THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *17/09/2018* **par :**

Abdelhamid CHELLAL

Event Summarization on Social Media Stream: Retrospective and Prospective Tweet Summarization

JURY

Lynda TAMINE-LECHANI	Professeur, Université Toulouse 3	Présidente du Jury
Sihem AMER-YAHIA	Directrice de Recherche CNRS	Rapportrice
Patrice BELLOT	Professeur, Université Aix-Marseille	Rapporteur
Philippe MULHEM	Chargé de Recherche CNRS,	Examinateur
Bernard DOUSSET	Professeur émérite, Université Toulouse 3	Directeur
Mohand BOUGHANEM	Professeur, Université Toulouse 3	Co-directeur

École doctorale et spécialité :
MITT : Image, Information, Hypermedia
Unité de Recherche :
Institut de Recherche en Informatique de Toulouse (UMR 5505)
Directeur(s) de Thèse :
Bernard DOUSSET et *Mohand BOUGHANEM*
Rapporteurs :
Sihem AMER-YAHIA et *Patrice BELLOT*

Event Summarization on Social Media Stream: Retrospective and Prospective Tweet Summarization

Tweet summarization

Abdelhamid CHELLAL

Juillet 2018

Manuscript submitted for the doctor of science degree
PhD student: Abdelhamid CHELLAL
Supervisor: Professor Bernard DOUSSET et Mohand BOUGHANEM
IRIT - Université de Toulouse III Paul Sabatier © September 2018

Contact me for any comments and corrections: abdelhamid.chellal@irit.fr
Institut de Recherche en Informatique de Toulouse, UMR 5505 CNRS,
Université Toulouse III Paul Sabatier,
118 route de Narbonne,
F-31062 Toulouse CEDEX 9

ACKNOWLEDGMENT

After over four years of dedicated work, it gives me a great pleasure to thank all the people who have supported me and guided me through the process of writing this thesis. First and foremost, I would like to express my sincere and deep gratitude to my supervisors Professor Mohand Boughanem and Professor Bernard Dousset for their availability, their real interest in this work from the very beginning of my research and who have advised me and helped me to get through all the challenges of this long Ph.D. study. I special thank goes to Professor Mohand Boughanem who has been a great mentor and who allocated me a lot of his time to shape my ideas and pursue my work. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D. study.

Besides my advisors, I would like to thank the reviewers of this thesis, Dr. Sihem Amer-Yahia Direc, Directrice de Recherche Première Classe at CNRS and Professor Patrice BELLOT, professor at the University of Aix-Marseille, for their careful and detailed reading of my manuscript as well as for their encouraging and insightful comments. I would also like to express my gratitude to Professor Lynda TAMINE-LECHANI, professor at the University of Paul Sabatier Toulouse, and Dr. Phillipe MULHEM, Chargé de Recherche at CNRS, for having accepted to examine my work, who were willing to participate in my final defense committee and for their brilliant comments and suggestions.

Moreover, I would like to thank my wonderful friends and colleagues at IRIT lab including (Mahadi Kandi, Mahdi washaha, Mohamed elmalki, Rafik Boudiba, Thiziri Belkacem, Imen Akermi, Paul Mousset, Gia Hung Nguyen, Ismail Baadache, Lamdjed Ben Jabeur) for supporting me and for all the fun we have had in the last four years. Also, I thank all member of IRIS team and Mr. Jacques Thomazeau administrator of OSIRIM platform for providing guidance throughout my studies and for their valuable advice.

Furthermore, I would like to express my gratitude to my friends Nounou, Lyes and Omar who have always been there when I needed them and who have supported me in difficult moments. Thank you guys

Many thanks to my friends and colleagues Abdesalme, Gano, Bachir, Adel, Tarek tizo, Moncef, Bachir, Boudjlal, Abdenour, Adlen, Maamar, Elhadi,.... All of you are great!

Last but not least, I would like to infinitely thank my parents, Mohammed and Zahia, my brothers, my sisters and my parents-in-law for believing in me, encouraging me to never give up on my dream. In closing, I would like to express my deepest gratitude to my lovely wife *Sawsen* for her support. I am deeply thankful to her. She was always cheering me up and she is the one I feel truly blessed to have had such extraordinary gracious wife, without whom I would have never been able to climb this hill. **Thank you so much *SAW*.**

ABSTRACT

User-generated content on social media, such as Twitter, provides in many cases, the latest news before traditional media, which allows having a retrospective summary of events and being updated in a timely fashion whenever a new development occurs. However, social media, while being a valuable source of information, can be also overwhelming given the volume and the velocity of published information. To shield users from being overwhelmed by irrelevant and redundant posts, retrospective summarization and prospective notification (real-time summarization) were introduced as two complementary tasks of information seeking on document streams. The former aims to select a list of relevant and non-redundant tweets that capture "what happened". In the latter, systems monitor the live posts stream and push relevant and novel notifications as soon as possible.

Our work falls within these frameworks and focuses on developing a tweet summarization approaches for the two aforementioned scenarios. It aims at providing summaries that capture the key aspects of the event of interest to help users to efficiently acquire information and follow the development of long ongoing events from social media. Nevertheless, tweet summarization task faces many challenges that stem from, on one hand, the high volume, the velocity and the variety of the published information and, on the other hand, the quality of tweets, which can vary significantly.

In the prospective notification, the core task is the relevancy and the novelty detection in real-time. For timeliness, a system may choose to push new updates in real-time or may choose to trade timeliness for higher notification quality. Our contributions address these levels: First, we introduce Word Similarity Extended Boolean Model (WSEBM), a relevance model that does not rely on stream statistics and takes advantage of word embedding model. We used word similarity instead of the traditional weighting techniques. By doing this, we overcome the shortness and word mismatch issues in tweets. The intuition behind our proposition is that context-aware similarity measure in word2vec is able to consider different words with the same semantic meaning and hence allows offsetting the word mismatch issue when calculating the similarity between a tweet and a topic. Second, we propose to compute the novelty score of the incoming tweet regarding all words of tweets already pushed to the user instead of using the pairwise comparison. The proposed novelty detection method scales better and reduces the execution time, which fits real-time tweet filtering. Third, we propose an adaptive Learning to Filter approach that leverages social signals as well as query-dependent features. To overcome the issue of relevance threshold setting, we use a binary classifier that predicts the relevance of the incoming tweet. In addition, we show the gain that can be achieved by taking advantage of ongoing relevance feedback. Finally, we adopt a real-time push strategy and we show that the proposed approach achieves a promising performance in terms of quality (relevance and novelty) with low cost of latency whereas the state-of-the-art approaches tend to trade latency for higher quality.

This thesis also explores a novel approach to generate a retrospective summary that follows a different paradigm than the majority of state-of-the-art methods. We consider the summary generation as an optimization problem that takes into account the topical

and the temporal diversity. Tweets are filtered and are incrementally clustered in two cluster types, namely topical clusters based on content similarity and temporal clusters that depends on publication time. Summary generation is formulated as integer linear problem in which unknowns variables are binaries, the objective function is to be maximized and constraints ensure that at most one post per cluster is selected with respect to the defined summary length limit.
Keywords: Information retrieval, Real-time tweet filtering, Tweet summarization, Social signals, Adaptive Learning, Integer Linear Programming.

RÉSUMÉ

Le contenu généré dans les médias sociaux comme Twitter permet aux utilisateurs d'avoir un aperçu rétrospectif d'évènement et de suivre les nouveaux développements dès qu'ils se produisent. Cependant, bien que Twitter soit une source d'information importante, il est caractérisé par le volume et la vélocité des informations publiées qui rendent difficile le suivi de l'évolution des évènements. Pour permettre de mieux tirer profit de ce nouveau vecteur d'information, deux tâches complémentaires de recherche d'information dans les médias sociaux ont été introduites : la génération de résumé rétrospectif qui vise à sélectionner les tweets pertinents et non redondant récapitulant "ce qui s'est passé" et l'envoi des notifications prospectives dès qu'une nouvelle information pertinente est détectée.

Notre travail s'inscrit dans ce cadre. L'objectif de cette thèse est de faciliter le suivi d'événement, en fournissant des outils de génération de synthèse adaptés à ce vecteur d'information. Les défis majeurs sous-jacents à notre problématique découlent d'une part du volume, de la vélocité et de la variété des contenus publiés et, d'autre part, de la qualité des tweets qui peut varier d'une manière considérable.

La tâche principale dans la notification prospective est l'identification en temps réel des tweets pertinents et non redondants. Le système peut choisir de retourner les nouveaux tweets dès leurs détections où bien de différer leur envoi afin de s'assurer de leur qualité. Dans ce contexte, nos contributions se situent à ces différents niveaux : Premièrement, nous introduisons Word Similarity Extended Boolean Model (WSEBM), un modèle d'estimation de la pertinence qui exploite la similarité entre les termes basée sur le word embedding et qui n'utilise pas les statistiques de flux. L'intuition sous-jacente à notre proposition est que la mesure de similarité à base de word embedding est capable de considérer des mots différents ayant la même sémantique ce qui permet de compenser le non-appariement des termes lors du calcul de la pertinence. Deuxièmement, l'estimation de nouveauté d'un tweet entrant est basée sur la comparaison de ses termes avec les termes des tweets déjà envoyés au lieu d'utiliser la comparaison tweet à tweet. Cette méthode offre un meilleur passage à l'échelle et permet de réduire le temps d'exécution. Troisièmement, pour contourner le problème du seuillage de pertinence, nous utilisons un classificateur binaire qui prédit la pertinence. L'approche proposée est basée sur l'apprentissage supervisé adaptatif dans laquelle les signes sociaux sont combinés avec les autres facteurs de pertinence dépendants de la requête. De plus, le retour des jugements de pertinence est exploité pour re-entrainer le modèle de classification. Enfin, nous montrons que l'approche proposée, qui envoie les notifications en temps réel, permet d'obtenir des performances prometteuses en termes de qualité (pertinence et nouveauté) avec une faible latence alors que les approches de l'état de l'art tendent à favoriser la qualité au détriment de la latence.

Cette thèse explore également une nouvelle approche de génération du résumé rétrospectif qui suit un paradigme différent de la majorité des méthodes de l'état de l'art. Nous proposons de modéliser le processus de génération de synthèse sous forme d'un problème d'optimisation linéaire qui prend en compte la diversité temporelle des tweets. Les tweets sont filtrés et regroupés d'une manière incrémentale en deux partitions basées respectivement sur la similarité du contenu et le temps de publication. Nous formulons la génération du résumé comme étant un problème linéaire entier dans lequel les variables inconnues sont binaires, la fonction objective est à maximiser et les contraintes assurent qu'au maximum un tweet par cluster est sélectionné dans la limite de la longueur du résumé fixée préalablement.

Mots clés : Recherche d'information, Filtrage temps réel de flux de tweets, Synthèse de tweets, Signes sociaux, apprentissage adaptatif, Optimisation linéaire.

Success is the sum of small efforts, repeated day in and day out.

— Robert Collier

PUBLICATIONS

Our ideas and contributions have already been published in the following scientific publications:

International conference papers

1. Abdelhamid Chellal, Mohand Boughanem. Optimization Framework Model For Retrospective Tweet Summarization. Dans : ACM Symposium on Applied Computing (SAC 2018), Pau, France, 09/04/2018-13/04/2018, ACM, p. 699-708, avril 2018 (à paraître). http://doi.org/10.1145/3167132.3167210
2. Abdelhamid Chellal, Mohand Boughanem, Bernard Dousset. Word Similarity Based Model for Tweet Stream Prospective Notification (short paper). Dans : European Conference on Information Retrieval (ECIR 2017), Aberdeen, Scotland, UK, 08/04/2017-13/04/2017, Springer-Verlag, P 655—661.
3. Abdelhamid Chellal, Mohand Boughanem. IRIT at TREC Real-Time Summarization 2017 (regular paper). Dans : Text REtrieval Conference (TREC 2017), National Institute of Standards and Technology Gaithersburg, Maryland USA, 15/11/2017-17/11/2017, National Institute of Standards and Technology (NIST), (en ligne), November 2017. Résumé Accès : $http : //trec.nist.gov/act_part/conference/papers/IRIT - RT.pdf$
4. Abdelhamid Chellal, Bernard Dousset. Impact of Social Signals in Real Time Tweet Filtering and Summarization task. Dans : International Symposium on Interdisciplinarity, Corte, 05/07/2017-07/07/2017.
5. Bilel Moulahi, Lamjed Ben Jabeur, Abdelhamid Chellal, Thomas Palmer, Lynda Tamine, Mohand Boughanem, Karen Pinel-Sauvagnat, Gilles Hubert. IRIT at TREC Real Time Summarization 2016 (regular paper). Dans : Text REtrieval Conference (TREC 2016), Gaithersburg, Maryland USA, 15/11/2016-18/11/2016, 2016
6. Abdelhamid Chellal, Mohand Boughanem, Bernard Dousset. Multi-criterion real time tweet summarization based upon adaptive threshold (regular paper). Dans : IEEE/WIC/ACM International Conference on Web Intelligence, Omaha, Nebraska, USA, 13/10/2016-16/10/2016, IEEE Computer Society, p. 264-271, octobre 2016.
7. Abdelhamid Chellal, Mohand Boughanem. IRIT at the NTCIR-12 MobileClick-2 Task (regular paper). Dans : Conference on Evaluation of Information Access Technologies (NTCIR 2016), Tokyo, Japan, 07/06/2016-10/06/2016, National Institute of Informatics, p. 143-146, juin 2016.
8. Rafik Abbes, Bilel Moulahi, Abdelhamid Chellal, Karen Pinel-Sauvagnat, Nathalie Hernandez, Mohand Boughanem, Lynda Tamine, Sadok Ben Yahia. IRIT at TREC Temporal Summarization 2015 (regular paper). Dans : Text REtrieval Conference (TREC 2015), Gaithersburg, Maryland USA, 17/11/2015-20/11/2015, National Institute of Standards and Technology (NIST), (en ligne), novembre 2015.

9. Abdelhamid Chellal, Lamjed Ben Jabeur, Laure Soulier, Bilel Moulahi, Thomas Palmer, Mohand Boughanem, Karen Pinel-Sauvagnat, Lynda Tamine, Gilles Hubert. IRIT at TREC Microblog 2015 (regular paper). Dans : Text REtrieval Conference (TREC 2015), Gaithersburg, Maryland USA, 17/11/2015-20/11/2015, National Institute of Standards and Technology (NIST), (en ligne), novembre 2015. Accès : $http : //trec.nist.gov/act_part/conference/papers/IRIT - MB.pdf$.

National conference papers

1. Abdelhamid Chellal, Mohand Boughanem, Bernard Dousset. Synthèse de flux de messages en temps réel (regular paper). Dans : Conférence francophone en Recherche d'Information et Applications (CORIA 2016), Toulouse, 09/03/2016-11/03/2016, Association Francophone de Recherche d'Information et Applications (ARIA), p. 515-529, mars 2016.

Submitted papers

1. Abdelhamid Chellal, Mohand Boughanem. Adaptive Learning Strategy Leveraging Social Signals for Real-time Tweet Summarization (Full Length Article). submitted to Information Processing & Management journal.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

API	Application Programming Interface
DCG	Discounted Cumulative Gain
EBM	Extended Boolean Model
ELG	Expected Latency-discounted Gain
EG	Expected Gain
JSON	JavaScript Object Notation
LDA	Latent Dirichlet Allocation
IDCG	Ideal Discounted Cumulative Gain
IR	Information Retrieval
IRS	Information Retrieval System
ILP	Integer Linear Programming
IDF	Inverse Document Frequency
KL-divergence	Kullback Leibler divergence
MB	Microblog
nCG	Normalized Cumulative Gain
nDCG	Normalized Discounted Cumulative Gain
NER	Named-entity Recognition
NIST	National Institute of Standards and Technology
RTF	Real-Time Filtering
RTS	Real-Time summarization
ROC	Receiver Operating Characteristic
REST	REpresentational State Transfer,
ROUGE	Recall-Oriented Understudy for Gisting Evaluation3
VSM	Support Vector Machines
TF	Term Frequency
TDT	Topic Detection and Tracking

TF-IDF	Term Frequency-Inverse Document Frequency
TPM	Tweet Propagation Model
TREC	Text REtrieval Conference
TOWGS	Twitter Online Word Graph Summarizer
UGC	User generated content
URL	Uniform Resource Locator
WSEBM	Word Similarity Extended Boolean Model

Part I

INTRODUCTION

Social media is an amazing tool, but it's really the face-to-face interaction that makes a long-term impact.

— Felicia Day

1 INTRODUCTION

1.1 Context

With the rise of Web 2.0, the World Wide Web (WWW) evolved from being static, where users were only consuming information, to a new version of Web where users have the ability to act as an information producer and consumer at the same time. The websites that enable people to create, share, or exchange information are known as online Social Media. Social media range from Blogs, Wikis, Microblogs, Social Networking, Media Sharing and Social Bookmarking. As an example of the most popular platform, we can mention: Twitter [1] (2006), Facebook [2] (2004), G+ [3], LinkedIn [4] (2006), Myspace[5](2003), Youtube [6] (2005), Wikipedia [7], and Flickr [8] (2004). These tools allow connectivity and interaction between web users and they encourage contributions and feedback from anyone who is a member of any virtual community [1]. These platforms continue to increasingly gain popularity and the number of their users worldwide is increasing every day. Figure 1.1 illustrates the number of social media users worldwide from 2010 to 2017 with an emphasis on the number of monthly active users on Twitter. These statistics clearly show that the number of social media users is getting increasing almost in a linear way with an average rate of about 0.3 billion users per year. In 2019, it is estimated that there will be around 2.77 billion social media users, up from 2.46 billion in 2017 [9].

With social media, a user becomes an individual news media that not only consumes/ absorbs information but also produces/propagates information about what is happening in the world or what is being said about an entity. People are spending countless hours on social media and their activities generate an incredible volume of data. Data published/produced by users in social media is known as User Generated Content (UGC). UGC covers a wide range of topics, from personal issues (i.e about users daily activity) to public policy (i. e.related to a topic of interest to a wide audience). The quick development of smartphone technology has played an important role in the explosive growth of the use of social media. Indeed, thanks to mobile devices, users can instantly report a real-world event in an in-situ manner. In Twitter, for instance, more than 80% of daily published tweets are posted from mobile devices [12]. Users publish in social media a valuable information that provides in many cases live coverage of scheduled (sports games) and unscheduled events (natural disaster).

[1] http://twitter.com/
[2] http://facebook.com/
[3] https://plus.google.com/
[4] https://www.linkedin.com/
[5] https://myspace.com/
[6] https://www.youtube.com/
[7] https://www.wikipedia.org/
[8] https://www.flickr.com/
[9] https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/
[12] https://expandedramblings.com/index.php/twitter-mobile-statistics/

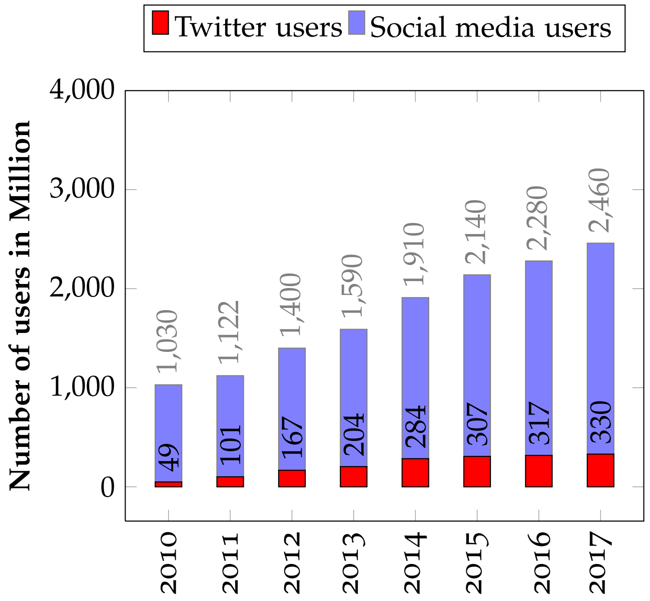
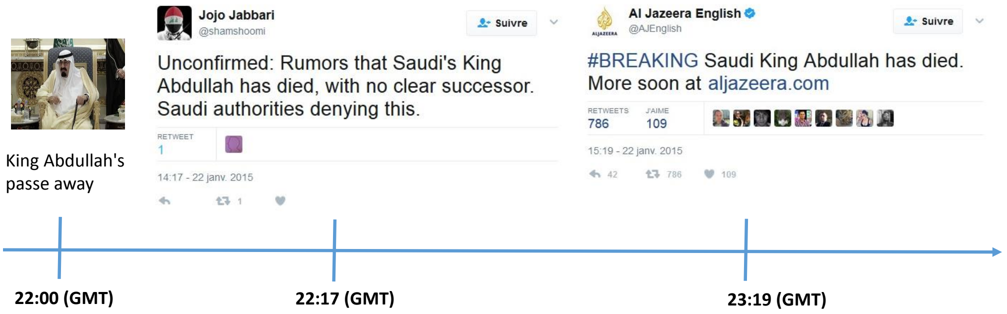


Figure 1.1: Number of users worldwide from 2010 to 2017 in all social media[10] and on Twitter[11].

The freshness and diversity of information in UGC have raised social media as an important source of real-time information to up-to-date about an ongoing event. Besides, it provides, in many cases, the latest news before traditional media, especially for unscheduled events. Hu et al. [65] have shown that quite a few big news stories have been broken on Twitter earlier than in more traditional news media. An example of this is the news about Osama bin Laden's death. Several sources on Twitter leaked the information before the President of the United States announced that bin Laden had been killed [65]. Figure 1.2 shows, another example, the tweet on the left leaked information about Saudi King Abdullah's death 17 minutes after he had passed away (at 22:00 GMT [13]) and more than one hour before that Al Jazeera news announced it as shown in the following tweet.


Jojo Jabbari
@shamshoomi

Unconfirmed: Rumors that Saudi's King Abdullah has died, with no clear successor. Saudi authorities denying this.

RETWEET 1

14:17 - 22 janv. 2015

Al Jazeera English
@AJEnglish

#BREAKING Saudi King Abdullah has died. More soon at aljazeera.com

RETWEETS 786 J'AIME 109

15:19 - 22 janv. 2015

Figure 1.2: Example of breaking new in Twitter before it breaks in traditional news media: The death of Saudi's King Abdullah.

Social Media such as Twitter, Facebook, are also important sources of real-time information related to emergency events which include natural disasters such as earth-

[13] http://www.bbc.com/news/world-middle-east-30945324

quakes, as well as man-made disasters such as terror attacks, and socio-political movements. During emergencies, users are turning to social media platforms to ask for assistance and to share what they see around them. Indeed, in [47, 148] author have shown that Twitter is useful to detect earthquake just by monitoring tweets containing the word "earthquake" and other related words. They assume that a peak in earthquake-related tweets correlates with an earthquake event. In doing this, 75% of earthquakes are detected by Twitter within two minutes and, as such, outpace traditional geological survey detection [148].

Social media as a source of real word event information is a double-edged sword [143]. Indeed, social media in their raw form, while being informative, can also be overwhelming. In Twitter, short messages known as tweets are being created and shared at an unprecedented rate. Twitter counts more than 330 million monthly active users worldwide that produce over 500 million tweets every day which corresponds to over 350,000 tweets sent per minute [14]. Hence, due to the high volume of daily produced posts, the velocity, and the variety in tweets stream, monitoring and following all published information describing the development of a given event over time or referring to an entity turns out to be time-consuming with a risk of overwhelming users with irrelevant and redundant posts.

When an important event occurs particularly an unscheduled one, users may develop two complementary information needs. The first one is retrospective, in which a user wants to be informed on what has occurred up until now regarding the event of interest, while the second need is prospective, it refers to the wish of the user to be kept up-to-date in a timely fashion whenever a new development occurs. In both cases, the main purpose is to produce a summary consisting of a list of tweets that fulfill the user information need. To achieve such a goal, a tweet summarization system has to monitor the live stream of tweets. Accordingly, two different scenarios of tweet summarization can be distinguished, namely retrospective tweet summarization and prospective (so-called real-time) tweet summarization. In the latter scenario, tweets are processed in real-time whereas in the former scenario tweets are treated on a batch.

