

[HP10] D. He and D. S. Parker. "Topic dynamics: an alternative model of bursts in streams of topics". In: *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 2010, pp. 443–452 (cit. on pp. 29, 30, 80).

[HSSTZW02] H. Hardy, N. Shimizu, T. Strzalkowski, L. Ting, X. Zhang, and G. B. Wise. "Cross-document summarization by concept classification". In: *Proceedings of the 25th ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 2002, pp. 121–128 (cit. on p. 21).

[HV09] A. Haghighi and L. Vanderwende. "Exploring content models for multi-document summarization". In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL '09)*. Association for Computational Linguistics, 2009, pp. 362–370 (cit. on pp. 17, 18, 26, 36, 43, 76).

[Har02] D. Harman. "Overview of the TREC 2002 Novelty Track." In: *Proceedings of the Text REtrieval Conference (TREC) 2002*. 2002 (cit. on p. 28).

[Har04] S. Harabagiu. "Incremental topic representations". In: *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, 2004, p. 583 (cit. on p. 17).

[How06] J. Howe. "The rise of crowdsourcing". In: *Wired magazine* 14.6 (2006), pp. 1–4 (cit. on p. 43).

[JBI04] A. Jatowt, K. K. Bun, and M. Ishizuka. "Change Summarization in Web Collections". In: *Innovations in Applied Artificial Intelligence*. Springer, 2004, pp. 653–662 (cit. on pp. 2, 13, 30, 31, 36, 48, 50, 64, 74–76, 99).

[JG96] K. S. Jones and J. R. Galliers. *Evaluating natural language processing systems: An analysis and review*. Vol. 1083. Springer Science & Business Media, 1996 (cit. on p. 37).

[JM00] H. Jing and K. R. McKeown. "Cut and Paste Based Text Summarization". In: *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*. Association for Computational Linguistics, 2000, pp. 178–185 (cit. on pp. 11, 46).

[JSMH10] X. Jin, S. Spangler, R. Ma, and J. Han. "Topic initiator detection on the world wide web". In: *Proceedings of the 19th international conference on World Wide Web*. ACM, 2010, pp. 481–490 (cit. on p. 32).

[JZX03] Q. Jin, J. Zhao, and B. Xu. "NLPR at TREC 2003: Novelty and Robust." In: *Proceedings of the Text REtrieval Conference 2003*. National Institute of Standards and Technology, 2003, pp. 126–137 (cit. on p. 28).

[Jav] *The DiffUtils library for computing diffs, applying patches, generationg side-by-side view*. URL: http://code.google.com/p/java-diff-utils/ (visited on 09/22/2014) (cit. on p. 47).

[Joa00] T. Joachims. "Estimating the generalization performance of a SVM efficiently". In: *Proceedings of the 17th International Conference on Machine Learning*. Universität Dortmund, 2000 (cit. on p. 29).

[Jon07] K. S. Jones. "Automatic summarising: The state of the art". In: *Information Processing & Management* 43.6 (2007), pp. 1449–1481 (cit. on p. 12).

[KA04] G. Kumaran and J. Allan. "Text classification and named entities for new event detection". In: *Proceedings of the 27th ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 2004, pp. 297–304 (cit. on p. 28).

[KBg15] N. Kanhabua, R. Blanco, and K. Nø rvå g. "Temporal Information Retrieval". In: *Foundations and Trends in Information Retrieval* 9.2 (2015), pp. 92–+ (cit. on p. 11).

[KGC12] B. Keegan, D. Gergle, and N. Contractor. "Staying in the loop: structure and dynamics of Wikipedia's breaking news collaborations". In: *Proceedings of the Eighth Annual International Symposium on Wikis and Open Collaboration*. ACM, 2012, p. 1 (cit. on p. 25).

[KJ00] R. Klinkenberg and T. Joachims. "Detecting Concept Drift with Support Vector Machines." In: *Proceedings of the Seventeenth International Conference on Machine Learning (ICML '00)*. Morgan Kaufmann Publishers Inc., 2000, pp. 487–494 (cit. on p. 29).

[KLPM10] H. Kwak, C. Lee, H. Park, and S. Moon. "What is Twitter, a social network or a news media?" In: *Proceedings of the 19th international conference on World Wide Web*. ACM, 2010, pp. 591–600 (cit. on p. 24).