Our work falls within these frameworks and focuses on the problem of a long ongoing event summarization from social media in which we tackle the two aforementioned scenarios of tweet summarization. The purpose of this thesis is helping users to efficiently acquire social media information and follow the development of an event of interest by providing summarization approaches for long ongoing events. In this thesis, we focus on Twitter as it is one of the most popular social media and unlike most social media it allows free access to a large part of its data, which motivate many researchers to study it.

In the following subsection, we detail the two complementary information needs that we consider in our work. Afterward, we describe the main issues challenging these two tasks of tweet summarization.

1.1.1 *Retrospective information need in social media*

In retrospective information need, a user is looking for what has happened thus far regarding an event that he just heard about it. A retrospective summary should capture

[14] http://www.internetlivestats.com/twitter-statistics/

what has occurred up until now so the user makes up what he would have missed. It is required that such summary be concise and contains relevant tweets that convoy the main aspects of the information need with a minimum of redundancy [87, 86, 85]. In addition to these requirements, it is desired that the summary includes information from different time periods to capture the development of the event over time.

To fulfill this information need, retrospective tweet summarization task was introduced with the aim to automatically select a list of meaningful tweets or keywords that are most representative of the given topic. This task is more like ad hoc tweet retrieval[85, 120].

1.1.2 *Prospective information need in social media*

In prospective information need, a user wishes to be notified whenever there are new developments regarding the event of interest. In this context, the relevance of a post with respect to the user's information need does not depend only on the content of the post (topical relevance) but it is more a combination of interesting content, novelty, and timely [129].

To address users' prospective information needs, it is required that a real-time tweet summarization system monitors the live streams of social media stream posts to identify relevant and novel content with respect to a given topic of interest. The goal is to push to the user salient posts that convey novel information with a minimum of latency between the publication time of the post and the time of notification to the user. It is desired that relevant and novelty tweets are pushed to the user in timely "fashion" (e. g.as notifications on his mobile phone).

1.2 Challenges of tweet summarization

Traditional document summarization is extensively studied [111, 163]. However, besides the fact that the proposed approaches in the context of traditional document summarization are retrospective in nature, these methods turn out to be less effective when handling tweets. This is due to the many issues that challenge the task of tweet summarization. These challenges stem from:

1. The streaming character of UGC in social media which is characterized by the high volume, the velocity, the variety of the published information, and the topic drift. The information reported is often highly redundant. In addition to that, the streaming character of social media makes that statistics about a term such as (Inverse Document frequency) (i) are not always available in particular at the beginning of monitoring the tweet stream and (ii) change each time a new tweet arrives;

2. The quality of the published posts which varies significantly. Indeed, tweets are short, informal, and with many abbreviations and misspellings. These arise two issues when computing the relevance of tweet with respect to a topic of interest. The first issue is the term mismatch and the second issue is related to the fact that term frequency is usefulness since a term rarely occurs more than once in a tweet.

The main objective of tweet summarization approaches is to select the most meaningful tweets that capture the key aspects of the event and its development over time with a minimum of redundancy. However, to be effective a tweet summarization system has to satisfy the following requirements:

- In retrospective summarization, summaries are expected to fulfill some important requirements such as relevancy with respect to the event, low redundancy/novelty, coverage (capture different aspect of the event), diversity and conciseness of the summary. Optimizing all these criteria jointly is a challenging task especially for long-running events [142]. The majority of the proposed approaches [67, 142, 168, 141, 123, 171, 97] generate summaries by iteratively selecting the most relevant tweets and discarding those having their similarity with respect to the current summary above a certain threshold. This way of selecting meaningful tweets do not take into the account the mutual relation among tweets. In addition, it does not consider the fact that important tweets may be spread out over the lifetime of the given event.

- Regarding prospective summarization, besides the relevancy and the low redundancy/novelty requirements a system has to:
  - Find a trade-off between pushing too many or too few tweets. In the latter case, the user may miss important updates and in the former case, the user may be overwhelmed by irrelevant and/or redundant information.
  - Balance between the timeliness (latency between publication and notification times) and the quality of notification. In this context, the incoming tweet that is identified as relevant and novel can be immediately pushed to the user or the system may choose to wait in order to accumulate evidence and see whether it is worth pushing. However, by delaying the submission of an interesting tweet, a system may miss the appropriate time windows for pushing the given tweet. Indeed, at the time the system submits the given tweet it may have become outdated. This is because, in the case of an ongoing event, information has generally a short lifetime and can rapidly become outdated.

1.3 Research Questions

This thesis focuses on the problem of summarization a long ongoing event in the social media stream with special attention dedicated to Twitter. With respect to the prospective tweet summarization, three main research questions are being addressed:

1. How can we evaluate the relevance of tweets without relying on stream statistic?

2. How can we overcome the issue of threshold setting?

   A How does machine learning based method contribute to improving the filtering performance?

   B What gain can be achieved by using an adaptive learning strategy that takes advantage of ongoing relevance feedback?

3. What is the impact of social signal in real-time tweet filtering?
    A What are the social features that can be suitable for real-time tweet filtering?
    B To what extent the consideration of social features is helpful in detecting relevant tweets?

Regarding, the retrospective tweet summarization, the following research questions were investigated:

A How can we optimize jointly all the criteria that a retrospective summary should fulfill?

B How to integrate the temporal context of tweets into the process of the generation of a retrospective summary?

1.4 Contributions

The work in this thesis focuses on developing methods for addressing the challenges raised in the two scenarios of tweet summarization described above: prospective and retrospective tweet summarization.

Regarding prospective tweet summarization, the main contributions of this thesis are at several levels:

- First, *relevance estimation*: we introduce Word Similarity Extended Boolean Model (WSEBM), a relevance model that does not rely on stream statistics when computing the relevance score of the incoming tweet by taking advantage of word embedding model (word2vec [106]). In WSEBM, the word similarity is used instead of the traditional weighting techniques used in Information Retrieval (IR) models. By doing this, we overcome the shortness and word mismatch issues in tweets. The intuition behind our approach being that context-aware similarity measure in word2vec is able to consider different words with the same semantic meaning and hence allows offsetting the word mismatch issue when calculating the similarity between a tweet and a topic. In addition, the relevance score of the incoming tweet is estimated at the time the new tweet arrives independently of the previously seen tweets and without the need for maintaining statistics about tweet stream to capture collection based parameter such as (inverse) document frequency, average document length, etc....

- Second, *novelty detection*: Instead of using the pairwise comparison to compute the novelty score of the incoming tweet against tweets previously seen by the user, we propose to compute the novelty score regarding all words of tweets already pushed to the user. The proposed novelty detection method scales better and reduces the execution time, which fits better the real-time tweet filtering scenario.

- Third, *relevance filtering*: To overcome the relevance threshold setting issue, we propose an adaptive Learning to Filter approach based on supervised machine learning algorithm after arguing that the key to effective notifications lies on identifying an appropriate relevance threshold value in which the decision to select/ignore an incoming tweet is based. Our contribution concerns the use of a binary classifier (relevant/not relevant) that predicts the relevance of the incoming tweet which allows overcoming the issue of relevance threshold setting. To enhance the ability of the classifier to identify correctly relevant tweets, we leverage social signals as well as query-dependent features. In this context, we proposed a set of social features and other non-content features suitable for real-time tweet filtering and we study their impact in tweet summarization. In addition, we extend the Learn to Filter approach to an adaptive learning approach which takes advantage of ongoing relevance assessments feedback to periodically retrain the classification model. Hence, we show the gain that can be achieved by an adaptive learning strategy that takes advantage of ongoing relevance feedback.

Regarding retrospective tweet summarization, we propose an alternative method that follows a different paradigm than the majority of state-of-the-art methods. Our contributions are as follows:

- While existing approaches generate a summary by selecting iteratively top weighted tweets, we consider the summary generation as an optimization problem to select a subset of tweets that maximizes the global summary relevance and fulfills constraints related to non-redundancy, coverage, temporal diversity and summary length. To this aim, tweets are filtered and incrementally clustered in two cluster types, namely topical and temporal clusters. The former clustering is based on tweet content similarity whereas the latter depends on tweet publication time. The summary generation is formulated as an integer linear problem in which unknowns variables are binaries, the objective function is to be maximized and constraints ensure that at most one post per cluster from the two categories of clusters (topical and temporal) is selected with respect to the defined summary length limit.
- In order to capture the development of the event over its lifetime, we take into account the temporal diversity of tweets as one criterion that needs to be fulfilled in the summary generation process. The fact that a maximum of one tweet per time period (temporal cluster) is selected guarantees that the summary covers different time periods as much as possible.

1.5 Thesis outline

This thesis is divided into five parts. The first part (this part) contains the introduction of this work. The second part is about background knowledge and state of the art. The third and fourth parts focus on our contributions for prospective (so-called real-time) tweet summarization and for retrospective tweet summarization respectively. The last part presents conclusions and future work. In this section, we will introduce the contents of each part.

- In part 2, we present background knowledge and state of the art. This part entails three chapters as follows:

  - Chapter 2 introduces the concepts and background of Information Retrieval (IR) and social media that will be used throughout this thesis. In this chapter, we particularly focus on Twitter;
  - Chapter 3 focuses on tweet summarization and presents a survey of related work that addresses the two scenarios of tweet summarization, namely prospective and retrospective tweet summarization;
  - Chapter 4 describes the datasets and the evaluation frameworks that have been used to evaluate our approaches.

- In part 2, we focus on our contribution for prospective tweet summarization. This part is composed of two chapters as follows:
  - Chapter 5 introduces a word similarity based model for real-time tweet summarization First, we present the general framework of the proposed approach. Afterward, we detail the computation of the relevance with respect to a given event then the estimation of the novelty of an incoming tweet with regard to tweets previously seen by the user. At the end of this chapter, we highlight the importance of proper relevance threshold setting in the task of real-time tweet filtering.
  - Chapter 6 presents a learning to filter approach based on supervised machine learning technique to overcome the issue of relevance threshold setting. First, we give a description of the proposed approach that combines query-dependent features with social features to build a binary classifier that predicts the relevancy of incoming tweets. Then, we present a set of social features suitable for real-time tweet filtering scenario. Afterward, we investigate to what extent the use of adaptive learning approach that takes advantage of an ongoing relevance feedback can improve the performance of real-time tweet filtering system.

- In part 3, we present our contribution for retrospective tweet summarization. Chapter 7 describes our approach based on an optimization framework to generate a retrospective tweet summary for a long ongoing event. First, we introduce the description of the proposed approach that aims to select a subset of tweets that maximizes the global summary relevance and fulfills constraints related to non-redundancy, coverage, temporal diversity and summary length. Then, we describe the incremental tweet clustering method. Thereafter, we present the Integer Linear Programming model that formulates the problem of tweet summary generation.

- The last part, Chapter 8, concludes this thesis, discusses findings and points out directions for future research.

Part II

BACKGROUND AND RELATED WORK

The idea of Twitter started with me working in dispatch since I was 15 years old, where taxi cabs or firetrucks would broadcast where they were and what they were doing.

— Jack Dorsey, Twitter CEO

2 BACKGROUND: INFORMATION RETRIEVAL AND SOCIAL MEDIA

In this chapter, we provide the concepts and background that will be used throughout this thesis. We start with a brief introduction to the field of information retrieval. Afterwards, we present a general overview of social media and then we focus on information retrieval for social media.

2.1 Information Retrieval

Information retrieval (IR) is about finding material (information) that satisfies an information need within a large collection of documents [99]. According to Baeza-Yates and Ribeiro-Neto [20], information retrieval deals with the representation, storage, organization of, and access to information items. Manning et al. [99] define information retrieval as:

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

The main purpose of an information retrieval system (IRS) is to fulfill user information need which is usually formulated using a textual query. Salton and McGill [135] defined information retrieval system as the set of processes that provide the user with information:

An information retrieval system is an information system, that is, a system used to store items of information that need to be processed, searched, retrieved, and disseminated to various user populations.

We provide in this section the basic concepts of information retrieval process then we present an overview of information retrieval models.

2.1.1 *Information Retrieval Process*

Information Retrieval consists mainly of building up efficient indexes, processing user queries and ranking algorithms to select documents that are relevant to user query [20]. These three main processes are more or less complex depending on the retrieval task. Figure 2.1 illustrates a simplified view of the IR process which is referred to as U-process [22].

Indexing is in general performed at the beginning of an information retrieval cycle and periodically each time the collection is updated with an important volume of new documents. The main purpose of this process is to ensure fast and efficient querying. It enables the mapping between terms and documents where they occur. The most common form of indexes in text collections is the *inverted index*. It is composed of a dictionary and posting lists. The dictionary contains all terms assigned to or extracted

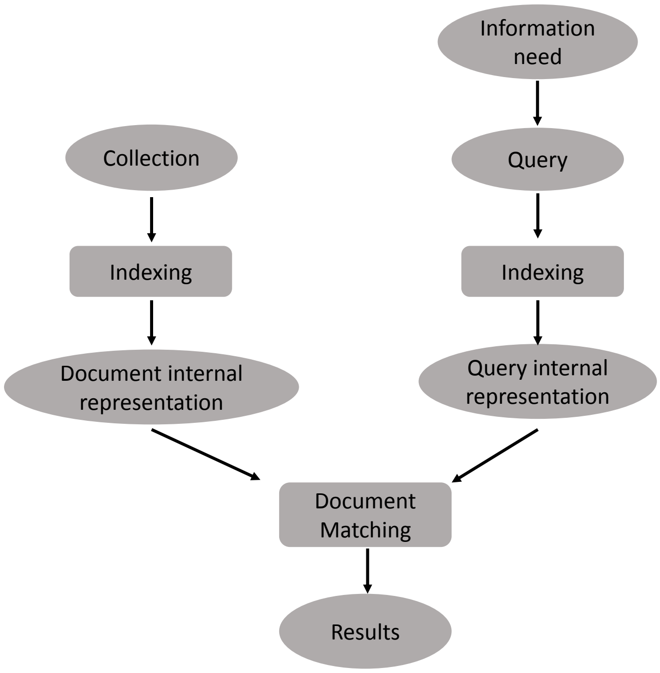


Figure 2.1: The Information Retrieval U-process: simplified schema [22].

from documents that are considered useful. Each term in the vocabulary has a posting list that indicates documents in which it appears. To build the index, documents are processed to extract only useful terms and their features (e.g frequency, position, ...). The following steps are applied during the indexing.

- Extracting a normalized sequence of characters: This step keeps just a linear textual sequence of document content and removes all format specific structure such as pdf, Microsoft Word format, and HTML;

- Tokenization: This corresponds to the task of splitting the textual sequence into terms (also called tokens). Tokenization is also associated with punctuation removal.

- Stop-word removal: Stop-words are terms that might be not important for information retrieval such as frequent terms "*the*", "*an*", "*a*", "*and*". These type of words are removed using a predefined list of stop-word specific for each language.

- Normalization: In this step, tokens that should match with each other are gathered in an equivalence class. For instance, words USA, US, U.S, U.S.A are mapped to USA.

- Lemmatization and stemming: The goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form [1]. For example,"computing", "computes", "computed" and "computation" are all different syntactic forms of "compute". Stemming is based

[1] https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html

on some heuristics which consist in removing word endings. Lemmatization corresponds to the removal of inflectional endings to return the dictionary base of a word also known as lemma. One of the most used techniques in IR is the Porter stemming algorithm [119] which is empirically shown to be very effective.

- Term weighting: Term weighting attempts to capture how important a word is to a document. The widely used term weighting scheme is TF-IDF that combines inverse document frequency (IDF) with term frequency (TF) as proposed by Salton et al. [134]. The TF component is based on the observation that frequent terms are important for describing the main topics of a document. The IDF component is known for capturing term specificity. The idea behind IDF is that a common term that appears in many documents is unlikely to be discriminative. There exists a probabilistic definition of IDF used in particular in the BM25 model[127] were IDF is interpreted as the probability of the term appearing if the document is irrelevant with respect to the query. Table 2.1 lists some weighting variants used in the literature to compute TF and IDF. This step produces some statistics used to compute the relevance score of a document with regard to the user's information need in a query matching process.

Table 2.1: Variants of TF and IDF weights [20, 72]

weighting scheme	TF	weighting scheme	IDF
binary	{0,1}	unary	1
raw frequency	$f_{i,j}$	inverse frequency	$log(\frac{N}{df_i})$
log normalization	$1 + log f_{i,j}$	inverse frequency smooth	$log(1 + \frac{N}{df_i})$
double normalization 0.5	$0.5 + 0.5\frac{f_{i,j}}{max f_{i,j}}$	inverse frequency max	$log(1 + \frac{maxdf_i}{df_i})$
double normalization k	$0.5 + (1 - 0.5)\frac{f_{i,j}}{max f_{i,j}}$	probabilistic inverse frequency	$log\frac{N-df_i}{n_i}$

Where $f_{i,j}$ is the frequency of occurrence of the i_{th} term in document j, dfi is the number of documents in which the term i occurs and N is the number of documents in the collection.

Querying This is the stage where the user issues his information need to the IR system. Usually, the information need is expressed in a short free-text query that includes a few words. To enrich the initial query it is common to extend it with other related terms. This process is called query expansion. The query passes through the same steps as documents in the indexing process.

Query-document matching This process assigns a relevance score to documents according to their similarity with the query. The relevance score indicates the relative or absolute degree of presumed relevance of a document with respect to the user's information need. In Boolean IR model [133], this score is binary (0 or 1) where the 0 score indicates that the document is irrelevant and the 1 score denotes that the document is relevant. The main drawback of the Boolean model is that all documents considered relevant are equal and cannot be differentiated and the terms frequency is not considered. Current IR models introduce a partial matching between a query and a document by assigning real value scores to documents. This allows ranking documents with each other. In response to the query, the system returns a list of documents ranked in descending order of their relevance score or by another criterion that the user may select.

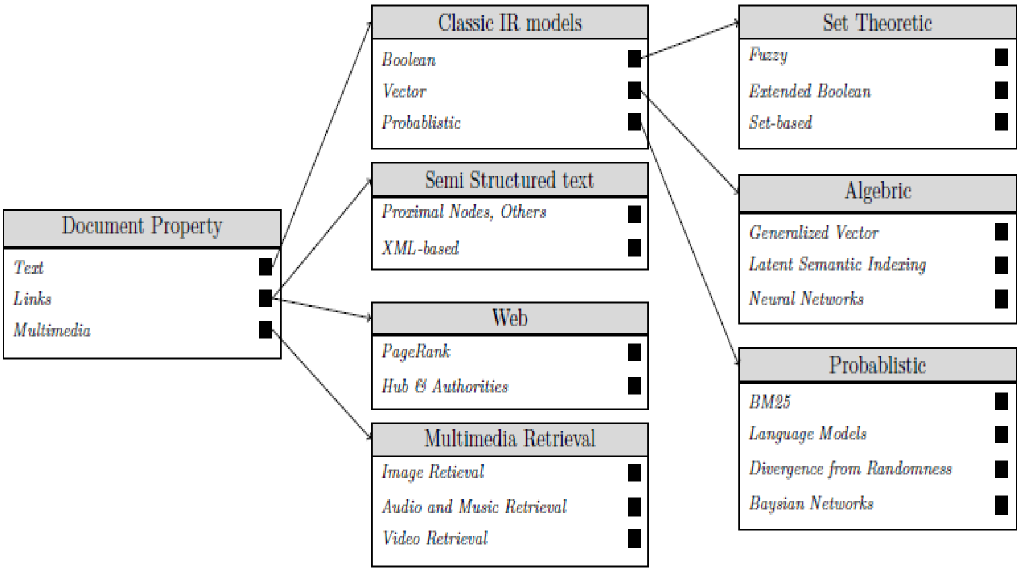


Figure 2.2: A taxonomy of Information Retrieval models [20].

2.1.2 *Information Retrieval models*

An information retrieval model aims to produce a ranking function that assigns scores to documents with respect to a given query. It defines a logical framework for representing documents and user's information need and a ranking function that computes a rank for each document. According to [20], an information retrieval model can be defined by a quadruple $[D, Q, F, R(q_i, dj]$ where: D and Q are a set of logical view (or representation) of the documents in the collection and the user's information need represssively. F is a framework for modeling the representation of documents and queries and their relationships such as a set of Boolean relations, vectors operation, and probability distribution, $R(q_i, dj)$ is a ranking function that associates a score (real number) to document d_j with regard to a query q_i. This value defines an ordering among documents with respect to the query. Figure 2.2 presents a taxonomy of information retrieval models proposed in the literature that address three main characteristics of documents including text, links, and multimedia. For unstructured text-based information retrieval models, documents are modeled as a sequence of words (bag of word representation). we distinguish the three classic models, namely Boolean, vector and probabilistic models.

We focus in what follows on the main information retrieval models that we use in this thesis namely the vector space model, the language model and the Extended Boolean Model(EBM). Note that the two first models are the most widely used models in the literature to measure the relevance of a tweet with respect to the user information need. The EBM is the one that we used in our contribution. For an exhaustive presentation of different IR models, the reader can refer to one of the multiples manuals that provide a thorough description of IR models and technologies such as [20, 99].

2.1.2.1 *Vector space model*

The vector space model was proposed in 1975 by Salton et al. [132] with the aim to introduce a partial matching between query and document terms. In this model, both the query and the document are represented by n-dimensional vectors of terms weights where n is the total number of index terms. The relevance score of document d_j with regard to the query Q is estimated according to similarity between the vector $\vec{d_j} = (w_{1,j}, w_{2,j}, ..., w_{n,j})$ and the vector $\vec{q} = (w_{1,q}, w_{s,q}, ..., w_{n,q})$. This similarity can be measured by the cosine of the angle between these two vectors as follows:

$$sim(\vec{d_j}, \vec{q}) = \frac{\vec{d_j} \cdot \vec{q}}{|\vec{d_j}| \times |\vec{q}|} = \frac{\sum_{i=1}^{n} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2} \times \sqrt{\sum_{i=1}^{n} w_{i,q}^2}} \tag{2.1}$$

Where $w_{i,j}$ and $w_{i,q}$ are the weights of term w_i in document d_j and query q respectively which are basically TF-IDF weights computed as follows:

$$w_{i,j} = TF(w_i, d_j) \times IDF(w_i) \tag{2.2}$$

Where TF (term frequency) and IDF (inverse document frequency) can be any of the functions presented in Table 2.1.

The main advantages of the vector space model are [20]: (1) the use of term-weight that take into account the document length normalization which improves the retrieval quality; (2) the documents are ranked according to their similarity to the query; (3) it is simple and fast. For these reasons, the vector space model continues to be used as retrieval model specifically as a baseline in the evaluation of new ranking models.

2.1.2.2 *Language model*

Language models in IR define probability distributions of terms in documents and use them to predict the likelihood of observing query terms [20]. In these models, it is assumed that the query is inferred by the user from the ideal documents [117]. The idea behind these models is to assume that a query is generated from the document language model [117, 164]. The relevance of a document with respect to a query is seen as the probability that the document's language model would generate the terms of the query. The first IR model based on, such hypothesis was proposed by Ponte and Croft [117] in which if one assumes that terms are independent (uni-gram model), the probability $P(Q|\theta_D)$ of a query Q being generated by the language model θ_D of document D is defined as follows:

$$P(Q|\theta_D) = \prod_{q_i \in Q} P(q_i|\theta_D) \tag{2.3}$$

Where $P(q_i|\theta_D)$ is the probability that term q_i occurs in the language model θ_D which is estimated using Maximum Likelihood (ML) estimation as follows:

$$P_{ML}(q_i|\theta_D) = \frac{f_D(q_i)}{|D|} \tag{2.4}$$

Where $f_D(q_i)$ is the frequency of the term q_i in the document D and $|D|$ is the number of terms in document D.

The maximum likelihood estimate would result in a zero probability if at least one of the query terms does not occur in the document. To overcome this issue, smoothing techniques, particularly those where the document language model is combined with the collection language model θ_C have been proposed[164]. The smoothing methods commonly used are the Dirichlet (DIR) smoothing and Jelinek-Mercer (JM) smoothing which are defined according to the following formula respectively:

$$P_{Dir}(q_i|\theta_D) = \frac{f_D(q_i) + \mu \cdot P_{ML}(q_i|\theta_C)}{|D| + \mu} \tag{2.5}$$

$$P_{JM}(q_i|\theta_D) = \lambda P_{ML}(q_i|\theta_D) + (1-\lambda) P_{ML}(q_i|\theta_C) \tag{2.6}$$

Where λ and μ are Jelinek-Mercerthe and Dirichlet smoothing parameters respectively. $P_{ML}(q_i|\theta_C)$ is the maximum likelihood probability of term q_i occurring in the collection language model θ_C which is computed as follows:

$$P_{ML}(q_i|\theta_C) = \frac{f_C(q_i)}{|C|} \tag{2.7}$$

Where $f_C(q_i)$ is the frequency of term q_i in the collection of documents and $|C|$ is the number of terms in collection C.

2.1.2.3 *Extended boolean model*

The Boolean model [133] is based on set theory. It relies on exact matches and takes only into account the presence and absence of a term in the documents without considering any term weighting. The Extended boolean model, introduced in 1983 by Salton et al. [136], extends the Boolean model with the functionality of partial matching by considering the weight of terms. In this model, queries are expressed through boolean logic which includes "*AND*" and "*OR*" operators. For instance, the query "French" AND "presidential" AND "election" means that we want all these terms to appear in the retrieved documents.

Assume that for document d_j the weight of the terms is normalized and hence lie between 0 and 1. For example these weight can be computed using normalized TF-IDF as follows:

$$w_{i,j} = \frac{TF(w_i, d_j)}{max(TF_j)} \times \frac{IDF(w_i)}{max(IDF_j)} \tag{2.8}$$

Where $w_{i,j}$ is the weight of the i^{th} term in document d_j and $TF(w_i, d_j)$ is the frequency of term w_i in the document d_j and $IDF(w_i)$ is the inverse document frequency of term w_i.

To illustrate how document matching is carried out, let us consider a document composed of two terms w_1 and w_2. Then, the term assignment can be described by a two-dimensional term space, as shown in Figure 2.3.

For conjunctive queries (so-called AND queries), the (1,1) point represents the case where both terms occur in a document. This means that this point is the most interesting one. Hence, the complement of the distance between a document and this point is considered to measure the similarity between a document and the AND queries. Conversely, for disjunctive queries (so-called OR queries), the (0, 0) point represents the case

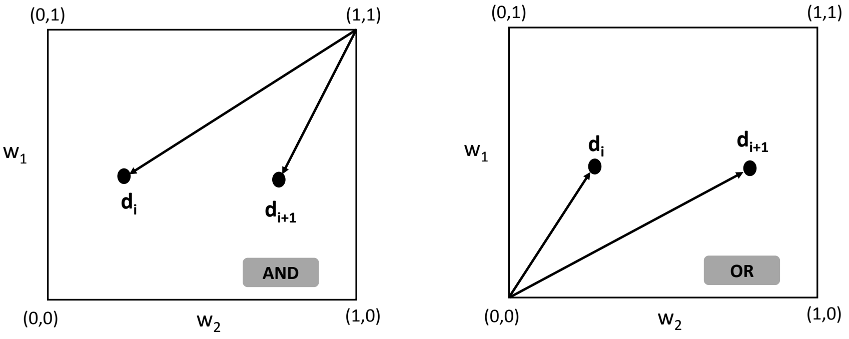


Figure 2.3: Extended Boolean logic: representation of AND and OR in the space composed of two terms w_1 and w_2.

where both terms are absent in a document. This means that the point (0, 0) is the least interesting one. Therefore, the distance from this point is used to measure the similarity of a document with respect to the OR queries.

In this model, the similarity is measured using the normalized Euclidean distance which can be generalized by using the p-norm distance to include p-distance. For clarity, we present here the relevance score based on Euclidean distances with 2-norms.

For a conjunctive query $Q_{and} = q_1 \wedge q_2 \wedge .... \wedge q_l$, the relevance score of document d is estimated as follows:

$$RSV(d, Q_{and}) = 1 - \sqrt{\frac{\sum_{q_i \in Q_{and}} (1 - w_{q_i,d})^2}{|Q_{and}|}} \tag{2.9}$$

Regarding a disjunctive query $Q_{or} = q_1 \vee q_2 \vee .... \vee q_l$, the relevance score of document D is computed as follows:

$$RSV(T, Q_{or}) = \sqrt{\frac{\sum_{q_i \in Q_{or}} (w_{q_i,d})^2}{|Q_{or}|}} \tag{2.10}$$

where $w_{q_i,d}$ is the weight of the query term q_i in the document d which is computed according to TF-IDF weighting scheme as described in Equation 2.8. $|Q_{and}|$ and $|Q_{or}|$ are the length of the conjunctive and disjunctive queries respectively.

To process a more general query, an AND/OR queries are combined by grouping the operator in a predefined order. For example, consider the query $Q = (q_1 \wedge q_2) \vee q_3)$. The relevance score of the document d is computed as follows:

$$RSV(d, Q) = (\frac{(1 - \sqrt{\frac{(1-w_{q_1,d})^2+(1-w_{q_2,d})^2}{2}})^2 + (w_{q_3,d})^2}{2})^{1/2} \tag{2.11}$$

Figure 2.4: Some of the most popular social media logos.

2.2 Information retrieval in social media

2.2.1 *Overview of social media*

Social media is defined as *websites and applications that enable users to create and share content or to participate in social networking*[2]. In this definition, we can distinguish three main aspects. The first aspect *websites and application* denotes the virtuality of the social media. The second aspect *create and share content* denotes the spreading of user-generated content (UGC) which refers to data published by users. Last but not least, the *social networking* aspect that denotes the social dimension and the connectivity of the users of social media.

According to Aichner and Jacob [2] social media websites and applications can be partitioned into eight categories: (1) blogs; (2) microblogs; (3) e-commerce portals; (4) multimedia sharing; (5) social networks; (6) review platforms; (7) social gaming; and (8) virtual worlds. Figure 2.4 depicts the logos of the most popular social media services such as Twitter, G+, Facebook and LinkedIn.

Forums, blog, and newsgroups such as Stackoverflow[3] are the oldest form of social media. LinkedIn is a professional networking site which is mainly based on densifying the social graph, while Google+ and Facebook combine the social graph with sharing content. These social media are more for networking people. Facebook is a multipurpose social networking platform, allowing users to chat, post photos and notes, and even play games. Twitter has a special characteristic as the social graph is bi-directional: user can follow another user without being befriended. This feature on Twitter allows users to get quick access to real-time information published by other users without any obligation to follow anyone. For this reason, Twitter is considered one of the most valuable sources of real-time information in addition to being a conversational social network [78].

Social networking is one of the most important features of social media. Wasserman and Faust [159] define a social network as "*a finite set or sets of actors and the relation or relations defined on them. The presence of relational information is a critical and defining feature of a social network*". According to Maslow's hierarchy of needs [100], humans need to

[2] https://en.oxforddictionaries.com/definition/social_media
[3] https://stackoverflow.com/

feel a sense of belonging and acceptance among their social communities. This primary need has driven the success of social media in recent years.

With social media, a user becomes an individual news media that not only consumes/absorbs information but also produces/propagates information about what is happening in the world or what is being said about an entity. The content generated by users (UGC) refers to different content published by users [20] as well as the interaction of users with the published resource[155]. Baeza-Yates and Ribeiro-Neto [20] define UGS as follows:

"User Generated Content is one of the main current trends in the Web. This trend has allowed all people that can access the Internet to publish content in different media, such as text (e.g. blogs), photos or video."

The content created and shared by users on social media can be categorized into three different groups as follows:

- Original or compiled materials that users make available over blogs, microblogs, wikis, and media sharing services (e.g.,YouTube[4] , Flikr[5]);

- Feedback and metadata such as comments, review, rates, and tags. Social media allows users to express whether they support, recommend or dislike a content through actions such as adding a like or retweeting/sharing a text, image, video or URL;

- Social network data including public profiles, social network structure, and user interactions.

2.2.2 *Social information retrieval*

In fact, the emergence of the social Web and the significant position that users have acquired in information producing and consuming processes have challenged traditional information retrieval approaches, that focus on document level regardless of the surrounding social context. Indeed, UGC in social media provides additional information than texts which refers to the social context of both users and information. Social information retrieval is an emerging research area that aims to search on UGC or to explore how UGC and social network data can be leveraged to enhance the performance of information retrieval systems. This new area brings together two areas of research, information retrieval and social networks analysis [74]. Social IR systems are characterized by the exploitation of UGC and social network in the information retrieval process. It considers the social network as well as implicit and explicit evidence of information interest such as tagging, rating, and friend activities in order to estimate the relevance of information [10]. The evaluation of the relevance of an information also takes into account the "importance" of its author in the social network. To do so, social network analysis methods are applied with the aim of identifying important users in a social network.

According to the purpose of the information retrieval task in the social media and the kind of UGC leveraged in the IR process, we can distinguish two different tasks of social information retrieval:

[4] http://www.youtube.com
[5] http://www.flickr.com/

1. The first task concerns seeking information in social media. In fact, UGC is a valuable source of real-time information that provides users with continuously updated information about developments of a topic of interest. In addition, UGC provides information that may not yet be published on the Web. Being aware of UGC availability and the original and fresh content that it may provide, users express a desire to access this type of data to fulfill their information need. This task covers for instance, information retrieval in microblogs [115], monitoring social media streams [86], experts search [96], opinion and sentiment retrieval [153] and conversation retrieval [98].

2. The second task focus on the exploitation of UGC and specific features of social network structure to enhance information retrieval. This is achieved by combining a query-based relevance score with a social-based relevance score in order to produce a final ranking of documents. These two factors can be combined either by an integrated (unified) approach or a modular approach [10]. In the integrated approach, relevance factors represent a transition probabilities on the social content graph and random walk algorithm such as PageRank [116] are used to rank retrieved documents. In the modular approach, the query-based and the social-based relevance scores are computed independently and then combined (e.g, linearly) to estimate the final relevance score [23]. Badache et al., [18] propose to use social signals such as (like, +1, share, tweet, comment) as sources of evidence to measure document prior probability of relevance. UGC is also considered as an information source for relevance feedback. To overcome the shortness of user queries, Koolen et al propose to expand it using Wikis [75].

2.2.3 *Social media vs traditional news media*

Social media as a source of real word event information is a double-edged sword [143]. Its low cost and easy access allow a rapid dissemination of information. It is an outstanding source of information that provides real-time news before traditional media which lead people to seek out and consume news from social media. For example, 62 percent of U.S. adults got news on social media in 2016[6]. The social media stream can be overwhelming with irrelevant and/or redundant information. However, this free and easy access leads the production of a content of different quality and even false. Indeed, social media is becoming the favorite support of ill-intentioned users to spread "fake news" that conveys news with intentionally false information[6].

The difference between social media such as Twitter and Facebook and traditional news media stems from the fact the former is generated in-situ by ordinary Web users. Posts are published by users which are interrupted by an event while they are going about their daily activities. Thanks to mobile devices, users can instantly report a real-world event. Indeed, in Twitter more than 80% of daily published tweets are posted from mobile devices [7]. Conversely, traditional news media provide professional content that is often created after the event occurs. Social media differs from traditional media in many aspects. Some of these aspects are advantages, others are disadvantages of social

[6] https://www.journalism.org/2016/05/26/news-use-acrosssocial-media-platforms-2016/

[7] https://expandedramblings.com/index.php/twitter-mobile-statistics/

media. We start by highlighting the main advantages of social media and afterwards we list some of their disadvantages compared to traditional media.

The main advantages of social media are as follows:

- **Coverage:** The user-generated content in social media covers a wide range of topics, from personal issues (i. e.about their daily activity) to public policy (i. e.related to a topic of interest to a wide audience). Although, tweets that concern personal issues represent more of the half of the tweets published by Twitter users [167], Twitter covers a similar range of public topic categories as traditional news media [166]. The comparative study conducted by Zhao et. al [166] between Twitter and traditional media reveals that Twitter covers more celebrities and brands that may not be covered in traditional media;

- **Freshness:** The strength of social media compared to traditional media is the immediacy of publication. Users publish valuable information that provides live coverage of scheduled (sports games) and unscheduled events (natural disaster). This specificity has raised social media as an important source of real-time information especially in the case of emergency events which include natural disasters such as earthquakes, cyclones, floods, fire, as well as man-made disasters such as terror attacks, or socio-political movements. In many cases, the most current news is provided by Twitter before traditional media [36]. For instance, in [47, 148] authors show that 75% of earthquakes can be detected by Twitter within two minutes just by monitoring tweets containing the word "earthquake" and other related words.

- **Timeliness:** Social media messages are streamed and posted with specific timestamps. They provide continuous updates and comments that allow tracking the evolution of topic over time. The dynamic nature of social media makes the text in social media quite different from the text in traditional collections which is more static. The statistical properties of social media text streams change over time because of topic and viewpoint drift.

- **Opinions:** Social media also provide means to users to express their opinions and hence hold a large number of opinionated content. Thus, identifying users'viewpoints on a specific issue and sentiment analysis become increasingly important for content analysis in social media [153].

In the following, we list some specific features of social media that can be considered as inconveniences. (inconveniences, in the sense that they challenge the automatic processing of the UGC).

- **Volume:** Sharing information on social networks has become common practice and even more a reflex. Where traditional media rely on a small number of contributors, each social media has a large number of users, each of them publishing messages more or less regularly. For instance, Twitter counts more than 330 million monthly active users worldwide. The reasons for these tremendous volumes in social media are inherent to the nature of these social media platforms: (i) it is easy and more timely to publish an information in social media; and (ii) it is easier to further share, comment on, and discuss the news with friends or other readers. The high volume of information published in social media while being informative, can also be overwhelming.

- **Velocity:** Content in social media is being created and shared at an unprecedented rate. This may be partly explained by the easy-to-use interactive interfaces, especially on mobile devices which allow posting messages at any time and anywhere. For instance, users on Twitter generate over 500 million tweets every day which corresponds to over 350,000 tweets sent per minute [8].

- **Variety:** The messages published on social media cover a variety of topics, ranging from ordinary everyday events to important events and/or global. Moreover, unlike traditional media, messages are not categorized or structured and topic drift is very common. In addition, people use different languages to discuss the same event.

- **Redundancy:** Social media are also characterized by the redundancy. Users do actively forward world event topics using, for example, the retweet mechanism in Twitter. This helps to spread news of important world events. But at the same time, this behavior yields a lot of redundancy. The same information might be reproduced by various users at different times.

- **Information quality:** In contrast to articles published by traditional new media, social media relies on users as primary contributors in generating and publishing content. Indeed, there is "in principle" no filtering on published content. This raises two main concerns regarding the quality of the information published in social media. The first one is related to the attention of the authors and the second one refers to the content quality (writing style). In fact, there is a particular kind of ill-intentioned users, so-called social spammers, who post spam contents in an automated way. For example, posting a tweet talking about "how to lose your weight in five days" under the "#Trump" topic. Another issue is the spread of fake news because of the fact that content in social media can be relayed among users with no significant third-party filtering, fact-checking, or editorial judgment. [6, 143]. For instance, US presidential election was characterized with a considerable amount of fake news [6] where most of them tended to favor Donald Trump over Hillary Clinton [9]. The second information quality issue in social media concerns the writing style that is characterized by:
  - Messages published by users on social media are commonly expressed in an informal way and are written in arbitrary style. Only some of them follow standard grammar requirements.
  - Messages are short, even sometimes very short. The length of messages is sometimes limited by the service.

Note that while identifying spam campaigns and individual spam accounts [158, 1] and fake new detection [143] are an important area of research, they are out of scope of this thesis which is about handling the volume and the redundancy in social media stream and overcoming the low content quality.

In this thesis, we focus on Twitter. The underlying motivation for this choice are: First, Twitter has gained increasing popularity since it was launched in 2006. As a result, a

[8] http://www.internetlivestats.com/twitter-statistics/

[9] https://www.buzzfeed.com/craigsilverman/viral-fake-election-news-outperformed-real-news-on-facebook?utm_term=.ayYvk2jAA#.kb3zE3K55

great volume of new tweets is being generated every second which provide continuously updated information about developments of interest to large audiences. It has become a popular communication tool for many journalists, politicians, and companies. Indeed, the study conducted by Hughes and Palen [8] reveals the important role of Twitter within the communication strategy adopted by Barack Obama during the 2008 presidential campaign in the United States. Donald Trump, the 45th President of the United States, seems to prefer tweeting to holding a press conference and uses Twitter to bypass traditional news media[10]. Second, Twitter, unlike most social media, allows free access to a large part of its data, which pushes many researchers to study it.

2.2.4 *Social media: Twitter*

2.2.4.1 *Overview of Twitter*

Twitter is a microblogging service launched in March 2006. Since then, it has become one of the most popular microblogging sites with more than 100 million daily active users[11]. The founders of Twitter defined it as a free social networking microblogging service that offers to everyone the opportunity to instantly create and share ideas and information, without any barriers [12][13]. Twitter allows users to post short messages known as "*tweets*". A tweet is a plain text in which the user can post photos, videos and web pages by adding the corresponding URL. Initially, a tweet was limited up to 140 characters. Since November 2017, this limit has been expanded to 280 characters. Twitter social network is based on the principle of followership. Twitter users follow others or are followed. Unlike most online social networking sites, such as Facebook, the relationship of following and being followed requires no reciprocation. A user can follow any other user, and the user being followed need not follow back. Being a follower on Twitter means that the user receives all the tweets from those the user follows. By default, tweets are publicly visible. Users can access other users' tweets in the social network unless no restriction is applied to their tweets.

A tweet can be addressed to a particular user by mentioning his *@username* at the beginning of the tweet. A user can interact with a tweet posted by another user in two different categories of actions. The first category refers to the actions that allow the user to show appreciation for a Tweet. If a user appreciates a tweet, then he has the possibility to show it by adding a like. Furthermore, Twitter allows saving tweets as favorite. The second category of actions consists of disseminating tweets (*retweet*) or *repelling* another user regarding one of his tweets as follows:

- **Retweeting**: A user can post a tweet of another user. This kind of tweets are known as "*retweet*" and this mechanism of information dissemination is similar to the "sharing" concept in other social networking services.

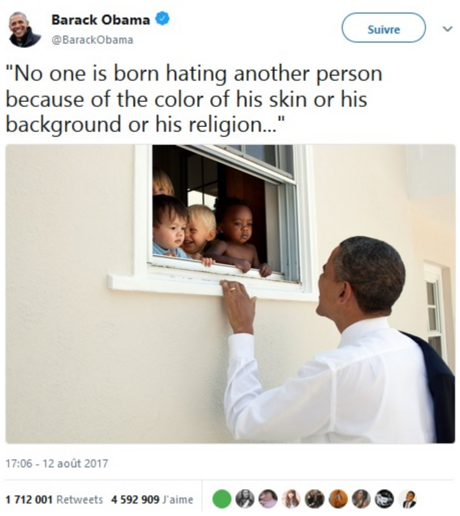
- **Replying:** A reply is a tweet points a previous tweet sent as a direct response to another twee. It is used to respond to another person's tweet by mentioning his *@username* at the beginning of the tweet before posting it.

[10] http://www.pbs.org/newshour/extra/daily-videos/trump-uses-twitter-to-bypass-media/

[11] https://www.omnicoreagency.com/twitter-statistics/

[12] https://about.twitter.com/fr.html

[13] https://en.wikipedia.org/wiki/Twitter#cite_note-Inc-31



Barack Obama
@BarackObama

"No one is born hating another person because of the color of his skin or his background or his religion..."

17:06 - 12 août 2017

1 712 001 Retweets 4 592 909 J'aime

Figure 2.5: The most retweeted tweet in Twitter: Tweet posted by Barak Obama.

Figure 2.5 is an example of a tweet posted by Barak Obama and represents, in fact, the most retweeted tweet in Twitter with over 1,712,001 retweets and more than 4.5 million likes.

In addition, users can annotate their tweets by using hashtags (a non-spacing word with prefix character "#"). A hashtag is used to draw attention to the main topic in the tweet. The use of hashtags helps to get found by a target audience because people research by searching for specific hashtags.

2.2.4.2 *Twitter data crawling*

Twitter allows access to streaming data that enables dynamically capturing the social activity of users. Twitter provides two types of APIs with different capabilities and limitations to access to the social activity of users namely: REST APIs and Streaming APIs. Below, we describe the difference between these types of APIs.

- REST APIs are based on the REST [14] architecture (REpresentational State Transfer) which is a web service that allows the requesting systems to access and manipulate textual representations of web resources. These APIs is used for getting access to historical data (old tweets for instances). To retrieve information a user must explicitly request it. For example, with this API, one can retrieve 3,200 of the most recent Tweets published by a user including retweets. Anonymous and free access to this API is limited to 180 requests per 15 minutes.

- Streaming APIs provide a continuous stream of public information from Twitter. The main advantage of these type of APIs is that once a request for information is made, the Streaming APIs provide a continuous stream of updates with no further input from the user. However, these APIs only grant access to a 1% sample of the Twitter data. At the time of writing this thesis, the main access points available for free are as follows:

[14] https://en.wikipedia.org/wiki/Representational_state_transfer

- **Public streams:** Returns a small random sample of all public tweets. This API provides approximately a 1% sample of all tweets (sometimes called the "spritzer");
- **User streams:** These are streams of public tweets published by single-user, with all a user'tweets.
- **Filter streams:** This access point allows to receive a sample of public tweets streams filtered by users, keywords, and location boxes. The free access to this endpoint allows specifying a maximum of 400 keywords, 5,000 userids, and 25 location boxes[15].

The public stream API is commonly used in literature to crawl Twitter data since it is the most versatile streaming API. In this thesis, we use public stream API to monitor tweet streams and to build the data collection used in our experiments.

In addition to tweet text, Twitter provides too many metadata in the tweet object that can be leveraged for different filtering purposes. Tweet object is provided in JSON format. The provided metadata on the tweet object is about the user who posted the tweet as well as the tweet content. The metadata about the user includes, for instance, the creation date of the user's account, user-name, screen-name and the number of followers and friends. The meta-data related to the content of tweet contains, for example, the publication time, the number of time the tweet has been retweeted, hashtags (if any), and URLs (if any).

2.2.5 *Information retrieval tasks in microblogs*

Information retrieval in social media needs to consider the specific features of social media documents and network structures [10, 23]. Based on these two aspects, the social networking and the creation of content, different research areas emerge. While social networking analysis is an important area of research, this thesis is about monitoring microblog stream in order to provide summaries that capture the development of an ongoing event over time.

The specificity of content generated by social media users has yielded new information retrieval tasks which correspond to new user's information needs. Information retrieval tasks in the content generated on social media can be divided into the following groups: microblog ad-hoc retrieval, opinion and sentiment retrieval and monitoring social media. In the following subsection, we present a brief description of these tasks.

2.2.5.1 *Microblog Ad-hoc retrieval*

Information retrieval within microblogs differs from Web search since the searched data differs in content and format as discussed in the previous section. The main advantage of searching for information over microblogs is the fact that it helps to find real-time information about the latest events. In contrast, it may take a certain time before this information becomes available on the Web and be indexed by search engines [43]. In this context, we can distinguish two different tasks based on the user's information need. The first one is a real-time search task, where the user wishes to see the most recent

[15] https://developer.twitter.com/en/docs/tweets/filter-realtime/overview

but relevant information to the query. This task has been tackled as tracks within TREC. In the TREC microblog track[115, 156], the task can be briefly described as follows: for a query submitted at time t, systems should provide a list of relevant microblogs ranked in reverse chronological order from newest to oldest, starting from the time the query was issued. Note that in this task the novelty between tweets was not considered. The second task is microblog search which is similar to traditional ad hoc information retrieval. In this task, relevant microblogs are ranked regardless of their freshness as required by real-time search.

Since the launch of the microblog track, several approaches have been proposed for microblog retrieval. They exploit three kinds of evidence, namely content features, social features and the time of publication. Many of them use temporal information related to microblogs [169, 9].