[KM02] K. Knight and D. Marcu. "Summarization beyond sentence extraction: A probabilistic approach to sentence compression". In: *Artificial Intelligence* 139.1 (2002), pp. 91–107 (cit. on pp. 11, 46).

[KNR15] M. Kar, S. Nunes, and C. Ribeiro. "Summarization of changes in dynamic text collections using Latent Dirichlet Allocation model". In: *Information Processing & Management* 51.6 (2015), pp. 809–833 (cit. on pp. 45, 75, 77, 78, 92).

[KTHMB12] R. Kessler, X. Tannier, C. Hagege, V. Moriceau, and A. Bittar. "Finding salient dates for building thematic timelines". In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012, pp. 730–739 (cit. on p. 32).

[Kar13] M. Kar. "Summarization of Changes in Dynamic Text Collections". In: *Proceedings of the 5th BCS IRSG Symposium on Future Directions in Information Access (FDIA 2013)*. 2013, pp. 14–19 (cit. on p. 11).

[Kle03] J. Kleinberg. "Bursty and hierarchical structure in streams". In: *Data Mining and Knowledge Discovery* 7.4 (2003), pp. 373–397 (cit. on pp. 29, 80).

[LBK09] J. Leskovec, L. Backstrom, and J. Kleinberg. "Meme-tracking and the Dynamics of the News Cycle". In: *Proceedings of the 15th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. ACM, 2009, pp. 497–506 (cit. on pp. 29, 80).

[LC08] X. Li and W. B. Croft. "An information-pattern-based approach to novelty detection". In: *Information Processing & Management* 44.3 (2008), pp. 1159–1188 (cit. on pp. 28, 29).

[LDS13] X. Li, L. Du, and Y.-D. Shen. "Update summarization via graph-based sentence ranking". In: *IEEE Transactions on Knowledge and Data Engineering* 25.5 (2013), pp. 1162–1174 (cit. on pp. 28, 41, 62, 99).

[LH00] C.-Y. Lin and E. Hovy. "The automated acquisition of topic signatures for text summarization". In: *Proceedings of the 18th conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 2000, pp. 495–501 (cit. on pp. 16, 17, 38).

[LH03] C.-Y. Lin and E. Hovy. "Automatic evaluation of summaries using n-gram co-occurrence statistics". In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL '03) - Volume 1*. Association for Computational Linguistics, 2003, pp. 71–78 (cit. on pp. 41, 62, 87, 101).

[LL10] F. Liu and Y. Liu. "Exploring correlation between ROUGE and human evaluation on meeting summaries". In: *IEEE Transactions on Audio, Speech, and Language Processing* 18.1 (2010), pp. 187–196 (cit. on p. 43).

[LM06] W. Li and A. McCallum. "Pachinko allocation: DAG-structured mixture models of topic correlations". In: *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 577–584 (cit. on p. 35).

[LM09] K. Lerman and R. McDonald. "Contrastive summarization: an experiment with consumer reviews". In: *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics, companion volume: Short papers*. Association for Computational Linguistics, 2009, pp. 113–116 (cit. on p. 36).

[LMFG05] J. Leskovec, N. Milic-Frayling, and M. Grobelnik. "Impact of linguistic analysis on the semantic graph coverage and learning of document extracts". In: *Proceedings of the 20th national conference on Artificial intelligence*. 2005, pp. 1069–1074 (cit. on p. 22).

[LP12] E. Lloret and M. Palomar. "Text summarisation in progress: a literature review". In: *Artificial Intelligence Review* 37.1 (2012), pp. 1–41 (cit. on pp. 12, 13, 34).

[LPAK09] J.-H. Lee, S. Park, C.-M. Ahn, and D. Kim. "Automatic generic document summarization based on non-negative matrix factorization". In: *Information Processing & Management* 45.1 (2009), pp. 20–34 (cit. on p. 17).

[LPT00] L. Liu, C. Pu, and W. Tang. "WebCQ-detecting and delivering information changes on the web". In: *Proceedings of the ninth international conference on Information and knowledge management*. ACM, 2000, pp. 512–519 (cit. on pp. 13, 30).