Regarding the relevance estimation based on the textual similarity between the microblog and the query, the main issue is the shortness of microblogs. Indeed, in microblogs terms do not appear more than once. This means that traditional IR models such as vector space model, which rely on terms frequency in documents and the length of documents, cannot be used in a straightforward manner. In order to overcome this issue, it has been suggested to not use the length normalization factor and term frequency [51, 3, 101]. In [51], authors have studied the impact of considering the term frequency and their normalization in the estimation of the relevance of microblogs using the BM25 model. The obtained results reveal that the use of these factors leads to a performance degradation. Based on this observation, Arifah et al. [3] adapt the vector space model by eliminating the length normalization factor. Massoudi et al. [101] propose a language model based on the occurrence of the term instead of its frequency in the microblog. Other works propose to enrich the content of microblog by considering the content of URL attached in the microblog [102]. Luo et al. [94] propose to use learning to rank method by considering features extracted from the metadata of tweets such as (retweet frequency, hashtags frequency, URL presence, is-reply). They show that considering these additional information enables significantly improving the performance. Empirical studies for microblog search show that relevance of tweets may depend on several features in addition to the content similarity to the query such as the number of followers and followings, the freshness of information, the presence of URLs and the user's location [109, 45].

As microblogging services provide additional information, other than text, such as social network, another category of approaches proposes to combine the content relevance with the social context of the microblog. This category of approaches considers that relevance is related to the credibility of the information source(the author of the microblog). The main criteria that reflect the importance of the users used in the literature are the number of tweets posted by the author [95], the number of times a user has been retweeted, the number of times the user has been mentioned by other users, the number of followers and friends. These criteria were simply linearly combined in [109, 165, 38]. In this context, learning to rank techniques have been widely used in tweet search in which social network and query-dependent features such as (cosine similarity between the query and the microblog) have been used as evidence of relevance. These learning approaches include linear regression [45] and RankSVM [35].

Another line of research attempts to use the importance of users in the social network to enhance the ranking quality of microblogs. The importance of users is captured by leveraging the relationship between users in the social network. In these approaches, the social network is modeled by a graph where nodes are users and edges are relationships that can be either directed or not. [162] propose TURank (Twitter User Rank), which is an algorithm for evaluating users'authority scores in Twitter based on link analysis. The authority of the user is estimated by using the PageRank algorithm [116]. Ben Jabeur et al. [23] propose a Bayesian network model that combines the social importance of microbloggers and the temporal magnitude of tweets. In particular, the importance of a user is assimilated to his influence on the social network. This property is evaluated by applying PageRank algorithm to the social network of retweets and mentions. The temporal magnitude of microblogs is estimated based on temporal neighbors that present similar query terms.

2.2.5.2 *Opinion and sentiment retrieval*

Social media is also a mean for users to express their opinions and debate on a wide range of issues, including society and politics. Users are seeking for such information to learn from similar experiences before making a decision (e.g., booking a hotel room). Opinion retrieval deals with finding relevant posts that express either a negative or positive opinion about some topic. During significant political events and campaigns, both citizens and politicians are increasingly relying on social media to disseminate information. The challenges of opinion retrieval approaches are to detect opinionated content and determinate associated sentiment (e.g., negative, neutral or positive sentiment) [112]. Opinion detection was studied on Twitter using supervised [118] and unsupervised methods [56]. Fang et al. [50] studied the individuals' voting intentions in the Scottish independence referendum held in September 2014 ("Yes": in favour of Independence vs. "No": Opposed). They propose a Topic-Based Naive Bayesian classifier that classifies the people's voting intentions based on the content of their tweets. To analyze opinionated content on Twitter, Liu et al.[89] propose to use a manually labeled data to train a language model and use the noisy emoticon data for smoothing.

Topic modeling for viewpoint discovery has been also applied to social media. In [124] authors introduced a time-aware topic model to summarize contrastive opinions based on sentiment (positive, negative, or neutral). Thonet et al. [153] propose Social Network Viewpoint Discovery Model (SNVDM), which is an unsupervised topic model to identify the issues topics and different users'viewpoints. This model leverages the users' social interactions in addition to the content generated by users for a given issue (e. g.U.S. policy). The intuition behind this proposition is being that users who connect together are more likely to share the same viewpoint. This intuition is shared by Fraisier et al. [55].

2.2.5.3 *Monitoring social media*

This group of tasks refers to a continuous systematic observation and analysis of social media. Because of social media features that we described in Section 2.2.3, monitoring social media is challenging. In recent years, there is growing interest in systems that address information extraction and exploitation issue on continuous document streams in social media. In this context, many tasks can be found, including topic detection and tracking, online reputation management, and event summarization. We present in what follows a brief description of these tasks.

Topic detection and tracking: The topic detection and Tracking (TDT) task was introduced to detect news event in traditional news streams. This task focuses on identifying the first document in a text document stream that corresponds to a previously unknown event [4]. Then, it is concerned with identifying all documents related to a particular event. To efficiently detect topics in textual streams, the majority of proposed approaches focus on bursts. The document stream is discretized in time windows and word frequencies are computed for each time windows. TDT concerns also novelty and redundancy detection task which is based on similarity/divergence measures such as the Manhattan, cosine similarities and language models. TDT in news streams is extensively studied in the literature and a more comprehensive literature review on this subject is provided in [4]

Topic detection and Tracking has been also applied to social media and particularly to microblogs[26, 160, 138]. Shamma et al. [138] propose a model based on uni-gram and in which the classical TF-IDF weighting is used to compute the importance of terms. All microblogs posted in the same time window are considered as "virtual document" and terms are ranked according to their TF-IDF weight. In [160] authors propose two-components based approach to detect bursty topics in real-time on Twitter. The first component maintains the occurrence of each bi-gram and tri-gram that occurs in the tweet stream. The second component used a sketch-based topic model to infer the bursty topics. Cheng et al. [36] suggests an alternative methodology for event detection in social media using space-time scan statistics. In this approach, tweets are clustered according to their space and time features, regardless of tweet content. For a detailed review of approaches that tackle this task, the reader can refer to [16, 62] which provide a complete survey of event detection on Twitter streams.

Online reputation management: Reputation management in social media [76] has been proposed in RepLab competitive evaluation campaign for Online Reputation Management Systems [12, 11]. It is aimed at developing systems for efficiently monitoring the reputation of entities (e. g.people, organizations, products, or services). This task is far from the one tackled in this thesis, we will not describe it furthermore.

Tweet summarization: Document summarization has been studied for years. Summarization approaches can be categorized as extractive and abstractive. The former selects sentences from the documents, while the latter may generate phrases and sentences that do not appear in the original documents. Both categories of approaches have been proposed to tackle microblog summarization. Most of the proposed approaches in the literature focus on selecting a list of meaningful tweets that are most representative with regard to a given topic. Recently, microblog summarization [85, 86, 87], have been tackled as tracks within TREC. In the next chapter, we provide a detailed review of related work on microblog summarization.

2.3 Conclusion

We presented in this chapter basic concepts of information retrieval and a brief overview of the main state-of-the-art retrieval models proposed for this aim. Moreover, we provided a comparison between tradition news media and social media in which we highlighted the main reasons that make straightforward adoption of the traditional IR models less effective. We focus on Twitter since it has gained increasing popularity in recent years. We presented the characteristic of this microblogging service and we introduced the different available approaches to crawl data generated by users on Twitter. Finally, we discussed the main information retrieval tasks over microblogs.

Among information retrieval tasks in microblog discussed at the end of this chapter, we are particularly interested in the microblog summarization task. This task exploits the user-generated content to provide a retrospective and a prospective summary that capture the key aspect of an ongoing event. The main aim is to shield the user from being overwhelmed. In the next two chapters, we will focus on this issue, particularly, we will discuss main approaches proposed in the literature for this application domain and the framework adopted to evaluate the performance of tweet summarization system.

3 TWEET SUMMARIZATION

3.1 Introduction

A text summarization system takes one or more documents as input and attempts to produce a concise summary that captures the most important information with a minimum of redundancy [111]. One of the first automatic text summarization was proposed by Luhn [93] in the 1950s, with a term frequency based strategy. Early work in text summarization focused on the single document summarization task where the input is only one document. Automatic text summarization has been gaining importance with the development of the World Wide Web. The enormous volume and redundancy on the web motivated research on multi-document summarization where the summary is generated from different documents about the same topic. Recently and with the emergence of Social Web and user-generated content (UGC) as a new continuous source of information, a considerable attention has been paid to automatic summarization of long-running events from social media streams such as Twitter.

Multi-document text summarization approaches can be categorized into two classes [111] extractive (selective) summarization and abstractive summarization [163, 123]. Extractive summarization consists of selecting of the most meaningful sentences from documents being summarized exactly as they appear in the original documents whereas abstractive summarization may generate sentences that do not appear in the original documents. Note that traditional document summarization is retrospective in nature. For further description of work on automatic summarization in the general case, the reader can refer to multiple state-of-the-art surveys such as the one conducted by Nenkova et al. [111] and by Yao et al. [163] which provide a recent progress made for document summarization within the last few years.

Tweet summarization aims at generating a digest for both long-ongoing or ended events from tweet streams in order to learn what is going on with regard to the topic, or what people think about the topic. This task is considered as an instance of multi-document summarization where each tweet is considered as a single document. Summarizing tweets streams is a challenging problem due to, on the one hand, the specificity of tweets and on the other hand to the volume, the velocity and the variety of the posts published in social media which is often highly redundant. Indeed, a long-running event may contain several unique information to summarize, conveyed by hundreds of tweets and spread-out over its lifetime. To be effective, such summaries are expected to fulfill some important requirements such as relevancy, low redundancy, coverage, and diversity.

This chapter introduces the prior work related to tweet summarization. We start with a description of tweet summarization in Section 3.2; we introduce the two different scenarios of tweet summarization namely retrospective and prospective tweet summarization followed by the description of the main challenges facing these tasks. Then, in Section 3.3 and Section 3.4 we survey background material on retrospective and prospective tweet summarization respectively. Because methods to evaluate relevancy

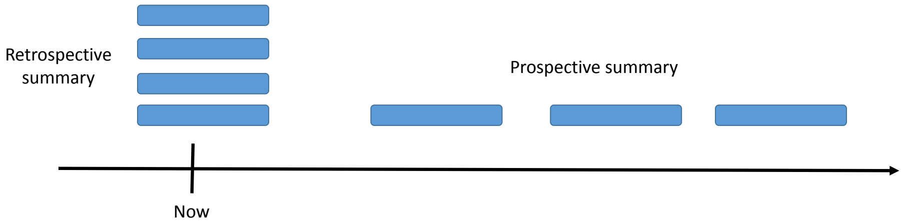


Figure 3.1: Illustration of retrospective and prospective summary models [120].

and novelty/redundancy are common in both tasks, we present in Section 3.5 and Section 3.6 the main approaches adopted in related work to estimate the relevancy and novelty/redundancy scores of an incoming tweet respectively. Finally, we provide in Section 3.7 a comparison of retrospective tweet summarization approaches as well as prospective tweet summarization.

3.2 Tweet summarization

Twitter is a valuable source of information to keep users up to date on topics they care about. However, timely following the development of long-running events is too difficult due to the velocity and the volume of the published information. Indeed, Tweets, in their raw form, while being informative, can also be overwhelming. For instance, more than 200,000 tweets were published during the 2018 EUROPA LEAGUE final match between Olympic Marseille and Atletico Madrid. Automatically generating a concise summary containing relevant and non-redundant posts that capture key aspects of information need, is one solution to keep users up to date. Tweet summarization provides means for effective information extraction and exploitation from social media.

As discussed in Chapter 1, when a significant event occurs, users may develop two complementary information needs. The first one is retrospective, it concerns what has happened up until now, while the second need is prospective and refers to the wish of the user to be kept up-to-date in a timely fashion whenever a new development occurs for the event of interest.

Based on these two complementary information needs, we can distinguish two different scenarios of tweet summarization namely, retrospective summarization and prospective notification (so-called real-time summarization). Figure 3.1 describes the task model of retrospective and prospective summary. These scenarios were introduced as two complementary tasks of information seeking on document streams [86, 120]. In retrospective tweet summarization, a set of the most relevant and non-redundant tweets that summarize "what happened" is periodically sent to the user (e. g.a daily email digest"). In prospective summarization, the system is expected to filter, in real-time, relevant tweets to the information need from a stream of tweets posted after the query time. It is required that updates are directly delivered to the user as notifications (e. g.to his mobile phone) as soon as a relevant and novel tweet is identified.

We describe in the following these two tasks and their challenges.

3.2.1 *Retrospective tweet summarization*

3.2.1.1 *Task description*

Retrospective tweet summarization is quite similar to ad hoc tweet retrieval. Consider a scenario where a user (e.g a journalist) just got breaking news about a political scandal that involves a presidential candidate. He turns to social media to find more details such as the major facts, reactions of people and whether the candidate can reach the finish line. In such a scenario, the user looks for a retrospective summary that captures what has occurred up until now to make up for what they would have missed. After that, it would be desirable to receive a periodic summary (e.g. daily) that highlights the important developments that occur recently (e.g last day).

The aim of retrospective tweet summarization is to fulfill this information need by providing a summary of an event or a topic that captures what has occurred up until now [120, 87]. This task consists of selecting a list of meaningful tweets or keywords that are most representative of the given topic. The summary should be concise and contain relevant and non-redundant tweets (avoid returning tweets that say the same thing) [87, 86, 85]. In addition to these requirements, the summary is expected to cover as many important aspects as possible of the event of interest. For example, a summary of a natural disaster should include aspects of what happened, when/where it happened, damages, rescue efforts, etc..., and these aspects are provided by different tweets. Also, it is desired that the summary includes information from different time periods to capture the development of the event over time. Optimizing all these criteria jointly is a challenging task especially for long-running events [142]. This is because the inclusion of relevant tweets relies not only on properties of tweets themselves but also on the properties of every other tweet in the summary.

Retrospective tweet summarization task has attracted a lot of attention in the last decade. This is because traditional document summarization approaches are less effective when handling tweets for the following reasons. On the one hand, the tweets are short, expressed in an informal way, and highly redundant. On the other hand, the tweet stream is characterized by the enormous volume of tweets that may arrive at an unpredictable rate. To promote the development of retrospective tweet summarization approaches many tracks were introduced at TREC such as Tweet timeline generation [84] track and scenario "B" in TREC real-time summarization [85, 86, 87].

3.2.1.2 *General framework of retrospective tweet summarieation*

Figure 3.2 shows the general framework of retrospective tweet summarization. The components displayed in a dashed line are considered as optional components. In this task, tweets are crawled, indexed and stored in real-time after prepossessing and trash filtering steps. Tweets published during a predefined period are treated on a batch. The inclusion of a tweet in the summary relies on its relevance score with regard to the topic of interest and its novelty/redundancy score with respect to previous tweets added on the summary. In addition to the tweet content, many approaches [90, 46, 123] have proposed to leverage social network features such as the importance of the author to compute the relevance of the tweets. The extraction of such features requires crawling social network information from Twitter. Query expansion techniques were investigated

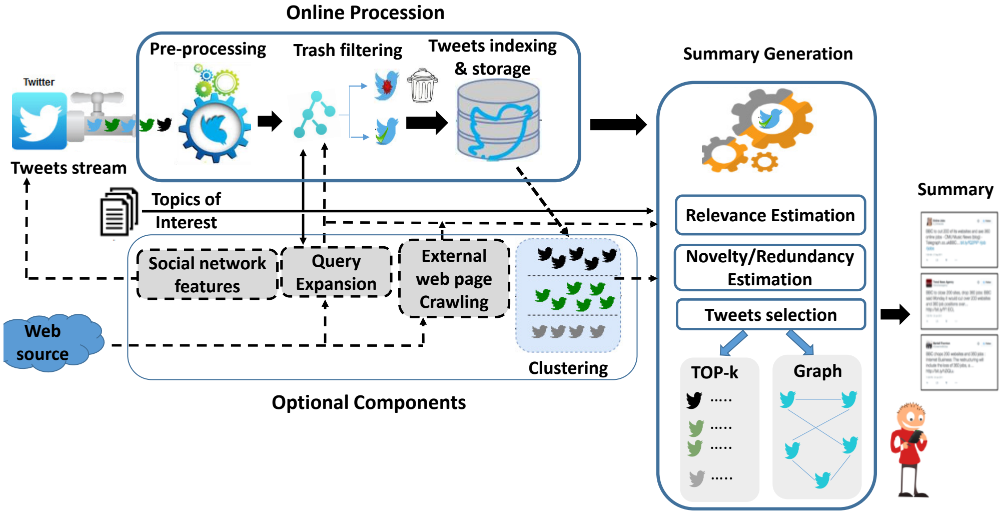


Figure 3.2: General framework of retrospective tweet summarieation .

to enrich the initial query [103, 170, 21]. Also, to overcome the shortness of the tweet, some approaches make use of the content of web pages linked from tweets [61, 59].

Regarding summary generation, most existing work are based on the extractive format, where the target is to extract salient tweets to construct a summary. This is achieved by iteratively selecting the top tweets with discarding the redundant ones [30, 141, 7]. Another used strategy is clustering with respect to the centroid of the tweet within a given set of tweets; this idea has been adapted by [113, 73, 142, 168]. Some authors use graph-based summarization to tackle the problem of the generation of tweet summary [114, 113, 90, 46, 63].

3.2.2 *Prospective tweet summarization*

3.2.2.1 *Task description*

Prospective tweet summarization is the reverse of ad-hoc tweet search, where a user provides an information need at a certain point in time, and the summarization system is expected to filter, in real-time, relevant and novel tweets from a stream of tweets posted after the query time. The goal of a prospective summarization (real-time tweet summarization) is to fulfill a prospective information need about an ongoing event. The prospective information need corresponds to the wish of the user to be updated in a timely fashion whenever a new development occurs for the event of interest [120]. In this task, as soon as an interesting update is identified, it is delivered in real time as notifications to users. The problem of real-time tweet summarization can be considered as an instance of secretary problem [17] which is described as hiring the best secretary out of n rankable applicants for a position. The applicants are interviewed one by one and the employee has to make an immediate decision after each interview. An applicant cannot be recalled once rejected. Prospective push notification system monitors and filter the live posts stream in order to identify relevant and novel content to be pushed to the user with respect to his information need. To shield users from unwanted notification, push notification should be relevant (on topic), novel (users should not be shown multi-

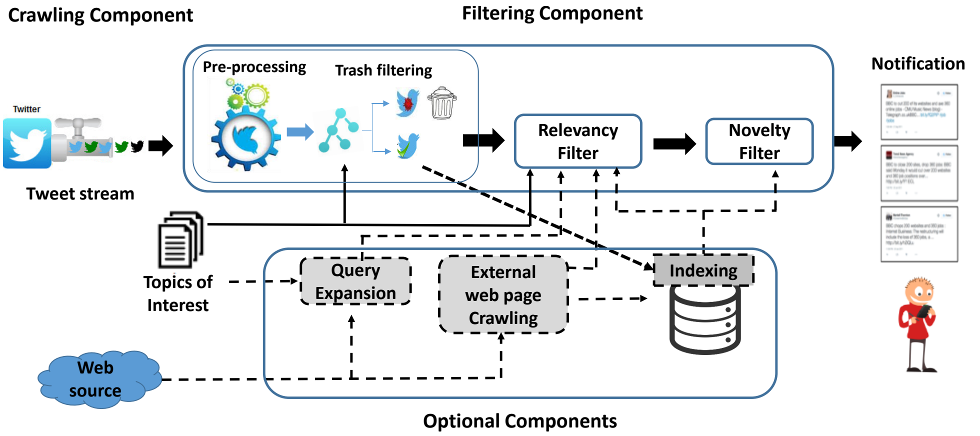


Figure 3.3: General framework of prospective notification system.

ple notifications that convey the same "thing"), and timely (provided as soon as possible) since information reported about an event can rapidly become outdated [120, 86, 91]. The problem of real-time event summarization can be defined as follows:

Given an event described by keywords and a stream S of timestamped tweets T, output a set R of representative tweets, such that:

1. $\forall T_i, \forall T_j \in S$ with the publication time t_i and t_j respectively $t_i < t_j$. It means that the two tweets are provided in chronological order.

2. $\forall T_i \in R$, $\Delta t = \tau_i - t_i$, is very low. where τ_i is a notification time (time of making decision to select tweet T_i).

3. $\forall T_i, \forall T_j \in R$, $T_i \nsim T_j$; it means that the two tweets T_i and T_j provide different information in order to keep the summary from being redundant and cover all sub-events (coverage);

4. $R \prec R'$, summary R is preferred to R' if R covers at least same sub-events than R' with less number of tweets (shortness properties).

3.2.2.2 *General framework for prospective tweet summarization*

Prospective tweet summarization systems have to monitor the continuous tweet stream and to deliver updates to the user whenever a new development occurs. Such system consist of two main components, namely tweets crawling and tweet filtering components. Other "optional" components can be considered depending on how tweets are represented (ie. vector of TF-IDF weight) and the model used to estimate the relevance and the novelty of an incoming tweet. Thus, a prospective notification system may include query expansion, tweet stream indexing and crawling web pages linked from tweet components. The overall description of the general framework is depicted in Figure 3.3 where the optional components are displayed in a dashed line.

The tweets crawling component is based on Twitter's streaming API. As was mentioned previously in Section 2.2.4.2, there are two different methods to get access to the tweet streams. The first one is user-centered stream that gives access to tweets published by a set of targeted users, whereas the second method (public stream API) gives access to a sample of 1% of all public tweets. Tweet summarization approaches rely on the public stream API because user-centered streams crawling has some drawbacks that include: (i) The number of users stream that can be crawled simultaneously is limited to 5000 users. (ii) The identification of relevant users to follow with respect to the topic of interest among the whole set of accounts is very challenging task [57].

The filtering component is commonly composed of three different filters adjusted sequentially. The two first filters (trash and relevance filters) select candidates tweets and the third filter reduces the redundancy. It evaluates the novelty score for only tweets that pass the two first filters. Before filtering, tweets are pre-processed. The preprocessing step consists of stop-words removal, tokenization and stemming. The purpose of the first filter is to discard trash tweets according to some criteria such as the length (number of tokens), the occurrence of a least one query term, the number of hashtags, number of URLs and the language[122, 61, 91]. In relevancy and novelty filters, a relevance and novelty/redundancy scores are computed. The decision to select/ignore an incoming tweet is commonly based on relevance and novelty threshold values.

To meet the prospective information need, notifications must be relevant, novel and timely [129, 86]. To fulfill the aforementioned requirements, a prospective notification system has to address the following issues:

- How to set the threshold values in which the decision is based? The appropriate selection of thresholds is critical to shield users from being overwhelmed with irrelevant and/or redundant tweets. The use of a high threshold value may lead to miss interesting tweets. Conversely, the use of a low threshold value may yield to overwhelm the user with irrelevant and/or redundant tweets. The threshold value can be static (the same empirically predefined value is used across all topics) or dynamic (value defined at the time of decision making).

- Which pushing strategy to adopt? After identifying the incoming tweet as being relevant and novel, the system may choose to push it immediately to the user or wait and see whether it is worth pushing. Delaying the submission of notification allows accumulating more evidence before making a decision. Meanwhile, in an ongoing event, developments may occur rapidly. Information has a lifetime (an expiry date) beyond which it may no longer be relevant. Hence, by delaying the submission, the system may miss the opportunity to push an interesting tweet at the appropriate time and push outdated information. That is why, in this task, a system has to find a trade-off between latency and the quality of notification in terms of relevancy and novelty.

3.2.3 *Challenges of tweets summarization*

Tweet summarization (either retrospective or prospective) faces many challenges that arise from the specificity of tweets such as the shortness and the writing style of tweets. In this context, relevancy depends on the social context of the tweet and its author (the user who published it) as well as the content similarity with the query. We argue that summarizing posts of social media such as tweets is substantially different from summarizing tradition documents for the following reasons:

- Tweets have been limited to 140 characters and since November 07, 2017 this limit has been expanded to 280 characters. Hence a post in Twitter contains only a handful of words which means that terms do not appear more than once. As a consequence, term frequency is less effective. Most term frequencies will be a small constant for a given tweet. Indeed, previous work has confirmed that document frequency (DF) and collection frequency(CF) are nearly identical in tweet search since terms almost always have term frequency (TF) equal to 1 in tweets [13];

- Tweets are noisy, ungrammatical, and may contain many abbreviations which pollute the text. Therefore, term mismatch issue is frequent [157], [39]. In addition, this makes it harder for standard Named-entity recognition (NER) methods to correctly detect entities in tweets [125].