[LWLM05] Z. Li, B. Wang, M. Li, and W.-Y. Ma. "A probabilistic model for retrospective news event detection". In: *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2005, pp. 106–113 (cit. on p. 23).

[Lin04] C.-Y. Lin. "ROUGE: A package for automatic evaluation of summaries". In: *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*. 2004, pp. 74–81 (cit. on p. 37).

[Lit03] K. C. Litkowski. "Use of Metadata for Question Answering and Novelty Tasks". In: *Proceedings of the Text REtrieval Conference 2003*. 2003, pp. 161–176 (cit. on p. 28).

[Luh58] H. P. Luhn. "The Automatic Creation of Literature Abstracts". In: *IBM Journal of Research and Development* 2.2 (1958), pp. 159–165 (cit. on pp. 15, 34).

[MC11] R. Mason and E. Charniak. "Extractive multi-document summaries should explicitly not contain document-specific content". In: *Proceedings of the Workshop on Automatic Summarization for Different Genres, Media, and Languages*. Association for Computational Linguistics, 2011, pp. 49–54 (cit. on p. 36).

[MCFMZ94] T. M. Mitchell, R. Caruana, D. Freitag, J. McDermott, and D. Zabowski. "Experience with a learning personal assistant". In: *Communications of the ACM* 37.7 (1994), pp. 80–91 (cit. on p. 29).

[MHKHFS99] I. Mani, D. House, G. Klein, L. Hirschman, T. Firmin, and B. Sundheim. "The TIPSTER SUMMAC text summarization evaluation". In: *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 1999, pp. 77–85 (cit. on p. 37).

[MLM07] D. Mimno, W. Li, and A. McCallum. "Mixtures of Hierarchical Topics with Pachinko Allocation". In: *Proceedings of the 24th International Conference on Machine Learning*. ACM, 2007, pp. 633–640 (cit. on p. 17).

[MMO14] R. McCreadie, C. Macdonald, and I. Ounis. "Incremental update summarization: Adaptive sentence selection based on prevalence and novelty". In: *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management (CIKM)*. ACM, 2014, pp. 301–310 (cit. on p. 27).

[MR95] K. McKeown and D. R. Radev. "Generating summaries of multiple news articles". In: *Proceedings of the 18th ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1995, pp. 74–82 (cit. on p. 19).

[MT04] R. Mihalcea and P. Tarau. "TextRank: Bringing order into texts". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2004, pp. 404–411 (cit. on p. 20).

[MT05] R. Mihalcea and P. Tarau. "A language independent algorithm for single and multiple document summarization". In: *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP)*. Jeju Island, Korea: Springer Science & Business Media, 2005 (cit. on p. 20).

[MW00] I. Mani and G. Wilson. "Robust temporal processing of news". In: *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2000, pp. 69–76 (cit. on p. 12).

[MZ05] Q. Mei and C. Zhai. "Discovering evolutionary theme patterns from text: an exploration of temporal text mining". In: *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*. ACM, 2005, pp. 198–207 (cit. on p. 12).

[McC04] V. McCargar. "Statistical approaches to automatic text summarization". In: *Bulletin of the american society for information science and technology* 30.4 (2004), pp. 21–25 (cit. on p. 16).

[Med] *Mediawiki API: A web service that provides convenient access to wiki features, data, and meta-data over HTTP*. URL: http://www.mediawiki.org/wiki/API (visited on 12/20/2015) (cit. on p. 59).

[Met] *Meta-wiki: List of wikipedias*. URL: https://goo.gl/DDFG92 (visited on 09/01/2014) (cit. on p. 58).

[Mur99] J. J. Murphy. *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. Penguin, 1999 (cit. on pp. 30, 80).

[NM12] A. Nenkova and K. McKeown. "A survey of text summarization techniques". In: *Mining Text Data*. Springer, 2012, pp. 43–76 (cit. on pp. 12, 34, 99).

[NP04] A. Nenkova and R. Passonneau. "Evaluating content selection in summarization: The pyramid method". In: *Proceedings of Human Language Technologies: The 2004 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL '04)*. Association for Computational Linguistics, 2004 (cit. on pp. 37, 42).

[NRD08] S. Nunes, C. Ribeiro, and G. David. "WikiChanges: exposing Wikipedia revision activity". In: *Proceedings of the 4th International Symposium on Wikis*. ACM, 2008, p. 25 (cit. on pp. 2, 30, 31, 46, 63, 64, 75, 76, 99).

[NRD11] S. Nunes, C. Ribeiro, and G. David. "Term weighting based on document revision history". In: *Journal of the American Society for Information Science and Technology* 62.12 (2011), pp. 2471–2478 (cit. on pp. 43, 52).

[NV05] A. Nenkova and L. Vanderwende. "The impact of frequency on summarization". In: *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005-101* (2005) (cit. on pp. 15, 34).

[NVM06] A. Nenkova, L. Vanderwende, and K. McKeown. "A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization". In: *Proceedings of the 29th ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 2006, pp. 573–580 (cit. on p. 34).

[Nem63] P. B. Nemenyi. "Distribution-free multiple comparisons". PhD thesis. Princeton, 1963 (cit. on p. 69).

[Ntc] *NTCIR Temporal Information Access (Temporalia)*. URL: http://ntcirtemporalia.github.io/index.html (visited on 12/20/2015) (cit. on p. 3).

[OCA10] M. G. Ozsoy, I. Cicekli, and F. N. Alpaslan. "Text summarization of turkish texts using latent semantic analysis". In: *Proceedings of the 23rd international conference on computational linguistics*. Association for Computational Linguistics, 2010, pp. 869–876 (cit. on p. 17).

[OER05] J. Otterbacher, G. Erkan, and D. R. Radev. "Using random walks for question-focused sentence retrieval". In: *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2005, pp. 915–922 (cit. on p. 22).

[OLZLL13] Y. Ouyang, W. Li, R. Zhang, S. Li, and Q. Lu. "A progressive sentence selection strategy for document summarization". In: *Information Processing & Management* 49.1 (2013), pp. 213–221 (cit. on p. 36).

[OPMMO12] M. Osborne, S. Petrovic, R. McCreadie, C. Macdonald, and I. Ounis. "Bieber no more: First story detection using Twitter and Wikipedia". In: *SIGIR 2012 Workshop on Time-aware Information Access*. 2012 (cit. on p. 25).

[PBMW99] L. Page, S. Brin, R. Motwani, and T. Winograd. *The PageRank citation ranking: Bringing order to the web.* Tech. rep. 1999 (cit. on pp. 20, 21).

[PC98] J. Ponte and W. Croft. "A language modeling approach to information retrieval". In: *Proceedings of the 21st ACM SIGIR conference on Research and Development in Information Retrieval*. 1998, pp. 275–281 (cit. on p. 23).

[POL10] S. Petrović, M. Osborne, and V. Lavrenko. "Streaming first story detection with application to twitter". In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 181–189 (cit. on p. 24).

[PRWZ02] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. "BLEU: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318 (cit. on p. 41).

[RBGZ01] D. R. Radev, S. Blair-Goldensohn, and Z. Zhang. "Experiments in single and multi-document summarization using MEAD". In: *Proceedings of the Document Understanding Conference 2001*. New Orleans, Louisiana USA: National Institute of Standards and Technology, 2001 (cit. on p. 34).

[RHM99] D. R. Radev, V. Hatzivassiloglou, and K. R. McKeown. "A description of the CIDR system as used for TDT-2". In: *Broadcast News Workshop'99 Proceedings*. Morgan Kaufmann Pub, 1999, p. 205 (cit. on p. 19).

[RJST04] D. R. Radev, H. Jing, M. Styś, and D. Tam. "Centroid-based summarization of multiple documents". In: *Information Processing & Management* 40.6 (2004), pp. 919–938 (cit. on pp. 19, 37, 38, 40).

[Rev] *Wikipedia revision toolkit*. URL: http://code.google.com/p/jwpl/wiki/WikipediaRevisionToolkit (visited on 09/22/2014) (cit. on p. 58).

[SB10] I. Subašić and B. Berendt. "Discovery of interactive graphs for understanding and searching time-indexed corpora". In: *Knowledge and Information Systems* 23.3 (2010), pp. 293–319 (cit. on p. 23).

[SCKDP06] N. Sahoo, J. Callan, R. Krishnan, G. Duncan, and R. Padman. "Incremental hierarchical clustering of text documents". In: *Proceedings of the 15th ACM international conference on Information and knowledge management*. ACM, 2006, pp. 357–366 (cit. on p. 27).