- Redundancy in the tweet streams is pervasive since the same information is disseminated by different users at different times.

- To be effective, the system has to balance between selecting too many and selecting too few tweets [91]. In the latter, the user may miss important updates and in the former case, the user may be overwhelmed by irrelevant and/or redundant information.

In addition to the aforementioned challenges of summarizing tweets, prospective summarization faces the following issues that stem from the specificity of the prospective information need which requires processing the tweet stream in real-time:

- Unlike retrospective summarization where all documents (tweets) are available before taking a decision, in prospective (real-time) tweet summarization, the documents are not known in advance. Besides, A tweet cannot be recalled once rejected.

- The latency (timeliness) is one aspect to take into account particularly for topics that have a short lifespan. From the user's perspective, a notification is considered relevant if: (i) it conveys an interesting content with respect to the topic of interest, (ii) it is not redundant regarding what a user has previously seen and (iii) it is timely delivered. Hence, prospective tweet summarization systems have to trade-off between the quality of notification and the latency (delay between the notification and the publication time of a novel information). A system may choose to push new updates in real-time as soon as they are identified or may choose to delay their submission and see whether it is worth pushing. It is desired that the latency between notification and the publication time (the time a tweet appears in the social media) will be reduced as much as possible.

- Statistics about a term are not always available in particular at the beginning of monitoring the tweet stream, for instance, collection-based features such as (Inverse) Document frequency, $(I)DF$, and the average document length;

- The velocity of tweets and topic drift might impact collection-based statistics such as IDF which varies while new tweets arrive. These features of tweets stream raise

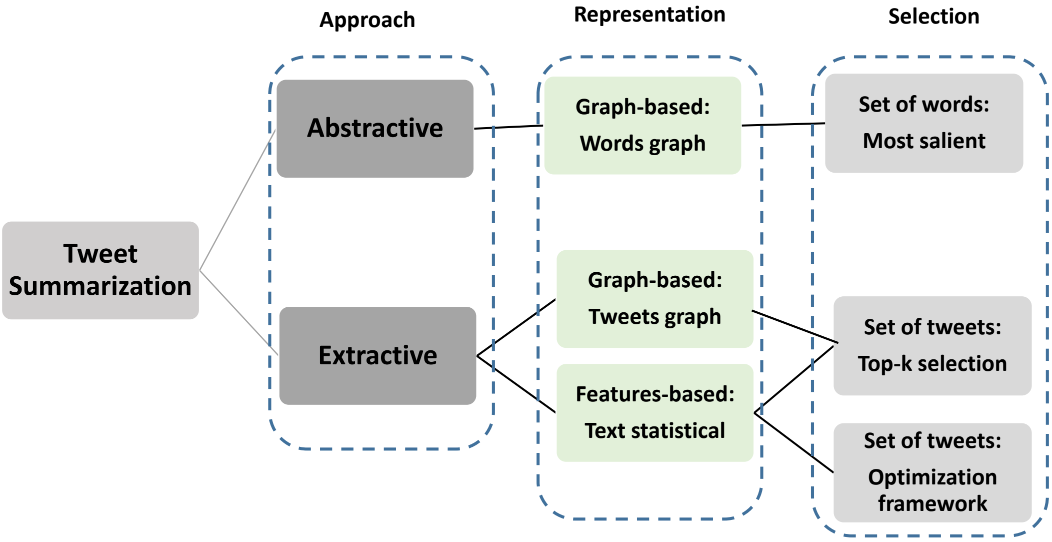

Approach
Representation
Selection
Tweet Summarization
Abstractive
Extractive
Graph-based: Words graph
Graph-based: Tweets graph
Features-based: Text statistical
Set of words: Most salient
Set of tweets: Top-k selection
Set of tweets: Optimization framework

Figure 3.4: A taxonomy of tweet summarization approaches.

the issue of indexing evolving events from tweet stream [27]. Updating collection-based statistics each time a new tweet arrives is a challenging task regarding the velocity of the tweet stream.

In the following sections, we will present related work that addresses the task of retrospective and prospective tweet summarization.

3.3 Related work on retrospective tweet summarization

Tweets (Microblogs) summarization can be considered as an instance of the more general problem of multi-document automatic summarization. Tweet summarization models define a framework for representing tweets collection and method that selects information nuggets (tweets or words) for inclusion in the summary with regard to a given event. Figure 3.4 presents a taxonomy of tweet summarization models based on the targeted kind of the summary (abstractive or extractive), how tweets are represented and how items (tweets or words) are selected for the inclusion in the summary. As we will see below, both abstractive and extractive approaches were investigated. Most of the abstractive techniques are graph-based approaches whereas two categories of extractive approaches can be distinguished, namely graph-based and feature-based. Regarding feature-based approaches and according to how tweets are selected for inclusion in the summary, we distinguish two groups of models. In the first group, candidates tweets are ranked according to their salient score and the TOP-k tweets are iteratively selected. In the second group, the summary generation is formulated as an optimization problem.

As in traditional document summarization, extractive approaches are predominant on tweet summarization [163]. This is due to the difficulty of abstractive summarization which usually requires advanced language generation and compression techniques[40]. Nevertheless, some work have attempted to propose an abstractive approach for tweet summarization [139, 71, 113]. These two categories of summarization approaches are discussed in more detail in what follows.

3.3.1 *Abstractive tweet summarization*

Abstractive summarization is difficult for tweet streams because of the noise and the variety of published content which can affect the quality of abstractive summary [114, 163]. The majority of the abstract approaches of the literature are based on word graphs where the nodes correspond to the terms that occur in tweets and edges represent their order of co-occurrence. The first abstractive tweet summarization approach was the Phrase Reinforcement (PR) algorithm proposed by Sharifi et al. [140] which extracts frequently used sequences of words. The algorithm builds up a word graph using the topic keywords as a root node and words of incoming tweets as nodes. The word graph used in this approach shows how words occur before and after the phrase in the root node. Each word node is weighted proportionally to its distance to the root and to its frequency. The summary sentence is selected as one of the highest weighted path. Note that Phrase Reinforcement algorithm generates a summary on offline for a given set of on-topic tweets.

In [113] authors showed that abstractive summarization is feasible in tweet stream if tweets are clustered based on similarity. They introduced the Multi-Sentence Compression (MSC) approach in which they combined event detection, text clustering, and text summarization. In this approach, a phrase summary is generated for each cluster of tweets. A directed word graph is built from the input tweets where nodes represent words and an edge between two nodes indicates the adjacency between words. Edges are weighted according to the frequency of occurrence of words. The phrase summary is built by selecting words that are given by the path between the start and the end nodes having the smallest average edge weight with a length greater than the minimum required. This approach has shown some limits for handling large datasets and datasets with a lot of unrelated tweets.

In [114] the same authors extended their original approach and introduced Twitter Online Word Graph Summarizer (TOWGS) which is considered as the first online abstractive summarization approach. TOWGS extend Multi-Sentence Compression (MSC)[113] approach by considering each node as bi-gram instead of a uni-gram in order to handle the noise generated by common words. To perform online summarization, the word graph is constantly updated when new tweets arrive. In order to control the size of the graph, old data (nodes and edges) that were not encountered in the previous time window are removed. The summary is built from the path that contains the highest weight node and maximizes a score function. The main disadvantage of this approach is that the use of tri-grams leads to a significant increase in the number of nodes.

3.3.2 *Extractive tweet summarization*

Extractive summarization consists of selecting a subset of relevant tweets with a minimum of redundancy that captures the main aspects of the event of interest. Unlike abstractive summarization where only graph-based approaches were proposed [140, 113, 114], in extractive summarization, two categories of approaches were proposed to measure the relevance of tweets namely graph-based and feature-based.

3.3.2.1 *Graph-based approaches*

In graph-based approaches, a tweet stream is modeled as a graph where a vertex denotes a tweet and an edge represents the similarity between tweets [46]. Liu et al. [90] combines the tweets content similarity and user social similarity based on features such as the number of followers and retweets when computing the weight of an edge. The summary is built from vertices that have the greatest salient score. In [46] tweets are clustered based on term frequency bursty and tweets in each cluster are ranked according to their salience score. A mutual reinforcement graph formed with three PageRank-like [116] graphs for words, tweets, and users are used to compute the salience score of tweets. In the proposed model, edges between tweets, words, and users represent tweets similarity, terms co-occurrence in the same tweet and users following-followee relationship respectively.

Inouye et al.[67] investigated two graph-based algorithms from traditional document summarization for tweet summarization, namely LexRank [48] and TextRank [105]. These methods exploit the relationships among tweets in addition to text statistical features. In LexRank, the weight of the edge represents the similarity between two tweets and the final score of a tweet is computed based on the weights of the edges that are connected to it. TextRank is based on the PageRank algorithm [116] and incorporates the whole complexity of the graph than just pairwise similarities as in LexRank. The final score of a tweet is recursively computed based on the weights of the edges that are related directly to it as well as the weights of the edges related to other tweets that are connected to the given tweet. A recent work proposes to make use of social-temporal context for summarization of tweets [63]. The proposed approach is based on the LexRank algorithm in which the weight of an edge is computed by combining tweets content similarity with the social context of the author as well as the temporal context of tweets. The social context is defined by the authority of the user in the social network and the popularity of the tweet (number of retweets). The temporal context is defined by the update rate of tweets for a given topic. To avoid redundancy in the summary in [63], the MMR (Maximal Marginal Relevance) [29] algorithm is used. In MMR, if a tweet is added to the summary, the other tweets are re-ranked according to the dissimilarity with the tweet summary. The similarity between two tweets is computed using cosine similarity. Xu et al. [161] proposes a Pagerank-like approach that leverages named entities, event phrases and their connections across tweets. Named entities and event phrases are represented by nodes and two nodes are connected by an edge if they co-occurred in k tweets. The weight of this edge is k.

The results obtained in the comparison conducted in [67] reveal that featured-based summarization approaches perform better than graph-based approaches. These results suggest that the added complexity of interrelationships in LexRank [48] and TextRank

[105] algorithms did not help in summarizing tweets. In the following subsection, we provide a review of feature-based approaches.

3.3.2.2 *Feature-based approaches*

Feature-based approaches are mostly based on text statistical such as term frequency [88] TF-IDF [30], HybridTF-IDF [141], Temporal TF-IDF [7] and language model [49, 131]. To rank a set of candidates tweet, some work suggest combining text statistical features and social features of users such as the number of followers as well as tweet features such as the number of retweets [21, 170, 108]. These approaches rely on tweet stream statistics. Alternative features based on the query terms occurrences in the text of tweet [91, 59] and in the web page linked from the tweet [61] have shown to be effective.

According to the method used to select tweets for inclusion in the summary, we can distinguish two different categories of feature-based approaches. In the first category, the top-k tweets are selected to form the summary where k denotes the desired limit length of the summary. This manner of building a summary is the most commonly used in the literature. In the second category, the generation of the summary is formulated as an optimization problem. In what follows, we review related work on these aforementioned categories of feature-based approaches.

Approaches based on TOP-K selection:

Sharifi et al [141] introduced a HybridTF-IDF approach where the TF component is calculated over the overall set of tweets (considered as one document). Top-weighted tweets are iteratively extracted with the exclusion of those having cosine similarity above an empirically predefined threshold with tweets of the current summary. The Sumbasic approach [110], initially proposed for document summarization, was reported to be effective as well for microblog summarization [97]. In this approach, the sentence that contains more frequent words in documents has a higher probability of being selected for summaries than the one with words occurring less frequently. However, these approaches are designed to retrospectively summarize an event given an on-topic relevant set of tweets and hence are not suitable to summarize an ongoing event over time.

The approach proposed in [171] is one of the first summarization approaches that monitor the live stream of tweets in real-time. This approach handles scheduled events such as sports games. It is based on term frequency in order to measure the relevance of tweets with respect to a given event and Kullback-Leibler divergence [77] to reduce redundancy. Shou et al. [142] proposed (Sumblr), a continuous cluster-based online tweet summarization approach that provides two types of summaries (online and historical). Tweets are clustered and those with the highest score in each cluster are selected for inclusion in the summary. However, In this approach authors presume the availability of a topic-related tweet stream.

Time-aware summarization has been studied by several authors to tackle automatic summarization of long-running events from tweet streams. Chakrabarti and Punera [30] split on-topic tweets into various periods as an event evolution map, and generate an update summarization result. Recently, Alsaedi et al, [7] investigated three different approaches to summarize real-world events: Temporal TF-IDF, retweet voting approach, and temporal centroid representation. The temporal TF-IDF approach, separate tweets into different time windows. The relevance of a given tweet is estimated using TF-IDF weighting scheme computed by considering only tweets published during the time windows that precedes the publication time of a given tweet. The retweet voting approach selects the most retweeted tweet in a time window as a representative tweet for this time window. In the temporal centroid method, tweets are clustered according to their time of publication and the centroid of each cluster is selected to form the summary. A cluster's centroid is the tweet with the highest similarity with other tweets.

The extractive approach is also usually modeled as topical modeling. [123] propose the Tweet Propagation Model (TPM), a time-aware user behavior model that tracks dynamic user's interest and topics. Topics are classified into three different classes (personal topics, common topics, or bursty topics). After inferring the probabilities of each tweet, top-ranked tweets for each time window are selected using an iterative algorithm to optimize coverage, novelty, and diversity of the summary. Chakrabarti et al. [30] propose a modified Hidden Markov Model which splits the corpus into events and each event is represented by a set of relevant tweets. This approach was applied in summarizing tweets related to sports events.

Based on scenario "B" (Section 4.2.1) of TREC RTS track (identifying a batch of up to 100 ranked tweets per day and per topic which corresponds to retrospective tweet summarization), a significant amount of approaches based on text statistics were proposed [49, 108, 59, 170, 61, 131]. Most of the proposed approaches attempt to generate summaries incrementally, by first selecting candidate relevant tweets, and then by discarding redundant one. Tweets are sorted according to their relevance score and the top-k tweet are iteratively selected with excluding those having a similarity score above an empirically predefined threshold with the current summary. Learning to rank model that combines several relevance scores were also investigated [108, 61] to re-rank candidates tweets before being returned to the user.

In [170], tweets are sorted according to a score that combines the social importance of tweet with its relevance score regarding the query. The social importance score is computed using a logistic regression model based on social attributes such as the number of followers and posts published by the user. The relevance score combines cosine similarity score based on TF-IDF and Okapi BM25 score. In [21] authors propose to expand query terms using the word embedding model. The relevance of the incoming tweet with respect to the extended query is based on Okapi BM25 model. Tweets are clustered incrementally using the Jaccard similarity in which the incoming tweet is assigned to the cluster containing the most similar tweet if the similarity falls above a certain threshold. The highest ranked tweet for each cluster is selected for the summary. The proposed approach in [170] was the high performing one in scenario "B" of TREC MB RTF-2015 track followed by the approach introduced in [21]

Haihui et. al, [59] propose to evaluate the relevance score by adding the number of occurrences of query terms in tweet text and in the web page linked from the tweet. In this approach, tweets are filtered according to the relevance and the redundancy predefined thresholds. The similarity between two tweets is determined by occurrences of their common vocabulary. The approach proposed in [122], first, ranks tweets according to their relevance score based on language model with Jelinek Mercer smoothing and then drops tweets that have a relevance score less than a predefined relevance threshold. For inclusion in the summary, the top-ranked tweets are selected by discarding those having overlap with any tweet that was previously selected higher than the empirically predefined threshold of 0.6. Notice that, the approach proposed in [59] is the best run in TREC RTS 2016 track and the run from [122] is ranked third among 40 runs.

In [61], authors investigated two different approaches. The first one relies on a language model to estimate the relevance score of incoming tweets with respect to the event of interest. The second approach is based on a learning to rank model based on the ListNet algorithm [28] to rank candidate tweets. In this approach, three different relevance scores were used as feature namely, scores based on a language model, a vector space model (cosine similarity) and the Jaccard similarity. Although this approach failed to beat the approach based on the language model, it achieves good performance since it falls in the third position. The approach based on a language model proposed in [61] was the best performing run in TREC RTS 2017 track. Some work have proposed to combine lineally several relevance scores. For example, [151] combines cosine similarity score based on IDF weight with negative KL-divergence language model score to compute the final relevance score of the incoming tweet with respect to the event of interest as well as the similarity score between two tweets.

Approaches based on optimization framework:

There is another line of studies which suggests building summaries using an optimization framework. In this context, Integer Linear Programming (ILP) combined with clustering techniques have been used in multi-document summarization [104, 81]. An ILP problem is a constrained optimization problem, where both the cost function and constraints are linear in a set of integer variables. This optimization problem is well-studied with efficient branch and bound algorithms for finding the optimal solution [64]. The selection of sentences is formulated as an optimization problem that is solved through a standard branch-and-bound algorithm [64] to provide an exact solution. In [81], authors proposed an event-aspect model based on LDA for sentence clustering that uses ILP for sentence selection. The optimization problem is based on sentence ranking information that selects one sentence which receives the highest possible ranking score from each aspect cluster subject to two other constraints related to redundancy and summary length.

For microblog summarization, a concept-based ILP formulation was proposed in [88]. This approach first extracts, for each topic, a set of n-grams that appear frequently in tweets related to a topic but do not appear frequently in a corpus. The extracted n-grams are considered as concepts. The summary is constructed by selecting a set of tweets that can cover as many important concepts as possible with the objective function sets to maximize the sum of the weight of concepts and constraints related to the length (number of tweets and words) and the coverage (number of concepts). Hiroya et al. [147] propose an adaptation of the tweet summarization model based on the budgeted median problem introduced in [146]. In this model, sentences (tweets) are selected so that every tweet in the given set of on-topic tweets can be represented by a tweet in the summary as much as possible. The summary is generated such that it maximizes the sum of the distances between the selected tweets. Constraints guarantee that the summary length is below the predefined limit and any tweet to which another tweet is assigned is selected in a summary. The distance between two tweets is computed by combining the content similarity and the temporal distance between their time of publication. The word overlap coefficient was used to compute the content similarity.

3.4 Related work on prospective tweet summarization

Although social event detection has been actively studied, how to efficiently monitor evolving events from continuous tweet streams remains an open and challenging problem. In the last few years, there has been a growing interest in this task. Guo et al. [58] introduced the temporal summarization task, whose goal is to generate concise update summaries about unexpected events, as they develop, from news sources (article, blog). This has been operationalized in the TREC Temporal Summarization (TS) track which runs from 2013 to 2015 [15, 14]. Recently, TREC organizers introduced in 2015 a new task dedicated to real-time summarization in the tweet stream. This task has been concretized in terms of mobile notification (so-called scenario A) in the TREC Microblog real-time filtering (RTF) 2015 [85] and the TREC real-time tweet summarization (RTS) [86, 87] tracks. The purpose of this track is to promote the development of approaches that monitor a tweet stream to keep the user up-to-date on topics of interest. A complete description of this track is presented in Section 4.2.1.

Note that despite similarities with document filtering [126] and topic detection and tracking (TDT) [4, 5] tracks, push notification differs by the fact that it focuses on identifying a small set of the most relevant updates to deliver to users [129]. In contrast, the document filtering task can be considered as a binary classification problem on every document in the collection with respect to the query and TDT focus also on identifying all documents related to the detected event. Furthermore, in the TREC Filtering and TDT tracks, systems must make online decisions as soon as documents arrive whereas, in our task, systems can choose to push older content. In the prospective notification task, the redundancy of a pair of tweets depends on their chronological order. If the chronologically later tweet contains information that is not present in the earlier tweet, the later tweet is considered novel; otherwise, the later tweet is considered as redundant.

In the majority of existing approaches, the decision of selecting or ignoring an incoming tweet is based on its relevance and redundancy scores. The relevance score is computed using query term occurrence in a tweet [91, 59, 149], stream statistics [49, 33, 122]. The novelty/redundancy of an incoming tweet with regard to the previous tweets selected in the summary is estimated using word overlap[91], a modified version of Jaccard similarity[145, 122, 144], KL-divergence[49] or cosine similarity [61].

In [49], the relevance score of tweets is evaluated by using the normalized KL-divergence distance, and the decision to select a tweet is based on a predefined threshold set manually. Precisely, the ranked list of tweets selected during the previous day is manually scanned from top to bottom, and the relevance score of the first irrelevant tweet is chosen as a threshold in the next day. In [149], the relevance score is based on the query term occurrence in the tweet. The relevance threshold is set dynamically according to the score of the tweets returned in the previous day. The TREC MB RTF-2015 official results showed that the runs PKUICSTRunA2 from [49] and UWaterlooATDK from [149] were the two best performing ones among 37 runs from 14 groups [85]. However, the approach proposed in [149] (the best performing automatic run in TREC RTF 2015) did not perform well and it failed to defeat the empty run in TREC RTS 2016 track. In [91], authors extended their previous approach [149] by using a daily feedback strategy to estimate the relevance threshold value. Results highlight the importance of proper threshold setting and show that the use of a simple feedback strategy improves the effectiveness of their approach proposed in [149].

In [59], authors propose a naive strategy to compute the relevance of a tweet. It is based on a query expansion and uses the content of the web pages linked from tweets. The relevance score is simply defined as the sum of the number of occurrences of terms of the expanded query in the tweet as well as in the web pages linked from tweets. The decision to select/ignore an incoming tweet is based on predefined relevance and redundancy thresholds values set dynamically according to pilot experiments. In the approach introduced in [122], the query and the tweet are represented as vectors using an IDF-based weighting scheme and the cosine similarity is used to compute the relevance score of the incoming tweet with respect to the query. The novelty is evaluated using the modified version of Jaccard similarity between the incoming tweet and each selected tweet. The decision to select an incoming tweet is based on predefined thresholds values. Notice that [59] and [122] were the two best performing run in TREC RTS 2016 track.

Query expansion techniques were investigated to enrich the initial query [108, 91, 170, 131], since relevant tweets may not contain all, or even any, of the query terms. in [91] pseudo-relevance feedback was used to expand the query. This was accomplished by querying Twitter search API with the initial query terms then the top-5 hashtags and top-10 terms are selected to expand the query. Suwaileh et al. [122] used Rocchio's pseudo-relevance feedback from two different sources to enrich the initial query. The first source is the top-2 terms extracted from the list of previous tweets that were assumed to be relevant. The second source is the Twitter search API. In [108] the query terms were expanded with the most similar terms using word-embeddings model [106]. Results in TREC 2015 and 2016 reveal that the query expansion is not effective since it may bring noise [122, 108].

Another line of research suggests combining more than one relevance scores using Learn to rank methods [28, 108]. In [61], authors use a learning to rank model based on the ListNet algorithm [28] to rank candidate tweets. In this approach, three different relevance scores were used as feature namely, scores based on a language model, a vector space model (cosine similarity) and the Jaccard similarity. In [108] authors propose to combine social features with query dependent features. The proposed approach is based on a learning-to-rank classifier with SVM^{rakn} model that combines some user, tweet-specific and query dependent features. Query depends features consists of the relevance scores computed using BM25, Jaccard similarity and a cosine similarity based on a doc2vec representation (where the document vector is the mean of the embedding vectors of its terms). Tweet specific features include the number of terms with and without stopwords, the ratio of the previous two features, count of characters in the stemmed tweet, count of URLs, count of hashtags, and count of user mentions. User feature is the log of the ratio of the number of followers to the number of friends.

We focus in what follows on main models used in the related work to estimate the relevance and novelty/redundancy scores of an incoming tweet. Notice that the approaches that we will describe below are the baselines against which we will compare our approaches. In Section 3.5 we detail how the relevance estimation baselines are implemented. Then, we describe novelty/redundancy estimation baselines in Section 3.6

3.5 Relevance estimation

The proposed approaches to compute the relevance score of an incoming tweet can be partitioned within two categories namely: stream-based and tweet-based models. The former models rely on collection based statistics (i. e.IDF, DF, the average length of tweets) whereas the former models use solely information related to a given tweet (i. e.the number of concurrences of query terms and the length of the tweet). Below, we list some models that were commonly used in the state-of-the-art.

3.5.1 *Stream-based models*

This category of models uses the collection based statistics to evaluate the relevance score of the given tweet with respect to a query. It includes traditional information retrieval models such as the vector space model, the probabilistic model, and language model. In these models, the frequency of terms in the tweet (TF) and document frequency (IDF) in the stream (which corresponds to the previously seen tweets) is used in this context to estimate the similarity between the tweet and the query. The literature on tweet summarization shows a variety of approaches that use vector space model with cosine similarity [114, 122, 170], language model in which tweet model is estimated using maximum likelihood estimation [61], language model with Kullback-Leibler (KL) divergence [77], Okapi BM25, TF-IDF and HybridTF-IDF[141].

In theses approaches, each tweet T^τ, with timestamp τ, is represented using bag-of-word approach, as $\{(t_1, w_1^\tau), (t_2, w_2^\tau), ..., (t_n, w_n^\tau)\}$, where t_i is the i^{th} term in tweet T^τ and w_i^τ is the corresponding weight computed with regard the previous seen tweets. It is supposed that when a new tweet T^τ arrives, the previous seen tweets are already indexed.

In BM25 [127], TF-IDF and HybridTF-IDF approaches the relevance score of tweet T^τ with respect to the query Q is defined as follows:

$$RSV(T, Q) = \frac{1}{norm(T^\tau)} \sum_{q \in Q} TF(q, T^\tau) \times IDF(q, T^\tau) \tag{3.1}$$

Where TF, IDF, $norm$ are the term frequency, inverse document frequency and normalization factors that are computed for each method as presented in Table 3.1.

Table 3.1: Components of TF-IDF, HybridTF-IDF and Okapi BM25 models.

Method	Term frequency	Inverse Document Frequency	Normalization
TF-IDF	$\#(w_i)$	$log(\frac{N+1}{df+0.5})$	$\lvert T \rvert$
HybridTF-IDF	$\frac{\#(w_i)\ InAllPosts}{\#WordInAllPosts}$	$log_2(\frac{N}{\#Tweet\ w_i\ Occurs})$	$\max[Minimum\ length, \lvert T \rvert]$
Okapi BM25	$\frac{(k_1+1)\times tf}{k_1\times(1-b+b\times\frac{\lvert T \rvert}{avgtl})+tf}$	$log(\frac{N-df+0.5}{df+0.5})$	$1 - b + b \times \frac{\lvert T \rvert}{avgtl}$

Note. N is the number of tweets in the stream and $avgtl$ is the average length of tweets, we set k1 to 1.2 and b to 0.75.

HybridTF-IDF [141] is a redefinition of TF-IDF in terms of hybrid documents. A tweet is considered as a single document when computing IDF. However, the entire collection of tweets is considered as a single document when computing the term frequency. This allows overcoming the shortness issue of tweets and yields to get differentiated term frequencies instead of a small constant for a given tweet.

In [61], the relevance score of the incoming tweet is computed by combing linearly the relevance scores of the tweet text and the web pages linked from tweets as follows:

$$RSV(T,Q) = \prod_{q_i \in Q} P(q_i|\theta_T) + \prod_{q_i \in Q} P(q_i|\theta_{URL}) \tag{3.2}$$

Where $P(q_i|\theta_T)$ and $P(q_i|\theta_{URL})$ are the probability that the query term q_i occurs in the language model of tweet T θ_T and in the language model of the web-page linked by URL mentioned in the tweet respectively. These probabilities are evaluated using Maximum Likelihood Estimation (MLE) and Dirichlet smoothing to overcome the issue of unseen terms as follows:

$$P(q_i|\theta_T) = \frac{f_T(q_i) + \mu \cdot P_{ML}(q_i|\theta_C)}{|T| + \mu} \tag{3.3}$$

$$P(q_i|\theta_{URL}) = \frac{f_{URL}(q_i) + \mu \cdot P_{ML}(q_i|\theta_C)}{|URL| + \mu} \tag{3.4}$$

Where $f_T(q_i)$ and $f_{URL}(q_i)$ are the number of occurrence of query term q_i in tweet and in the web page linked from the tweet respectively. $P_{ML}(q_i|\theta_C)$ is the probability that the query term q_i occurs in previously seen tweets. This probability is computed using Maximum Likelihood Estimation.

A negative KL-divergence language model was proposed in [151] to estimate the relevance of an incoming tweet as follows: ata

$$RSV(T,Q) = \sum_{q_i \in Q} P(q_i|\theta_Q) \times log\big((1-\lambda)P(q_i|\theta_T) + \lambda P(q_i|\theta_C)\big) \tag{3.5}$$

with: $\lambda = \frac{\mu}{|T|+\mu}$

3.5.2 *Tweet-based models*

In these models, the relevance of the incoming tweet with regard to the topic of interest is based on the number of occurrences of query terms in a tweet. The main advantage of these models is the no need to maintain terms statistics of incoming tweets. The scoring function proposed by Luchen et, al. [91] supposes that the user's information need includes a title of the information need and a description that indicates what is and what is not relevant. The relevance score of the incoming tweet is based on the query term occurrence with the occurrence of title terms having greater importance than the occurrence of expansion terms as follows [91]:

$$RSV(T,Q) = (3 \times |T \cap Q^t| + |T \cap Q^d|) \times \frac{|T \cap Q^t|}{|T|} \tag{3.6}$$

Where $|T \cap Q^t|$ and $|T \cap Q^d|$ are the number of occurrences of title and description terms in tweet T respectively.

Notice that the approach proposed by Luchen et, al [91] was the best official run in the TREC RTF 2015 track [85].

In [59], authors proposed to estimate the relevance score of the given tweet by simply summing the number of occurrences of query terms in the tweet text and in the web page text mentioned in the tweet. This approach was the best TREC 2016 run among 42 runs [86].

3.6 Novelty estimation

A tweet is considered novel if it conveys substantive information that is not present in the previous tweets. Thus, the novelty and redundancy are used interchangeably in opposite contexts. We notice that the notion of novelty/redundancy in tweet stream is not symmetric since tweets are aligned on a timeline. For a pair of tweets T_1 and T_2, if T_2 was posted after T_1 and it contains information that does not appear in T_1 then T_2 is considered novel. If tweet T_2 precedes tweet T_1 and T_1 contains similar information provided by tweet T_2, then T_1 is redundant with respect to T_2, but not the other way around.

Novelty detection is based on similarity/divergence measures such as the Manhattan distance, cosine similarities, and Kullback-Leibler (KL) divergence [77]. If the similarity (or distance) of the new incoming document is below (or above) than a certain threshold then this document is considered novel. According to the way that the similarity metric is used, two kinds of approaches can be distinguished, the document-to-document approaches and the document-to-summary approaches [72]. In the former, the new document is compared with all the previously seen documents while in the latter the incoming document is compared to the summary or to the centroid of clusters. The document-to-document approaches were shown to be more effective than document-to-summary approaches [72]. However, their main drawback is their computational complexity which is an important issue to consider when processing the tweet stream. For this reason, novelty detection in the task of tweet summarization relies on the document-to-summary comparison.

The similarity functions used in tweet summarization in the literature range from simple word overlap to vector space based model and language model. We present in what follows how these models are used in the novelty detection task.

Novelty detection based on vector space model:

Cosine similarity is proven to be effective in TDT task [53] and it is frequently used as a baseline in novelty detection. In tweet summarization, cosine similarity was widely used to estimate the similarity between tweets [63, 61, 123]. In such approach, tweets are represented as a bag-of-weighted-word. To evaluate the novelty score of an incoming tweet, its similarity to all tweets in the summary is computed. The maximum from these similarities is assigned as novelty score to the incoming tweet. Hence, the novelty score of the incoming tweet T with respect to the current summary $S = \{T_1, ..., T_n\}$ is computed as follows:

$$Sim(T,S) = maxCS(T,S) = \max_{T' \in S}[CS(T,T')] \tag{3.7}$$

Where $CS(T,T')$ is the cosine similarity among two tweets T and T' which is computed as follows:

$$CS(T,T') = \frac{\sum_{t_i \in T} w_{t_i,T} \times w_{t_i,T'}}{\sqrt{\sum_{t_i \in T}(w_{t_i,T})^2 \sum_{t_i \in T'}(w_{t_i,T'})^2}} \tag{3.8}$$

Where $w_{t_i,T}$ and $w_{t_i,T'}$ is the weight of the term t_i in the tweet T and T' respectively. The weight of term is computed using TF-IDF weighting schema as follows:

$$w_{t_i,T} = \frac{f(t_i)}{|T|} \times log(\frac{N+1}{df(t_i)+0.5}) \tag{3.9}$$

Where $(f(t_i))$ is the number of occurrences of the term t_i in tweet T, N is the number of the previous tweets already seen in the stream and $df(t_i)$ is the document frequency of the term t_i in the tweets stream at the moment the tweet T arrives.

Because novelty control uses divergence measurement, we compute the distance between two tweets using cosine similarity as follows:

$$Nov(T,S) = 1 - maxCS(T,S) \tag{3.10}$$

In this method, the new tweet with low maximum similarity to any of the tweets in the actual summary is considered as novel.

Novelty detection based on language model:

A comparative study conducted by Verheij et al [154], where several methods were evaluated for novelty detection, reveals that the best performing method was minimum Kullback-Leibler (KL) divergence [77]. This metric compute the distance between tow language models of tweets. The divergence score of the incoming tweet is computed as follows:

$$minKL(T,S) = \min_{1 \le i \le |S|} [KL(\theta_T, \theta_{T_i})] \tag{3.11}$$

Where $KL(\theta_T, \theta_{T_i})$ is the Kullback-Leibler divergence [77] of a tweet T given a tweet T_i which is computed as follows:

$$KL(\theta_T, \theta_{T_i}) = \sum_{t \in T} P(t|\theta_T) log \frac{P(t|\theta_T)}{P(t|\theta_{T_i})} \tag{3.12}$$

Where θ_T is the uni-gram language model on tweet T and $P(t|\theta_T)$ is the probability that the term t occurs in the tweet T. In order to avoid the problem of zero probabilities, Jelinek-Mercer smoothing is used. This smoothing linearly combines the tweet language model with the stream language model as follows:

$$\theta_T(t) = \lambda \times P(t|\theta_T) + (1-\lambda) \times P(t|\theta_C) \tag{3.13}$$

Where $\lambda \in [0,1]$ is the smoothing parameter. $P(t|\theta_C)$ and $P(t|\theta_c)$ are the probabilities that the term t occurs in the tweet T and the collection C (Tweet stream) respectively. These probabilities are computed using Maximum Likelihood Estimation (MLE). In [52], authors studied the impact of smoothing in language models for novelty detection and suggest to set the smoothing parameter λ to = 0.9.

Novelty detection based on word overlap: In this method, the similarity score between the incoming tweet T and a previous tweet T' is defined by the number of terms that occur in both tweets divided by the number of terms in the incoming tweet. Hence the novelty score of the incoming tweet T with respect to T' is computed as follows:

$$NS(T,SW) = 1 - \frac{|T \cap T'|}{|T|} \tag{3.14}$$

3.7 Synthesis of the state-of-the-art

3.7.1 *Retrospective summarization approaches*

We present in Table 3.2 a comparison between retrospective tweet summarization approaches from state-of-the-art. First, we provide a categorization of the different referenced approaches into three classes as previously discussed namely: abstractive approaches, extractive approaches based on graph and extractive methods based on features. Then we compare these approaches according to four criteria as follows:

- Online application: whether the approach can be used online to monitor tweet stream and to periodically generate an updated summary to be issued to the user at a predefined frequency;
- Time awareness: Whether the fact that information related to an event are spread-out over time is considered or not;
- the use of social features: Whether the social context of the tweet and its author (the user who published it) are taken into account when computing the relevance of tweets with regard a given event;
- The use of social network interaction: Whether the interaction on the social network is leveraged to evaluate the importance/authority of the user when evaluating the relevance of a tweet

The comparison of several tweet summarization approaches conducted in [67] has revealed that simple term frequency performs well for topic-sensitive microblog summarization because of the unstructured and shortness nature of tweets. Thereby, HybridTF-IDF was reported as the best summarization approach for tweets. Furthermore, the results obtained in this comparison reveal that featured-based summarization approaches perform better than graph-based approaches. These results suggest that the added complexity of tweet interrelationships in graph-based algorithms did not help in summarizing tweets. Ruifang et al [63] have shown that considering the social and the temporal context of tweets yields to enhance the performance of the graph-based approach.

Mackie et al. [97] compared eleven extractive summarization approaches based on textual features using four microblog data sets. The results indicate that SumBasic [110] and centroid-based summarisation with redundancy reduction [130] were the most effective.

Within TREC RTS track, a considerable number of models were proposed. All these methods are extractive approaches based on features such as TF-IDF and language model. The selection of tweets for inclusion in the summary is based on the TOP-K selection method. Table 3.3 provides a brief description of the best performing models in 2015,2016 and 2017 tracks. Furthermore, we compare these approaches based on whether the query expansion, the web pages linked from tweets and social feature are used or not.

The results obtained in TREC RTS tracks are consistent with previous results [97, 67] since approaches based on simple term frequency are among the high performing ones [149, 91, 59, 66].

Table 3.2: Comparison between retrospective tweet summarization approaches.

Reference	Abstractive	Extractive		Online Application	Time awareness	Use of social features	Use of social networking interactions
	Graph based	Graph based	Feature based				
PR [140]	•			✗	✗	✗	✗
MSC [113]	•			✗	✗	✗	✗
TOWGS[114]	•			✓	✗	✗	✗
LexRank [48]		•		✗	✗	✗	✗
TextRank [105]		•		✗	✗	✗	✗
Liu et al. [90]		•		✗	✗	✓	✓
Yajuan,et al. [46]		•		✗	✗	✗	✓
Ruifang et al. [63]		•		✗	✓	✓	✗
Ren et al.[123]		•		✗	✓	✗	✓
TF-IDF[30]			•	✓	✗	✗	✗
HybridTF-IDF [141]			•	✓	✗	✗	✗
Liu et al. [88]			•	✗	✗	✗	✗
SumBasic [110]			•	✓	✗	✗	✗
Centroid [130]			•	✓	✗	✗	✗
Xu et al. [161]			•	✗	✗	✗	✗
Zubiaga et al.[171]			•	✓	✓	✗	✗
Sumblr [142, 168]			•	✓	✓	✗	✗
Alsaedi,et al.[7]			•	✗	✓	✗	✗
Hiroya et al.[147]			•	✗	✓	✗	✗

To summarize, the majority of the feature-based approaches generate summaries by iteratively selecting the most relevant tweets and discarding those having their similarity with respect to the current summary above a certain threshold. Such approaches ignore the mutual relation among tweets. The estimation of relevance and novelty scores relies on stream statistics which may change when new tweets arrive. Regarding the graph-based approaches, while the mutual relation among messages is leveraged, these approaches are designed to retrospectively summarize an event given a set of on-topic tweets and/or mainly focus on ended events making them unsuitable to provide a summary of a long and ongoing event. Indeed, the temporal context is often not taken into consideration in state-of-the-art approaches.

To tackle retrospective tweet summarization, we introduce in Chapter 7 a novel approach for retrospective tweet summarization in which incoming tweets are filtered and clustered continuously and the summary is generated periodical using an Integer Linear Programming (ILP). The proposed approach falls within extraction approaches based on features. The relevance and the novelty scores of incoming tweets are computed without using stream statistics. The use of ILP allows optimizing simultaneously the different criteria required in the summary. To capture the different aspect of event spread-out over time, we take into account the temporal diversity of tweets as one criterion that needs to be fulfilled in the summary generation process.

Table 3.3: Comparison between the best-performing models in retrospective summarization scenario (so-called scenario "B") of TREC RTS track (2015,2016 and 2017).

Notation	Relevance estimation	Query expansion	URL	Social features	Redundancy	Track year	Rank
NUDTSNA,[170]	Combination of relevance score regarding the query and social quality score. Quality of tweet: logistic regression model based on social attributes. Query dependent score: Cosine similarity based on TFIDF weight combined with Okapi BM25	✓	✗	✓	Hamming distance based on simhah code of tweet	2015	1/42
CLIP [21]	Learning-to-rank (L2R) model based on SVM^{rank}: TFIDF,BM25, Jaccard similarity, doc2vec similarity and social features	✓	✗	✓	Jaccard similarity	2015	3/42
COMP2016 [59]	Classification model based on query dependent features and user features. Three classes: Highly relevant, relevant not-relevant.	✓	✓	✓	Character overlap	2016	1/40
	Query terms occurrences	✓	✓	✗	Character overlap	2016	5/40
QU [122]	Language model with Jelinek-Mercer smoothing	✓	✗	✗	Word overlap	2016	3/40
	Scoring function that combines language model with Jelinek-Mercer smoothing,and Dirichlit smoothing	✓	✗	✗	Word overlap	2016	4/40
HLJIT [61]	Language model with with Jelinek-Mercer smoothing	✓	✗	✗	Cosine similarity	2017	1/40
	Linear combination between cosine similarity based on IDF and negative KL-divergence language model	✓	✓	✗	Cosine similarity	2017	3/40
PKUICST,[151]	Linear combination between cosine similarity,based on IDF and negative KL-divergence language model	✓	✗	✗	Same model used to estimate the relevance.	2017	2/40
Udel [131]	Language model: Jelinek-Mercer smoothing	✓	✗	✗	Centroid-cluster comparison using Language model.	2017	4/40

3.7.2 *Prospective summarization approaches*

Table 3.4 provides a comparison of the best performing models that tackle prospective tweet summarization (push notification scenario) in TREC RTS track. Notice that. The comparison is based upon how the different methods handle the relevance and novelty evaluation, the setting of relevance threshold value, whether selected tweets are pushed immediately or delayed and whether user query is expanded, the web page linked from the tweet is used and social features are considered or not. Interestingly, we observe that run [149, 91]that achieves the best performance in TREC 2015 track did not perform well in TREC 2016 track. The same trend is also noted in TREC 2016 and 2017 tracks. The best performing method in TREC 2016 track [122] fell in the eleventh position in TREC 2017. This suggests that real-time tweet summarization is still an unsolved problem.

The analysis of these approaches reveals the following limitations in the state-of-the-art approaches:

- With a few exceptions, the common approach to compute the relevance of an incoming tweet with respect to the topic of interest relies on text statistical features such as TF, IDF, and language model. These widely-adopted approaches, however, have several drawbacks that stem from the shortness, the context dependency nature of a tweet and the streaming character of the collection.

- The novelty detection is based upon a pairwise similarity/divergence measures such as cosine similarity in which the incoming tweet is compared to all tweets previously pushed to the user;

- The relevance filter is threshold-based. Thus, the effectiveness of the tweet filtering relies on identifying an appropriate threshold for pushing updates [91, 33]. Several strategies have been proposed to set the threshold value which range from, static value overall topic and dynamic value set using relevance feedback. [91, 49, 33, 122].

- The majority of state-of-the-art approaches consider only query dependent features to measure tweet relevance, which include features corresponding to particular statistics of query terms such as term frequency, and term distribution in the stream. Despite the fact that it is recognized that social signals are important for relevance, it is unclear how effective is the use of social signals in tweet filtering;

- The majority of existing approaches tend to trade latency for a high quality of (relevance and novelty).

To overcome these shortcomings, our contribution is at different levels as follows:

- First, *relevance estimation*: we introduce, in Chapter 5 Word Similarity Extended Boolean Model (WSEBM), a relevance model that does not rely on stream statistics when computing the relevance score of the incoming tweet by taking advantage of word embedding model (word2vec). By doing this, we overcome the shortness and word mismatch issues in tweets;

- Second, *novelty detection*: Instead of using the pairwise comparison to compute the novelty score of the incoming tweet against tweets previously seen by the user, we propose to compute the novelty score regarding all words of tweets already pushed to the user. The proposed novelty detection method scales better and reduces the execution time, which fits real-time tweet filtering.

- Third, *relevance filtering*: To overcome the issue of relevance threshold setting, we propose, in Chapter 6 an adaptive Learning to Filter approach based on supervised machine learning algorithm to build a binary classifier that predicts the relevance of the incoming tweets.

- To enhance the effectiveness of the classifier to identify correctly relevant tweets, we leverage social signals as well as query-dependent features. To fit the real-time filtering scenario, we define a set of social features based solely on data provided on the meta-data of tweets. This allows avoiding the crawling of further information from Twitter servers which can be costly in terms of time.

- Finally, *latency between push notification and publication time*: we adopt a real-time push strategy and we show that the proposed approach achieves a promising performance in terms of notification quality (relevance and novelty) with low cost of latency;

Table 3.4: Comparison between the best-performing models in push notification scenario (so-called scenario "A") of TREC RTS track (2015,2016 and 2017).

Notation	Relevance	Relevance Threshold	Novelty	Query expansion	URL	Social features	Pushing strategy	Track Year	Rank
Waterloo[149, 91]	Query term occurrence	Adaptative	Word overlap	✓	✗	✗	Immediate	2015	2/37
								2016	22/42
PKUICST[49]	Language model: KL-divergeance with Dirichlet and Jelinek-Mercer smoothing	Adaptative: set manually	Same model used to estimate the relevance.	✗	✗	✗	Immediate	2015	1/37
NUDTSNA[170]	Quality of tweet: logistic regression model based on social attributes. Query dependent score: Cosine similarity based on TFIDF weight combined with Okapi BM25	Adaptative	Hamming distance based on simhah code of tweet.	✓	✗	✓	Immediate	2015	3/37
COMP2016 [59]	Query terms occurrences	Predefined	Character overlap	✓	✓	✗	Immediate	2016	1/42
QU [145, 122, 144]	Cosine similarity Based on IDF component	Predefined	modified version of of Jaccard similarity	✗	✗	✗	Delayed	2015	5/37
								2016	2/42
		Dynamic: live relevance feedback	modified version of of Jaccard similarity	✓	✗	✗	Delayed	2017	11/41
HLJIT [61]	Learning to rank: ListNet algorithm Combines relevance scores based on Jaccard similarity, cosine similarity and Language model with Dirichlet smoothing	Predefined	Cosine similarity:Push only one tweet per topic per day	✗	✓	✗	Delayed	2017	1/41
	Language model with Dirichlet smoothing	Predefined	Cosine similarity: Push only one tweet per topic per day	✗	✓	✗	Delayed	2017	2/41
IRIT [66]	Word overlap	Predefined	Push only the first tweet per topic per day	✗	✗	✗	Immediate	2017	4/41
PKUICST [151]	Linear combination between cosine similarity based on IDF and negative KL-divergence language model	Predefined	Same model used to estimate the relevance.	✓	✗	✗	Immediate	2017	6/41

3.8 Conclusion

Summarizing events using information obtained from social media sources such as Twitter has received a lot of attention from the research community in recent years. This can be explained by the ability of such sources to provide up-to-date information about ongoing events as they evolve. We presented in this chapter a review of the main approaches for tweet summarization. We distinguish two different scenarios that aim to fulfill two complementary information needs namely retrospective and prospective summarization. In particular, we describe each task and the challenges that face them. The review of the related work shows that with few exceptions, current systems either directly apply, or build upon, classical summarization approaches previously shown to be effective within the traditional document news stream. A comparative analysis of the most adopted methods, based upon how these methods select tweets for inclusion into the summary, is done to understand the main strengths and limitations of them. Finally, we highlight the main limitation of the reported state-of-the-art approaches.

In accordance with the two tasks discussed in this chapter, we will propose in Chapter 5 a model for real-time tweet stream filtering which does not rely on text stream features and overcomes the issue of word mismatch. Following that, we will present in Chapter 6 a model based on a supervised learning approach. The proposed model has two-fold advantages: it overcomes the issue of relevance threshold setting and it leverages the social signals as well as the query dependent features for identifying relevant tweets.

In Chapter 7, we will present a model based on an optimization framework to generate a retrospective summary that considers the mutual relationship between tweets as well as the temporal context of tweets.

Before that, we present in the next chapter the framework introduced within TREC tracks to evaluate the performance of retrospective tweet summarization as well as prospective notification systems.

4 BENCHMARK DATASETS FOR REAL-TIME TWEET SUMMARIZATION

4.1 Introduction

Our experimental setups are grounded in the real-time summarization track at TREC. In this chapter, we present the framework adopted in these tracks to evaluate the performances of tweet summarization approaches. The retrospective summaries are evaluated based on batch evaluation methodology. The prospective summarization was evaluated following two different methodologies, namely the online in-situ evaluation and the batch evaluation. The batch evaluation was performed after the end of the evaluation period via polling. The online evaluation, in contrast to batch evaluations, was performed during the evaluation period while systems pushed tweets. This evaluation follows interleaved evaluation framework introduced by Qian et al [120]. The studies conducted in [120, 129] show that the online in-situ and batch evaluations are correlated.