[SCL08] S. Sweeney, F. Crestani, and D. E. Losada. "'Show me more': Incremental length summarisation using novelty detection". In: *Information Processing & Management* 44.2 (2008), pp. 663–686 (cit. on p. 29).

[SF14] Y. Suzuki and F. Fukumoto. "Detection of Topic and its Extrinsic Evaluation Through Multi-Document Summarization". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, 2014 (cit. on p. 30).

[SH03] I. Soboroff and D. Harman. "Overview of the TREC 2003 Novelty Track." In: *Proceedings of the Text REtrieval Conference (TREC) 2003*. 2003, pp. 38–53 (cit. on p. 28).

[SJ05] J. Steinberger and K. Ježek. "Text summarization and singular value decomposition". In: *Advances in Information Systems*. Springer, 2005, pp. 245–254 (cit. on p. 40).

[SK07] M. Scholz and R. Klinkenberg. "Boosting classifiers for drifting concepts". In: *Intelligent Data Analysis* 11.1 (2007), pp. 3–28 (cit. on p. 30).

[SK08] F. Schilder and R. Kondadadi. "FastSum: fast and accurate query-based multi-document summarization". In: *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*. Association for Computational Linguistics, 2008, pp. 205–208 (cit. on pp. 28, 41, 62).

[SKLC08] F. Schilder, R. Kondadadi, J. L. Leidner, and J. G. Conrad. "Thomson reuters at TAC 2008: Aggressive filtering with fastsum for update and opinion summarization". In: *Proceedings of the 1st Text Analysis Conference, TAC-2008*. National Institute of Standards and Technology, 2008 (cit. on pp. 28, 41, 62).

[SOM10] T. Sakaki, M. Okazaki, and Y. Matsuo. "Earthquake shakes Twitter users: real-time event detection by social sensors". In: *Proceedings of the 19th international conference on World Wide Web*. ACM, 2010, pp. 851–860 (cit. on p. 23).

[SPKJ07] J. Steinberger, M. Poesio, M. A. Kabadjov, and K. Ježek. "Two uses of anaphora resolution in summarization". In: *Information Processing & Management* 43.6 (2007), pp. 1663–1680 (cit. on pp. 17, 40).

[SSTLS09] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, and J. Sperling. "Twitterstand: news in tweets". In: *Proceedings of the 17th acm sigspatial international conference on advances in geographic information systems*. ACM, 2009, pp. 42–51 (cit. on p. 24).

[STDK12] K. M. Svore, J. Teevan, S. T. Dumais, and A. Kulkarni. "Creating temporally dynamic web search snippets". In: *Proceedings of the 35th ACM SIGIR Conference on Research and Development in Information Retrieval* (2012), p. 1045 (cit. on p. 33).

[SVS13] T. Steiner, S. Van Hooland, and E. Summers. "Mj no more: using concurrent wikipedia edit spikes with social network plausibility checks for breaking news detection". In: *Proceedings of the 22nd international conference on World Wide Web companion*. International World Wide Web Conferences Steering Committee, 2013, pp. 791–794 (cit. on p. 25).

[Sob04] I. Soboroff. "Overview of the TREC 2004 Novelty Track". In: *Proceedings of the Text REtrieval Conference (TREC) 2004*. National Institute of Standards and Technology, 2004 (cit. on p. 28).

[Ste14] T. Steiner. "Telling Breaking News Stories from Wikipedia with Social Multimedia: A Case Study of the 2014 Winter Olympics". In: *Proceedings of the 1st International Workshop on Social Multimedia and Storytelling (SoMuS), co-located with the 4th International Conference on Multimedia Retrieval (ICMR '14)*. Glasgow, Scotland, UK: ACM, 2014 (cit. on p. 25).

[TAH15] G. Tran, M. Alrifai, and E. Herder. "Timeline Summarization from Relevant Headlines". In: *Advances in Information Retrieval*. Springer, 2015, pp. 245–256 (cit. on pp. 32, 43).

[TRW11] M. Tsagkias, M. de Rijke, and W. Weerkamp. "Linking online news and social media". In: *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 2011, pp. 565–574 (cit. on p. 24).