In what follows, we present TREC real-time summarization task. Then, we describe the two methodologies adopted to evaluate prospective summarization and the metrics used in each one. Afterwards, we discuss the precautions that need to be taken into consideration when reusing the data collection of these tracks in a replay mechanism. Finally, we present the evaluation metric used in the retrospective summarization task.

4.2 TREC microblog real-time filtering and summarization tracks

4.2.1 *Tasks Description*

The TREC Microblog (MB) real-time filtering and summarization tracks [85, 86, 87] is an evaluation campaign organized annually by NIST[1] since 2015. Until now three iterations were organized, MB Real-Time Filtering (MB RTF) 2015 track [85] and Real-Time Summarization (RTS) 2016 and 2017 tracks [86, 87]. It is planned that RTS track will be pursued in 2018. The aim of this track is to explore prospective and retrospective information needs over document streams containing novel and evolving information. In this track, participant systems are required to monitor the live stream provided by Twitter streaming API over a period of many days (defined by the organizers) and to identify relevant tweets per day with respect to predefined user interest. Tweets identified as relevant and novel to the user's interest profile are pushed in two different ways operationalized in terms of two scenarios:

1. **Scenario A**, called "*Push notifications*": In this scenario, a tweet that is identified as relevant and novel is pushed in real-time to the user as a notification on his mobile device. Participating system are allowed to push up to 10 notifications per

[1] https://trec.nist.gov/

day per interest profile and pushed tweets are routed immediately to the mobile phones of assessors for relevance judgment.

2. **Scenario B**, called "*Periodic email digest*": This scenario is more like a top-k (k=100 in this track) ad-hoc retrieval task based on a one-day tweets collection. It consists of identifying a batch of up to 100 ranked tweets per day and per topic to be pushed as an email notification to the user after the day ends.

Notice here that Real-Time Summarization track is the result of the merger of the Microblog (MB) Track, which ran from 2010 to 2015, and the Temporal Summarization (TS) Track, which ran from 2013 to 2015 [15]. The main differences from these previous TREC evaluation tracks are:

- As opposed to traditional TREC tracks, no collection was distributed ahead of the evaluation period. In these tracks, data collection was generated by each participant independently by crawling tweets using Twitter's streaming API during the evaluation period.
- The participants have to maintain a running system that continuously monitors the tweet stream during the evaluation period. Although this requirement demands additional software engineering effort, it did not exclude the participation of a considerable number of teams worldwide as shown in Table 4.1.
- Participants in RTF and RTS tracks were required to process tweets posted in real time whereas Temporal Summarization, the streaming nature of the document collection were simulated.
- Temporal summarization task deals with news articles and blog posts crawled from the web whereas RTS track is designed to generate concise update summaries from tweet streams.

The difference between RTF 2015 and TREC RTS 2016 tracks lies only on the framework used to evaluate system output. In the former, participant runs were evaluated on the basis of batch evaluation after the end of the evaluation period while in the latter, runs were evaluated using two methodologies namely batch and online in-situ evaluation. In the in-situ evaluation, notifications were evaluated in an online manner by mobile assessors during the evaluation period while system submitted notifications. Tweets pushed by participant systems were routed directly to the mobile phones of assessors as notification.

Comparing TREC RTS 2017 track with previous editions RTF 2015 and RTS 2016, the major change consists in the deployment of a mechanism whereby participant systems can fetch mobile assessor relevance judgments in real time for its pushed tweets. The availability of relevance feedback provides opportunities for techniques based on adaptive learning and relevance feedback.

4.2.2 *Data collection*

The corpus of this track includes a collection of tweets (which were under the responsibility of participants), topics (interest profile) and relevance judgments provided by track organizer. Table 4.1 provides some statistics about the corpus of these tracks.

Table 4.1: Statistics of TREC RTF 2015 and RTS 2016 and 2017 tracks.

				Tweets Judged		Scenario A		Scenario B	
Year	**Period**	**Topics**	**Nb Tweets**	**Batch Evaluation**	**In-situ Evaluation**	**#T**	**#R**	**#T**	**#R**
2015	20/07/2015 29/07/2015	51	40.242.516	94066	-	14	37	16	42
2016	02/08/2016 11/08/2016	56	36.908.568	67525	12.115	19	42	15	40
2017	29/07/2017 05/08/2017	97	29.255621	94.307	50.124	15	41	15	40

Note. #T and #R indicate the number of participating teams and submitted runs respectively.

Tweets collection were generated by each participant independently during the evaluation period. Tweets were crawled using Twitter's streaming API during the predefined period. Through this streaming API, clients can obtain a sample (approximately 1%) of public tweets, which offers a live feed of tweets to be downloaded free of charge available to anyone who signs up for an account. It is important to note that multiple listeners to the public Twitter sample stream receive the same tweets. This was confirmed in the study conducted by [69] which reveals that tweets set crawled over a three day by six independent systems have Jaccard overlap of 0.999.

Topics (interest profile) provided by the track organizer following the "standard" TREC topic format that includes a "*title*", "*description*", and "*narrative*". The "*title*" consists of few keywords that provide the essence of the information need. The "*description*" and "*narrative*" indicate what is and what is not relevant. The following example presents an interest profile extracted from TREC RTS 2017 track.

- *"topicid":"RTS54"*
- *"title":"North Korea missile test"*
- *"description":"Find statements made by North Korea about its missile tests."*
- *"narrative":"The user is trying to understand North Korea, and wants to know what anybody in that country is saying about its missile tests. Statements by President Kim Jong Un, members of the North Korean military, or any other North Korean are all relevant."*

In the following section, we present the evaluation metrics adopted in this track to assess the performance of real-time push notification approaches which corresponds to scenario A. Then, we discuss the reusability of these corpora using a reply mechanism. The evaluation metrics of scenario B will be present in Section 4.4.

4.3 Prospective summarization evaluation

The prospective tweet summarization approaches were evaluated according to two methodologies, namely the traditional batch evaluation methodology based on pooling and the in-situ evaluation methodology conducted in an online manner. In the in-situ evaluation, the assessor received the notification submitted by the participant systems as soon as they were generated. The assessments happened online as systems generated output during the evaluation period whereas batch evaluation was performed once the challenge ends. Studies conducted in [129, 120] show that results of the online in-situ evaluation are correlated with the results of a more traditional post-hoc batch evaluation. Notice that in the TREC RTF 2015 push notifications have been assessed with only a post-hoc batch evaluation methodology.

In the following subsection, we describe the batch evaluation as well as the different adopted metrics in this methodology. Afterward, we present the in-situ evaluation methodology and its metrics.

4.3.1 *Batch evaluation*

This evaluation occurred after the live evaluation period ended and it was performed in two stages: relevance assessment and semantic clustering. First, tweets returned by participating systems were judged for relevance by NIST assessors via pooling. Then, relevant tweets that share substantively similar content were grouped into a semantic cluster. Tweets belonging to the same cluster are considered redundant. The quality of notification (relevancy and novelty) are evaluated on in terms of recall and precision which are captured using a gain-oriented metrics. To penalize systems for returning redundant information, only the first tweet from each cluster receives a gain, all other tweets from the same cluster are automatically considered as not relevant. The gain of each tweet is set as follows:

- irrelevant tweets receive a gain of 0;
- relevant tweets receive a gain of 0.5;
- highly relevant tweets receive a gain of 1.0.

The precision is assessed using Expected gain metric (EG) while the recall is evaluated using normalized cumulative gain (nCG). The timeliness of notification is captured by computing the latency between the notification and the creation time of a tweet. In the following subsections, we present the adopted metrics.

4.3.1.1 *Precision oriented metric*

In TREC RTF 2015 track, a single metric, namely expected latency-discounted gain (ELG), that attempted to incorporate both relevance, novelty, and timeliness were adopted [85]. This metric considers both the relevance and the time at which tweets were pushed as follows:

$$ELG(S) = \frac{1}{N} \times \sum_{T \in S} G(T) \times max(0, (100 - delay)/100) \tag{4.1}$$

where S is the generated summary, N is the number of returned tweets. The delay is the latency (in minutes) between the tweet creation time and the time the system decides to push it. G(T) is the gain of each tweet.

In TREC RTS 2016 and 2017, instead of using a single-point metric for precision that tries to combine relevance, novelty and the latency, the organizers decided to separately capture output quality (relevance and redundancy) and timeliness (latency). Hence, the precision of notification is assessed using the Expected gain (EG) metric which is defined as follows:

$$EG(S) = \frac{1}{N} \times \sum_{T \in S} G(T) \tag{4.2}$$

4.3.1.2 *Recall oriented metric*

The recall is captured using the normalized cumulative gain (nCG) which is defined as follows:

$$nCG(S) = \frac{1}{Z} \times \sum_{T \in S} G(T) \tag{4.3}$$

where S is the generated summary, Z is the maximum possible gain (given the 10 tweet per day limit). G(T) is the gain of each tweet which is computed as above.

4.3.1.3 *Latency*

In the RTF 2015, the latency was combined with EG metric as described in Equation 4.1. Since 2016, this metric has been given up and the latency was reported separately in addition to the precision and recall-oriented metrics. The latency is evaluated in terms of the mean and median which were computed only for pushed tweets that contribute to gain. For a given tweet, its latency is defined as the difference between the time the tweet was pushed and the time of the first tweet in the semantic cluster that the tweet belongs to. This means that a system may have a high latency even if it submits a tweet immediately after it is identified.

Notice here that this choice to separate latency from quality is not unanimously supported. Hubert et al. [66] suggest that this choice should also be reconsidered because splitting between latency and quality may lead to some side effects that could be avoided with a single-point metric.

4.3.1.4 *The issue of silent days*

An interesting issue that arises when evaluating the performance of real-time push notification is how gain should be computed for days in which there are no relevant tweets. Indeed, due to the setup of the task, it was empirically observed that for some days no relevant tweets occur in the judgment pool [92]. Intuitively, from the end user perspective, systems should be rewarded for identifying that there are no relevant tweets for some days and for some topics. Inversely, systems that submit tweets in such case should be penalized. Days in which there are no relevant tweets for a particular topic are called $''silentdays''$, in contrast to $''eventfuldays''$ (where there are relevant tweets) [92].

Tan et al. [92] examine this issue and proposed two variants of nCG and EG metrics namely (nCG-1, nCG-0) and (EG-1, EG-0) which were adopted in TREC RTS 2016. In nCG-1 and EG-1 variants, the system receives a score of one if it does not push any tweets for a silent day, or zero otherwise. This means that under these metrics an empty run (a system that never returns anything) may have a non-zero score. In nCG-0 and EG-0, all systems receive a gain of zero no matter what they do for the silent day, therefore, it never hurts to push tweets.

Later, a study conducted by Roegiest et al. [129] showed that EG-0 and nCG-0 metrics are poorly formulated metrics because they correlate with the number of submitted tweets. Under this metric, systems are not penalized for being talkative, and therefore systems that push more relevant tweets tend to score higher. Hence, these variants were disused in TREC RTS 2017 and new variants were introduced by the organizer, namely EGp and nCGp (*p for proportional*). As an alternative to the 0-1 discontinuity on EG1 and nCG1 metrics on silent days, EGp and nCGp suggest that the penalty be gradually increased from 0 to 1 according to the number of submitted tweets with the limit of ten-tweet daily quota. For a day D the EGp and nCGp scores are computed as follows:

$$EGp(S,D) = \begin{cases} 1 - \dfrac{\min(10,N)}{10} \; if\, D\, is\, a\, silend\, day \\ \dfrac{1}{N} \times \sum\limits_{T \in S} G(T) \; otherwise \end{cases} \tag{4.4}$$

$$nCGp(S,D) = \begin{cases} 1 - \dfrac{\min(10,N)}{10} \; if\, D\, is\, a\, silend\, day \\ \dfrac{1}{Z} \times \sum\limits_{T \in S} G(T) \; otherwise \end{cases} \tag{4.5}$$

Where N is the number of submitted tweets in day D and Z is the maximum possible gain (given the 10 tweet per day limit). The EGp and nCGp scores of the summary S is the average scores across all days in the evaluation period.

Finally, note that EG-1 and EG-p metrics were respectively considered as the official metrics in 2016 and 2017 tracks.

4.3.2 *In-situ evaluation*

The in-situ evaluation followed the interleaving strategy proposed by Qian et al and Roegiest et al. [120, 129]. Judgments happened online as systems submitted tweets. A full description of the platform used for gathering online relevance judgments for mobile push notifications in RTS track is provided in [128]. Tweets pushed by system participant were routed directly to the mobile phone of assessors who may choose to judge the tweet immediately or later if it arrives at an inopportune time. Note that in this evaluation, a tweet might be judged by more than one mobile assessor. For each tweet, an assessor makes one of three judgments [129]:

- relevant, if the tweet contains relevant and novel information;
- redundant (e. g.duplicate), if the tweet contains relevant information, but is substantively similar to another tweet that the assessor had already seen;
- not relevant, if the tweet does not contain relevant information.

In this evaluation, the redundant might be complex to assess due to the interleaved character of the in-situ evaluation. An assessor is likely to encounter tweets from different systems. This suggests that an assessor might judge a tweet as redundant because the same information was seen is a tweet previously pushed by another system. From the aforementioned judgments, two aggregate metrics are computed namely online precision and online utility. Note that there is no good way to compute a recall-oriented metric since we have no control over when and how frequently user judgments are provided.

4.3.2.1 *Online precision*

The precision of tweets pushed by a system is simply computed as the fraction of relevant judgments. Two variants of the precision metric were considered namely "strict" and "lenient" precision. In the former, the system doesn't get credit for redundant judgments while in the latter redundant tweets are considered as relevant. These metrics are defined as follows:

$$P_S = \frac{relevant}{relevant + redundant + notrelevant} \tag{4.6}$$

$$P_l = \frac{relevant + redundant}{relevant + redundant + notrelevant} \tag{4.7}$$

4.3.2.2 *Online utility*

This metric measure the total gain received by the user. It is an alternative to online precision [129]. By analogy to the precision variants, two variants of online utility were defined as follows:

$$U_S = relevant - redundant - notrelevant \tag{4.8}$$

$$U_l = (relevant + redundant) - notrelevant \tag{4.9}$$

Where U_S and U_S denotes the "strict" and "lenient" utility respectively.

4.3.3 *Reusability of the collection*

Tracks organizer has made publicly available the evaluation script as well as the related ground truth files. This allows research groups to compare new approaches against the TREC track results using the replay mechanism (which, by the way, what we actually did in our experimental evaluations). A thorough analysis conducted by Hubert et al. [66] reveals some limitations on the batch evaluation framework of *scenario A* adopted in TREC RTS 2016 and 2017. In addition, authors identify some precautions to be taken into account when reusing the evaluation framework. The main finding of this study are as follows:

1. Tweets are evaluated according to the day that corresponds to their publication time instead of the day on which they were submitted. This significantly impacts the way metrics are evaluated particularly in the silent days. Nevertheless, it is important to note that this limitation concerns only systems that delay the submission of tweets. In such case, a tweet may be pushed a day after its publication. If this case occurs in a silent day, the system should be penalized for pushing a tweet. However, this is not the case in the evaluation framework since the pushed tweet is considered as if it was pushed in the day before (which corresponds to its publication day).

2. The coverage is not really evaluated by the official metrics. A consequence, systems returning few tweets are likely to score higher than systems that try to maximize the coverage by submitting more tweet;
3. Concerning the reusability of the collection, it was observed that unassessed tweets in the judgment pool were ignored instead of being considered as irrelevant. This behavior leads to erroneous evaluation results. This bias can be solved by adding all returned tweets when using a replay mechanism in the epoch file that contains the publication date of tweets from the pool.

Note that we take into a count these precautions in the experiments carried out in our work as follows:

- In the proposed approaches for real-time push notification, tweets are submitted as soon as they are identified without delay. This means that the publication and pushing days are always the same. Hence, our approaches are not concerned by the first precaution discussed above.
- As recommended by Hubert et. al, [66], we added all tweets returned by our approach to the epoch file which contains *tweetids* and publication times of tweets of the judgment pool. This means that tweets that are returned by our approach and do not exist in the judgment pool are considered as irrelevant instead to be ignored when computing the evaluation metrics.

In addition, it is worth mentioning that tweets collection used in our experiments were crawled during the evaluation period of each track which means that there is no lost data. If we had crawled tweets after the TREC evaluation period, we would have a corpus with missing data because not all tweets remain available.

4.4 Retrospective summarization evaluation

Retrospective tweet summarization is commonly evaluated using intrinsic methods. This evaluation is based on a direct comparison of the automatically generated summary against one or more manual summaries. To perform this evaluation one or many manual summaries need to be created for each topic. In this method, the quality of automatic summarization system is evaluated using the ROUGE metric (Recall-Oriented Understudy for Gisting Evaluation)[83, 82]. This metric calculates the recall that indicates the word overlap between the "ideal" summaries created by humans(generated manually) and the summary generated automatically. Among different ROUGE measures, the ROUGE-1 metric is commonly used to evaluate the quality of tweets summary because on the one hand, it is the most consistent with the human judgment [82] and on the other hand, tweets are short and informal. ROUGE-1 counts the total number of matching 1-grams (excluding stop-words) between the reference summary and the summary generated by the model. The intrinsic evaluation based on the aforementioned metric has been used in many works [140, 141, 7, 147, 142, 67].

However, the intrinsic methodology based on the ROUGE metric has many shortcomings when it comes to evaluating the performance of a long ongoing event summarization. Perhaps the most critical is the fact that generating summary manually on the basis of a given set of "relevant tweets" means that many other relevant tweets are missed out. Additionally, in the case of a large-scale dataset that includes a considerable number of events, building summaries manually is apparently impractical due to the size of the datasets (in terms of the number of relevant tweets as well as the number of considered events). It is interesting to note that the dataset used in the experimental evaluation in [7, 67] includes only ten (10) topics. The same number of topics (10) was also considered on the experiments carried out on [140, 141].

In the TREC RTS track, retrospective summaries (which correspond to scenario "B" runs) are evaluated using Normalized Discounted Cumulative Gain (NDCG)[68].NDCG evaluates the usefulness of a retrieval system for retrieving and ranking documents by decreased order of relevance. In contrast to binary relevance judgment used by recall-precision metrics, NDCG supports gradual relevance scale. This metric gives higher value to the well-ranked list. First, Discounted Cumulative Gain DCG is computed at the n^{th} position for each topic as follows:

$$DCG@n(S) = G(T_1) + \sum_{i=2}^{n} \frac{G(T_i)}{log_2(i)} \tag{4.10}$$

Where $G(T_i)$ is the gain of the ith tweet in the summary S which is set as described in batch evaluation methodology Section 4.3.1. The NDCG@n of a summary S of a given topic is computed as:

$$NDCG@n(S) = \frac{DCG@n(S)}{IDCG@n(S)} \tag{4.11}$$

Where $IDCG@(S)$ is the ideal DCG obtained by the perfect ranking of returned tweets where $G(T_i) \geq G(T_{i+1})$.

4.5 Conclusion

In this chapter, we presented the datasets and the framework that we will follow during the experimental evaluation of the proposed approaches in the next chapters. Moreover, we focused on the methodologies used to evaluate the prospective notification system since the nature of this task offers the possibility to align two different methodologies namely the tradition batch evaluation and the online in-situ evaluation. Finally, we presented the precautions that we take for reusing the data collection in order to provide a fair comparison with the official results obtained in the TREC track.

In the batch evaluation, the precision and the recall of prospective summaries are evaluated through two gain oriented metrics namely, expected gain (EG) and normalized cumulative gain (nCG). In the in-situ evaluation, online precision and utility metrics were defined to evaluate a prospective summary.

For a retrospective summary, the evaluation is based on the batch methodology. The quality of a summary is estimated through the normalized discounted cumulative gain metric.

Part III

REAL-TIME MICROBLOG FILTERING

Information Filtering and Information Retrieval: Two Sides of the Same Coin?

— Belkin, Nicholas J. and Croft, W. Bruce

5 PROSPECTIVE TWEET SUMMARIZATION BASED ON WORD SIMILARITY MODEL

5.1 Introduction

Prospective summarization (push notification) within social media stream differs from other information retrieval tasks by a specific information need of a user. Indeed, in this task, the notion of relevancy of a tweet is defined as a combination of three factors: interesting content (relevance), novelty, and timeliness (latency between notification and publication time). To be effective, notifications should be relevant, novel, and timely [86, 87, 150, 129].

As shown in Chapter 3, the widely-adopted approach to compute the relevance score of an incoming tweet use collection based statistics such as inverse document frequency, collection frequency [88, 30, 171, 141] or the query terms frequency [59, 91]. These approaches have several limits that stem from the shortness, the context dependency nature of a tweet and the streaming character of the collection. These arise two issues when computing the relevance of tweet with respect to a topic of interest. The first issue is the term mismatch and the second issue is the uselessness of the term frequency since a term rarely occurs more than once in a tweet. Additionally, the statistics about a term such as (Inverse Document frequency) are not always available in particular at the beginning of monitoring the tweet stream and may change each time a new tweet arrives.

Regarding the novelty detection, most of the existing approaches are based upon a pairwise similarity/divergence measures such as cosine similarity, Kullback-Leibler divergence [77], word overlap and Jaccard coefficient. Due to the volume of the tweet stream, the comparison of the incoming tweet with previous ones in the stream turns out to be not feasible because of its computational complexity. A way to avoid the computationally expensive comparisons of the new tweet with regard to the previously seen ones is to conduct a pairwise comparison between an incoming tweet with those already selected in the summary.

In order to cope with these issues, we introduce in this chapter, a word similarity based model [34] that takes advantage of word embedding representation to assess term-term matching. Our approach takes into account the aforementioned issues and it is designed to work at a high-velocity stream of incoming posts. The main contributions of this work[34, 33] are detailed in the following:

1. To evaluate the relevance of an incoming tweet with respect to user's information need, we propose an adaptation of the Extended Boolean Model (EBM)[136] in which the weights of query terms are estimated by taking advantage of the word embedding model *Word2Vec* [106]. Instead of using TF-IDF weighting scheme, we propose to consider the similarity between a term of the query and all tweet's terms as the weight of the given query term. Therefore, the query-tweet matching is computed according to the Word2Vec representations of their respective terms.

Each query term is matched with all tweet's terms of the incoming tweet. The similarity between two terms is computed by cosine similarity between their vector in the word2vec embedding model.

2. The novelty score of the candidate tweet is computed regarding all words of tweets already pushed to the user instead to examine it against all previously pushed tweets. This enables to avoid pairwise comparison with previous tweets in the summary, thus scaling better and allowing to reduce the computational complexity. The novelty of the incoming tweet with respect to the summary set is computed using a modified version of word overlap similarity measure.

For the timeliness (latency) of the notification, a system has to make a choice between pushing the incoming tweet as soon as it appears in the stream or waits and see whether it is worth pushing. By delaying the submission of tweets, a system may miss the occasion of pushing novel information and hence pushes an outdated information. To fulfill the prospective information need, the system should achieve a good trade-off between latency and quality. The latency should not be traded for a higher notification quality.

To evaluate the quality (relevancy and novelty) and the timeliness of notification delivered by our approach, we carried out several experiments on TREC MB RTF 2015 [85], TREC RTS 2016 [86] and TREC RTS 2017 datasets. Moreover, we first compare the effectiveness of the proposed approach against the state-of-the-art approaches based on stream statistical features and query terms occurrence. Finally, we compare the results obtained by our approach using a replay mechanism of scenario "A" on the aforementioned TREC tracks against the high-performing official runs of these tracks.

In the next section, we describe our general approach. Then we describe our experimental setup (section Section 5.3) and the corresponding results (section Section 5.4)

5.2 Real-time tweet filtering approach based on word similarity model

5.2.1 *Method overview*

Knowing that to fulfill the prospective information need an effective system needs to optimize three constraints: the relevance with respect to the topic of interest, the novelty/redundancy and the latency between the publication time and the notification time of selected tweets. To satisfy these requirements, we propose an approach that consists of three filters adjusted sequentially and in which the decision to select/ignore a tweet is made as soon as the tweet is collected.