[TTHW07] A. Turpin, Y. Tsegay, D. Hawking, and H. E. Williams. "Fast generation of result snippets in web search". In: *Proceedings of the 30th ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 2007, pp. 127–134 (cit. on p. 35).

[TV04] S. Teufel and H. Van Halteren. "Evaluating Information Content by Factoid Analysis: Human annotation and stability." In: *Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP '04)*. Barcelona, Spain, 2004, pp. 419–426 (cit. on p. 37).

[TYC09] J. Tang, L. Yao, and D. Chen. "Multi-topic based Query-oriented Summarization". In: *SIAM International Conference on Data Mining (SDM09)*. Vol. 9. Society for Industrial and Applied Mathematics, 2009, pp. 1147–1158 (cit. on p. 17).

[Tai] *Workshop on Time Aware Information Access (TAIA)*. URL: http://sigir.org/sigir2013/workshops.html (visited on 12/20/2015) (cit. on p. 3).

[Tem] *Temporal Web Analytics Workshop*. URL: http://www.temporalweb.net/ (visited on 12/20/2015) (cit. on p. 3).

[Trea] *TREC Temporal Summarization*. URL: http://www.trec-ts.org/ (visited on 12/20/2015) (cit. on p. 3).

[Treb] *TREC: Text REtrieval Conference*. URL: http://trec.nist.gov/ (visited on 12/20/2015) (cit. on p. 3).

[Usu] *DUC 2007: Document Understanding Conference*. URL: http://www-nlpir.nist.gov/projects/duc/duc2007/tasks.html (visited on 12/20/2015) (cit. on pp. 3, 26).

[VH06] R. Varadarajan and V. Hristidis. "A system for query-specific document summarization". In: *Proceedings of the 15th ACM international conference on Information and knowledge management*. ACM, 2006, pp. 622–631 (cit. on p. 35).

[VSBN07] L. Vanderwende, H. Suzuki, C. Brockett, and A. Nenkova. "Beyond SumBasic: Task-focused summarization with sentence simplification and lexical expansion". In: *Information Processing & Management* 43.6 (2007), pp. 1606–1618 (cit. on p. 16).

[VV98] V. N. Vapnik and V. Vapnik. *Statistical learning theory*. Vol. 1. Wiley New York, 1998 (cit. on p. 29).

[WFQY08] L. Wenjie, W. Furu, L. Qin, and H. Yanxiang. "PNR 2: ranking sentences with positive and negative reinforcement for query-oriented update summarization". In: *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2008, pp. 489–496 (cit. on pp. 3, 13, 28, 62).

[WH07] D. M. Wilkinson and B. A. Huberman. "Cooperation and quality in wikipedia". In: *Proceedings of the 2007 international symposium on Wikis*. ACM, 2007, pp. 157–164 (cit. on p. 25).

[WK96] G. Widmer and M. Kubat. "Learning in the presence of concept drift and hidden contexts". In: *Machine learning* 23.1 (1996), pp. 69–101 (cit. on p. 29).

[WL10] D. Wang and T. Li. "Document update summarization using incremental hierarchical clustering". In: *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, 2010, pp. 279–288 (cit. on pp. 27, 41, 62).

[WL12] D. Wang and T. Li. "Weighted consensus multi-document summarization". In: *Information Processing & Management* 48.3 (2012), pp. 513–523 (cit. on p. 2).

[WLL08] C. Wang, L. Long, and L. Li. "HowNet based evaluation for Chinese text summarization". In: *International Conference on Natural Language Processing and Knowledge Engineering 2008 (NLP-KE'08)*. Beijing, China: IEEE, 2008, pp. 1–6 (cit. on p. 42).

[WLZD08] D. Wang, T. Li, S. Zhu, and C. Ding. "Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization". In: *Proceedings of the 31st ACM SIGIR conference on Research and Development in Information Retrieval*. Singapore, Singapore: ACM, 2008, pp. 307–314 (cit. on p. 20).

[WY06] X. Wan and J. Yang. "Improved affinity graph based multi-document summarization". In: *Proceedings of the human language technology conference of the NAACL, Companion volume: Short papers*. Association for Computational Linguistics, 2006, pp. 181–184 (cit. on pp. 20, 22).

[WY08] X. Wan and J. Yang. "Multi-document summarization using cluster-based link analysis". In: *Proceedings of the 31st ACM SIGIR conference on Research and Development in Information Retrieval*. Singapore, Singapore: ACM, 2008, pp. 299–306 (cit. on pp. 20, 21).

[Wik] *Wikimedia database dumps*. URL: http://dumps.wikimedia.org/ (visited on 06/01/2014) (cit. on p. 58).

[XL08] S. Xie and Y. Liu. "Using corpus and knowledge-based similarity measure in maximum marginal relevance for meeting summarization". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2008, pp. 4985–4988 (cit. on p. 35).

[YGVS07] W.-t. Yih, J. Goodman, L. Vanderwende, and H. Suzuki. "Multi-Document Summarization by Maximizing Informative Content-Words." In: *Proceedings of the 20th international joint conference on Artifical intelligence (IJCAI'07)*. Vol. 2007. Hyderabad, India: Morgan Kaufmann Publishers Inc., 2007, 20th (cit. on pp. 15, 34).

[YKYM05] J.-Y. Yeh, H.-R. Ke, W.-P. Yang, and I.-H. Meng. "Text summarization using a trainable summarizer and latent semantic analysis". In: *Information Processing & Management* 41.1 (2005), pp. 75–95 (cit. on p. 17).

[YWOKLZ11] R. Yan, X. Wan, J. Otterbacher, L. Kong, X. Li, and Y. Zhang. "Evolutionary timeline summarization: a balanced optimization framework via iterative substitution". In: *Proceedings of the 34th ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 2011, pp. 745–754 (cit. on pp. 13, 32, 37).

[YZCJ02] Y. Yang, J. Zhang, J. Carbonell, and C. Jin. "Topic-conditioned novelty detection". In: *Proceedings of the 8th ACM international Conference on Knowledge Discovery and Data Mining (SIGKDD '02)*. Edmonton, AB, Canada: ACM, 2002, pp. 688–693 (cit. on p. 28).

[ZCM02] Y. Zhang, J. Callan, and T. Minka. "Novelty and redundancy detection in adaptive filtering". In: *Proceedings of the 25th ACM SIGIR conference on Research and Development in Information Retrieval*. Tampere, Finland: ACM, 2002, pp. 81–88 (cit. on p. 29).

[ZDSML05] D. Zajic, B. Dorr, R. Schwartz, C. Monz, and J Lin. "A sentence-trimming approach to multi-document summarization". In: *Proceedings of HLT/EMNLP 2005 Workshop on Text Summarization (HLT/EMNLP'05)*. 2005, pp. 151–158 (cit. on p. 34).

[ZDXC09] J. Zhang, P. Du, H. B. Xu, and X. Q. Cheng. "ICTGrasper at TAC2009: Temporal Preferred Update Summarization". In: *Proceedings of the Second Text Analysis Conference (TAC 2009)*. Gaithersburg, Maryland, USA: National Institute of Standards and Technology, 2009 (cit. on pp. 28, 41, 62).

[ZGH12] Z. Zhang, S. S. Ge, and H. He. "Mutual-reinforcement document summarization using embedded graph based sentence clustering for storytelling". In: *Information Processing & Management* 48.4 (2012), pp. 767–778 (cit. on p. 22).

[ZGYHL13] X. W. Zhao, Y. Guo, R. Yan, Y. He, and X. Li. "Timeline generation with social attention". In: *Proceedings of the 36th ACM SIGIR conference on Research and Development in Information Retrieval*. Dublin, Ireland: ACM, 2013, pp. 1061–1064 (cit. on p. 32).

[ZS03] Y. Zhu and D. Shasha. "Efficient elastic burst detection in data streams". In: *Proceedings of the 9th ACM international conference on Knowledge Discovery and Data mining (SIGKDD)*. Washinghton D.C.: ACM, 2003, pp. 336–345 (cit. on pp. 30, 80).

[ZWH09] L. Zhao, L. Wu, and X. Huang. "Using query expansion in graph-based approach for query-focused multi-document summarization". In: *Information Processing & Management* 45.1 (2009), pp. 35–41 (cit. on p. 22).

[Zha02] H. Zha. "Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering". In: *Proceedings of the 25th ACM SIGIR conference on Research and Development in Information Retrieval*. Tampere, Finland: ACM, 2002, pp. 113–120 (cit. on p. 20).