Figure 5.1 depicts an overview of our approach that monitors the continuous stream and acts like a filter with three levels related to the topicality and the quality of tweet, its relevance, and its novelty respectively. The first filter is a simple tweet quality and topicality filter. It will be described in the experiment section (Section 5.3.2.2). The second and third filters are based upon the relevance and novelty measures that we will describe in the next subsections. The decision to select or discard an incoming tweet depends on whether its scores fall above certain thresholds.

Regarding the general framework presented in Figure 3.3, the proposed approach does not include either the query expansion and the external web page crawling com-

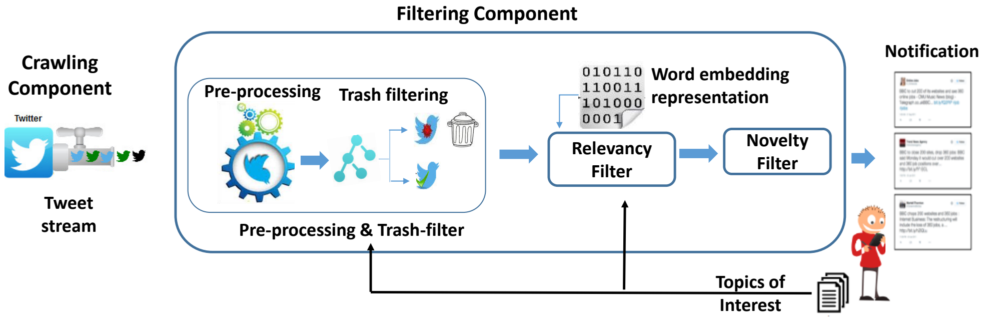


Figure 5.1: Overview of the real-time tweet filtering approach.

ponents or the tweet stream indexing module. Our approach relies on word similarity based on word embedding model [107] to compute the relevance score of the incoming tweet with regard to the event of interest. A word embedding is a distributed vector representation for words [107] in which each word is represented by a low-dimensional vector learned from raw text data.

Additionally, the evaluation of the novelty score is based on terms overlap. Hence, our approach does not require to maintain an index of tweet stream to compute the relevance and novelty scores as it is the case in the majority of the state-of-the-art approaches. We detail in the following subsections the proposed methods to compute the relevance and the novelty scores on the incoming tweet.

5.2.2 *Relevance estimation*

The relevance score of an incoming tweet with respect to a query (user interest) can be evaluated using statistical weighting based techniques such as BM25, vector space model or language model. However, in the case of an ongoing event, we believe that the use of relevance estimation functions based on cumulative statistics is not suitable. In fact, the main drawbacks of the use of statistical weighting techniques are: (i) at the beginning of the monitoring of the stream statistics are not available. (ii) It is required to update them each time a new tweet arrives which is a challenging task regarding the velocity in the tweet stream.

Another issue in the estimation of the relevance score of the incoming tweet with respect to the query is the shortness of tweets. Tweets have a limit length of 140 characters (280 since November 2017), are noisy and ungrammatical, which implies that the statistical features such as term frequency may be less useful. In addition, these characteristics of tweets arise the issue of word mismatch.

We believe that using similarity between tweets and query terms, where terms are represented by a vector (word embedding), is the key feature to overcome the aforementioned issues. To this aim, we propose an adaptation of the Extended Boolean Model (EBM) [136] to evaluate the relevance score of the incoming twee without relying on streams statistics. This is achieved by considering the weight of query words as the similarity score between query words and tweet words based on word embedding. This allows on the one hand to avoid the use of stream statistics and on the other hand to overcome the word mismatch issue. Indeed, word embedding-based retrieval models have the ability to tackle the word mismatch problem by making use of the similarity between distinct words.

In our approach, we adopt TREC like query representation in which a query Q (user interest) consists of a title Q^t and a description Q^d of the information need. The query title Q^t represents "**AND**ed terms" while Q^d represents "**OR**ed terms". In the Extended Boolean Model, the relevance scores of tweet $T = \{t_1, ..., t_n\}$ to "AND query" Q^t and "OR query" Q^d are estimated respectively as follows:

$$RSV(T, Q^t_{and}) = 1 - \sqrt{\frac{\sum_{q^t_i \in Q^t}(1 - W_T(q^t_i))^2}{|Q^t|}} \tag{5.1}$$

$$RSV(T, Q^d_{or}) = \sqrt{\frac{\sum_{q^d_i \in Q^d}(W_T(q^d_i))^2}{|Q^d|}} \tag{5.2}$$

where $W_T(q)$ is the weight of the query term q in the tweet T. $|Q^t|$ and $|Q^d|$ are the length of the title and the description of the query respectively. q stands for the term q^t_i in the query title Q^t or the term q^d_i in the query description Q^d.

We propose to estimate the weight $W_T(q)$ by evaluating the similarity between the query term q and all the terms of tweet T as follows:

$$W_T(q) = \max_{t_i \in T}[w2vsim(t_i, q)] \tag{5.3}$$

where $w2vsim(t_i, q)$ is the similarity between tweet word t_i and query word q. We propose to represent terms using word2vec word embedding model[106]. The similarity between two terms is measured by cosine similarity between their the vector representation of terms. In the case of either the term t_i or q is out of the vocabulary of the word embedding model, the similarity score is set to zero.

The intuition behind this proposition is that tweets that have words sharing many contexts with the query words will be more relevant. The context-aware similarity measure in word2vec allows considering different words with the same semantic meaning when computing the similarity between a tweet and a query. The main advantage of using word2vec model is that a query word which does not appear in a tweet but shares many contexts with the tweet words will get nonzero weight. Additionally, the relevance score of an incoming tweet is evaluated at the time the new tweet arrives, independently of tweets previously seen in the stream and without the need for indexing the tweet stream.

With $RSV(T, Q^t_{and})$ and $RSV(T, Q^d_{or})$, we got two relevance scores for tweet T regarding the title and the description of the query, respectively. The final relevance score of tweet T is measured by combining the aforementioned scores linearly with title terms having greater weight than description terms as follows:

$$RSV(T, Q) = \lambda \times RSV(T, QT_{and}) + (1 - \lambda) \times RSV(T, QD_{or}) \tag{5.4}$$

where $\lambda \in [0, 1]$ is a tuning parameter that determines the trade-off between the query title's words and the description's words. By setting λ to 1, only the relevance score with respect to the title of the query is considered. With $\lambda = 0$ only the description of the query is taken into account. We conduct in subsection Section 5.4.2.1 empirical experiments in order to set the appropriate value of this parameter for prospective notification in social media stream.

Remark: The proposed approach is tuned to handle TREC like query. However, it can be used for any other format of a query. We just have to set λ to 1 or drop the description part of the query.

5.2.3 *Novelty estimation*

In the context of real-time tweet filtering, the novelty is not symmetric because of the timeline constraint in the stream. We define "Novelty" to mean that information in the incoming tweet is not covered by tweets previously delivered. For a pair of tweets T and T', tweet T is considered novel with respect to tweet T' only if it is the chronologically later tweet and contains substantive information that is not present in the earlier tweet T'; otherwise, T is redundant. Note that redundancy and novelty are antonyms, and so we use them interchangeably, but in opposite contexts.

The widely adopted approaches to measure the similarity/ divergence between two tweets are based on a standard similarity/distance function such as cosine similarity or Kullback-Leibler (KL) divergence [77]. As discussed previously, the streaming character and the specific nature of tweets in which meaningful words rarely occur more than once suggests that the aforementioned similarity functions based on collection based statistics are less useful for evaluating the distance between two posts.

Another straightforward method of computing the similarity between two tweets is to use similarity measures based on term occurrence such as word overlap. This method does not require stream statistics and is simple to implement. However, this measure is symmetric which may be an issue in the novelty detection task. Indeed, in the case where the new tweet T contains all terms of an old tweet T' ($T' \subset T$), the tweet T is considered redundant event if it is longer than the T'. However, a longer tweet may provide new information (ie. tweet T completes or updates the information already provided in the tweet T') and hence it should be considered novel.

In this section, we introduce a new novelty score function based on word overlap and which does not use stream statistics. Due to the velocity of the stream, it seems natural to adopt a document-to-summary based approach to estimate the novelty score. In order to make the novelty estimation computationally efficient, we propose to use an aggregated comparison instead of a pairwise comparison in which the incoming tweet is compared to all tweets of the summary. Therefore, to avoid the pairwise comparison, the tweets of the summary are aggregated into a summary set of terms SW and the new tweet is compared to this summary set. The novelty score of the incoming tweet $T = \{t_1, ..., t_n\}$ is computed using word overlap as follows:

$$NS(T, SW) = 1 - \frac{|SW \cap T|}{|T|} \tag{5.5}$$

With:

$$SW = \bigcup_{j=1}^{M} \{t_1^j, t_2^j, ..., t_n^j\} \tag{5.6}$$

Where M is the number of tweets already selected in the summary and t_i^j is the i-th term in tweet j in the summary.

This novelty scoring function can be considered as a way to compare the incoming tweet with what have been already seen by the user in terms of redundancy and coverage, which is the essence of the task of novelty detection. The intuition of this measure is that if a new tweet contains information (words) that was not seen before in the actual summary, the incoming tweet is different and thus it can be considered as novel.

Also, notice that the number of overlapping words is divided by the size of tweet $|T|$ instead of the minimum of either the length of the summary word set $|SW|$ and the incoming tweet. This yields to get an asymmetric scoring function that fits better the task of novelty detection as was discussed earlier at the beginning of this section. In addition, a long tweet and a short tweet with the same number of overlapping words with SW will not have the same novelty score. In this case, the shorter tweet is penalized.

5.2.4 *Threshold setting*

In the relevance and novelty filters, the decision to select or discard the incoming tweet depends on whether its relevance and novelty scores fall above predefined thresholds. These thresholds are set as follows:

Relevance threshold: The proposed relevance function enables the use of simple threshold across all topics. Hence, instead of setting a single static predefined value that is the same across all topics, we propose to use an adaptive relevance threshold for each topic which is estimated at the time of a new tweet arrives. Our thresholding strategy is to consider the average of the previously seen values of the relevance score. However, under this strategy, we do not lower the threshold below a global minimum value (GT). Hence the relevance threshold is defined by:

$$max(GT, avg(RSV(T, Q)).$$

Notice that the use of a global minimum value allows dealing with cold start problem at the beginning of the monitoring of tweet stream. The value of GT was set experimentally.

Novelty threshold: For novelty detection, we adopt a simple thresholding strategy that selects a single static threshold value across all queries. The novelty threshold value was set experimentally using TREC MB RTF 2015 dataset.

5.2.5 *Pushing strategy*

Another trade-off that a prospective notification system has to deal with is the latency between the tweet's publication time and the notification time. In this context, two different pushing strategies can be adopted namely, wait and see or instantly pushing strategy. A system has to make a choice between pushing the incoming tweet that was identified as relevant and novel as soon as it becomes available or waits to accumulate evidence and see whether it is worth pushing. In the latter case, the system may miss the "opportunity" to push novel information. By delaying the submission of tweets, the notification may arrive too late since it may be about a tweet that is already seen by the user (e. g.a tweet might be retweeted by someone the user is following) or that conveys an outdated information. This is because an information reported about an ongoing event can rapidly become outdated. We argue that in the case of an ongoing event that evolves rapidly, taking a time to accumulate evidence before making a decision may hurt the quality of the notification.

For this reason, and in order to reduce the latency between the publication time and the notification time; we choose to adopt the instantly decision making strategy in which the decision to select/ignore a tweet is made immediately. Tweets that pass all these filters are pushed in a real-time fashion to the user without delay.

5.3 Experimental Evaluation

We carried out a series of experiments on large-scale real-world microblog data-set in order to evaluate the performance of the proposed approach. Our experiments are divided into four different parts. The first part concerns the evaluation of the proposed method to detect novel tweets. We began by the novelty detection because its evaluation can be carried out independently of the relevance. The second part is dedicated to evaluating the effectiveness of our approach Word Similarity Extended Boolean Model(WSEBM) to estimate the relevance of the incoming tweets with respect to a given topic. The third part is about the performance of the proposed approach in real-time tweet summarization task. Last and not least, the fourth part concerns a comparative evaluation of our approach with the official results on TREC real-time tweet summarization tracks.

The purpose of this evaluation is fourfold:

1. Comparing the performance of the proposed method for novelty detection with the state-of-the-art approaches;
2. Evaluating the effectiveness of WSEBM to estimate the relevance of the incoming tweet with respect to a given event;
3. Highlighting the effectiveness of immediate tweet pushing strategy. In this experiment, we aim to validate the hypothesis that delaying tweets submission hurts the quality of the notification;
4. Comparing the performance of our approach against the official results in TREC RTF 2015 [85], TREC RTS 2016 [86] and 2017 [87] tracks.

The experiments were carried out on tweet collections created from the recent Microblog tracks at TREC namely TREC MB RTF 2015, TREC RTS 2016 and 2017. Further details about these data collections and tracks were discussed in Chapter 4. Experiments conducted in this chapter are from post hoc runs using a replay mechanism over tweets of the aforementioned TREC Microblog tracks. Notice here, that corpora used in our experiments were crawled during the official evaluation period of each TREC track which means that there is no lost data. The loss of any tweet could have occurred if tweets had been crawled after the evaluation period because not all tweets remain available.

We present in what follows the evaluation protocol and then we discuss results and findings.

5.3.1 *Methodology*

We will use various evaluation protocols to assess the different aspects of our approach as follows:

1. *Protocol 1*: To evaluate the effectiveness of the proposed novelty detection method, we use a subset of tweets from each corpus (TREC MB 2015, 2016,2017) that consists of tweets from the judgment pool that were assessed relevant by NIST assessor. These tweets were clustered such as all tweets that share similar information were gathered in the same cluster. Within each cluster, the earliest tweet is considered novel; all other tweets in the cluster are redundant with respect to all earlier tweets. In this experimentation, we use these clusters (provided by TREC track organizer) as ground truth to evaluate the performance of novelty detection methods. A general statistics of the aforementioned subsets are shown in Table 5.1.

Table 5.1: Novelty detection datasets statistics.

Corpus	Novel	Redundant	Total
TREC 2015	3100	5135	8235
TREC 2016	765	2575	3340
TREC 2017	2889	3260	6149

2. *Protocol 2*: The retrieval effectiveness of the Word Similarity Extended Boolean Model(WSEBM) is evaluated in terms of ranking quality. To this aim, we follow a more like a top-100 retrieval task based on a one-day tweet collection. Tweets are ranked according to their relevance score per day per topic. We are interested in these experiments in the identification of relevant information in the social media stream regardless of the latency. Runs in these experiments were generated as follows: First, we filter tweets by applying the quality filter that we will describe below in Section 5.3.2.2. Then, for each day we select iteratively the TOP-100 tweets but with discarding those having a similarity score above the predefined threshold.

3. *Protocol 3*: To evaluate the performance of the proposed method on real-time tweet summarization task, we carried out experiments using a replay mechanism of scenario "A" over tweets captured during the evaluation period of TREC RTF 2015, RTS 2016 and 2017 tracks. We follow this protocol to evaluate the impact of the relevance threshold value (Section 5.4.3.1) and the effectiveness of immediate pushing strategy (Section 5.4.3.2) as well as to compare the performance of our approach against the official results in TREC tracks (Section 5.4.4).

5.3.2 *Parameter setting*

5.3.2.1 *Word embedding model*

Word embeddings have been utilized in many applications in the natural language processing and information retrieval for their ability to model term similarity and other relationships. Many pre-trained word embeddings built from global training dataset have been made available such as Google's trained Word2Vec[1]. It includes word vectors for a vocabulary of 3 million words and phrases that they trained on roughly 100 billion words from a Google News dataset. In our experiments, we can use for instance Google's trained Word2Vec model. However, Diaz et. al,[42] have shown that word embeddings when trained globally underperform embeddings trained on the specific corpus for retrieval tasks. For this reason, authors suggest that embeddings should be learned on a topically-constrained collection, instead of large topically-unconstrained corpora.

Following Fernando et al. [42] suggestion, we propose to train a word embedding model from tweets collection crawled before the period covered by the dataset used in our experiments. Besides, our choice is also motivated by the fact that the use of word embeddings models trained globally may arise the problem of word out-of-vocabulary because of the specific writing style used in tweets. Indeed, tweets are informal and ungrammatical which means that many words that occur in tweets may not be presented in the embeddings models trained globally. Thus, we argue that learning word embedding model on tweet collection may reduce the problem of word out-of-vocabulary.

For each dataset (TREC RTF 2015, TREC RTS 2016 and 2017), we built a separate word embedding. As training data, we used tweets crawled by Twitter stream API during 9 days before the official evaluation period of each track. The crawling period and the size of the three training collections used to build the word2vec models are presented in Table 5.2.

Table 5.2: Word2vec training corpus statistics.

Corpus	Crawling Period	Tweets	Terms
TREC 2015	11-19/07/2015	8,084,633	242,419
TREC 2016	23/07-01/08/2016	11,952,354	327,659
TREC 2017	20-28/07/2017	15,776,692	432,396

Notice here that we performed the following preprocessing steps on crawled tweets before generating word vector. First, we drop non-English tweets and those containing less than 3 terms. We removed stop-words, URLs and then we tokenize tweet text and we perform stemming.

We use the skip-gram learning schema of word2vec model, which produces better word vector for infrequent words than Continuous Bag-of-Words (CBOW) learning schema [106]. In the skip-gram model, given word context (the maximum distance between two words) denoted (C), for each training word a number R is selected randomly in the range [1;C], and then use R words from history and R words from the future of the current word as correct label. The dimension of the word vector was set to 300 and the context window (the maximum distance between two words) was set to 5 since the average length of a tweet is 11 words.

[1] http://mccormickml.com/2016/04/12/googles-pretrained-word2vec-model-in-python/

5.3.2.2 *Pre-processing step and quality filtering*

The pre-processing step consists of stop-words removal, stemming and tokenize the tweet using Twokenize tool [2]. Then, we apply a simple quality and topical filters to discard potential trash and irrelevant tweets and yields to boost the efficiency of our approach to handle the velocity of the tweet stream. The quality filter excludes any tweet that does meet at least one of the following rules:

- **Non-English Filtering:** We rely on Twitter's language detector to discard the non-English tweets. In addition, tweets that contain more than 35% of non-English characters are also filtered out.
- **BadWrods Filtering:** Tweets including swear or bad words are filtered out since we assume that it would be inappropriate to push notification containing such kind of vocabulary.
- **Retweet de-duplication:** Since the practice of "retweeting" the same content is very common on Twitter, we de-duplicated tweets using retweet mechanism and tweet identifier (tweet id). If the incoming tweet is a re-tweet of an already seen tweet in the stream then this new tweet is eliminated.
- **Trash filtering:** Any tweet that meets one of the following conditions is considered as trash and hence it is filtered out:
  - It contains less than five unique tokens;
  - It includes more than one URL ;
  - It mentions more than two usernames ;
  - It contains more than three hashtags ;

The topical filter step is a word overlap filter that drops all incoming tweets that do not contain a predefined number of query words. The incoming tweet T is considered as a candidate tweet if its number of overlapping words with the query title is higher than the minimum of either a predefined constant (K) or the number of words in the query title $min(K, |Q^t|)$.

To set the constant K used in the pre-processing step and quality filtering, we carried out an experimental evaluation of the quality of the tweet filter based on TREC MB RTF-2015 data set. Two values were tested (i) at least one word (K=1) and (ii) at least two words (K=2). Table 5.3 reports the precision and recall obtained by each filter. As shown in the last row, the filter (at least 2 query words) increases significantly the precision. The number of tweets that pass this filter is 15878 while the number of tweets that pass the filter (at least 1 query word) is 140 times larger. The filter (at least 2 query words) captures about 40% of relevant tweets while the filter (at least 1 query word) return 74% of relevant tweets but it also brings up a lot of noise. These results motivated our choice to use (at least 2 query words) as a threshold. Since our goal is to generate a concise summary, we think that having 40% of relevant tweets might be enough to reach this purpose.

[2] http://www.ark.cs.cmu.edu/TweetNLP/

Table 5.3: Quality of the tweet filter on MB RTF-2015 data set.

K	R	S	RS	Precision	Recall
K = 1	8164	2225344	6101	0,0027	0,7473
K= 2	8164	15878	3344	0,2106	0,4096

Note. R, S and RS are the total number of relevant tweets, selected tweets and relevant selected tweets, respectively.

5.3.3 *Baselines*

Regarding the novelty detection method, the different baselines from the state-of-the-art that we will compare our novelty detection method are; on the one hand, the stream statistics approaches based on a vector space model (cosine similarity) and based on a language model (minimum Kullback-Leibler (KL) divergence) and on the other hand the terms co-occurrence approach (standard word overlap). We implanted these baselines as they were described in Section 3.6. All these baselines are pairwise, the incoming tweet is compared to the previous N tweets already pushed to the user. Notice that a vector space and a language model based methods can be used for either pairwise or aggregate comparison. However prior work showed that these methods perform better with a pairwise comparison [154] [72].

For the proposed relevance model (WSEBM), we will compare it to the following baselines:

- The stream-based baselines used in the related work (Section 3.5) which rely on the collection-based statistics such as IDF and average tweet length. These baselines include a Cosine similarity as a baseline for Vector Space Model with TF-IDF weight, Okapi BM25, TF-IDF and HybridTF-IDF[141]. Note that the two latter methods were recommended by [97] to be considered as baselines since it turned out to be the best one among 11 different tweet summarization approaches.

- The best performing approach in TREC MB 2015 track [91]. This baseline is tweet-based since the relevance estimation depends solely on the number of occurrences of query terms and the length of the given tweet (Equation 3.6).

- The standard Extend Boolean Model (EBM). This baseline is based on the proposed equations (Equation 5.1, Equation 5.2) in which we consider the TF-IDF weight of the query term in the tweet instead of using the proposed word embedding similarity weighting technique. Notice that in this baseline, the relevance scores regarding the title and the description of the query are linearly combined in the same way as the proposed approach (Equation 5.4). This baseline allows evaluating the effect of using word embedding similarity as weighting technique.

To evaluate the impact of the use of the instantly pushing strategy, we adopt as a baseline the approach that delays the submission of relevant tweet until the end of the day. In this baseline, up to ten (10) relevant tweets are iteratively selected with excluding those having a similarity score above the predefined threshold. In fact, this baseline corresponds to the scenario "B" of TREC RTS track with pretending that tweets were emitted at 23:59:59 for each day.

5.4 Results

In this section, we present and review the results of the experiments carried out on the datasets mentioned in the previous sections. We first report the performance of the proposed novelty detection method. Then, we evaluate the relevance scoring function followed by a study that highlights the impact of relevance threshold value and the use of the instantly tweet pushing strategy. We finally report the performances of our approach in real-time tweet stream summarization task and we compare the obtained results with the official results of TREC RTF 2015, TREC RTS 2016 and 2017 tracks.

5.4.1 *Effectiveness of the novelty detection method*

5.4.1.1 *Comparative evaluation with state-of-the-art baselines*

In this sub-section, we follow protocol 1 to evaluate the novelty scoring function described in sub-section Section 5.2.3 (Equation 5.5) on TREC MB 2015, 2016 and 2017 datasets. The performance of our method for novelty detection task is compared against the three common baselines approaches from the state-of-the-art. In these approaches (Max Cosine Similarity, Min KL Divergence, and word overlap), the incoming tweet is compared against tweets previously pushed to the user. The new tweet is considered novel if its novelty score falls above a threshold. In our experiments, we gradually vary the novelty threshold value from 0.05 to 0.95 at the step of 0.05.

Methods are labeled using the following convention: the first part indicates the similarity/ distance measure (word overlap (WO), minKL, or maxCOS), the second part indicates if the method is used pairwise (tweet to tweet comparison(T2T)) or aggregate (tweet to the summary (T2S)).

The performance of a novelty detection method is evaluated in terms of missed detection and false alarm error probabilities (P_{Miss} and P_{Fa}) as defined by [53]. P_{Miss} is the probability of missing a tweet that conveys a novel information whereasP_{Fa} is the probability of pushing a redundant tweet (false alarm). These probabilities are computed as follows:

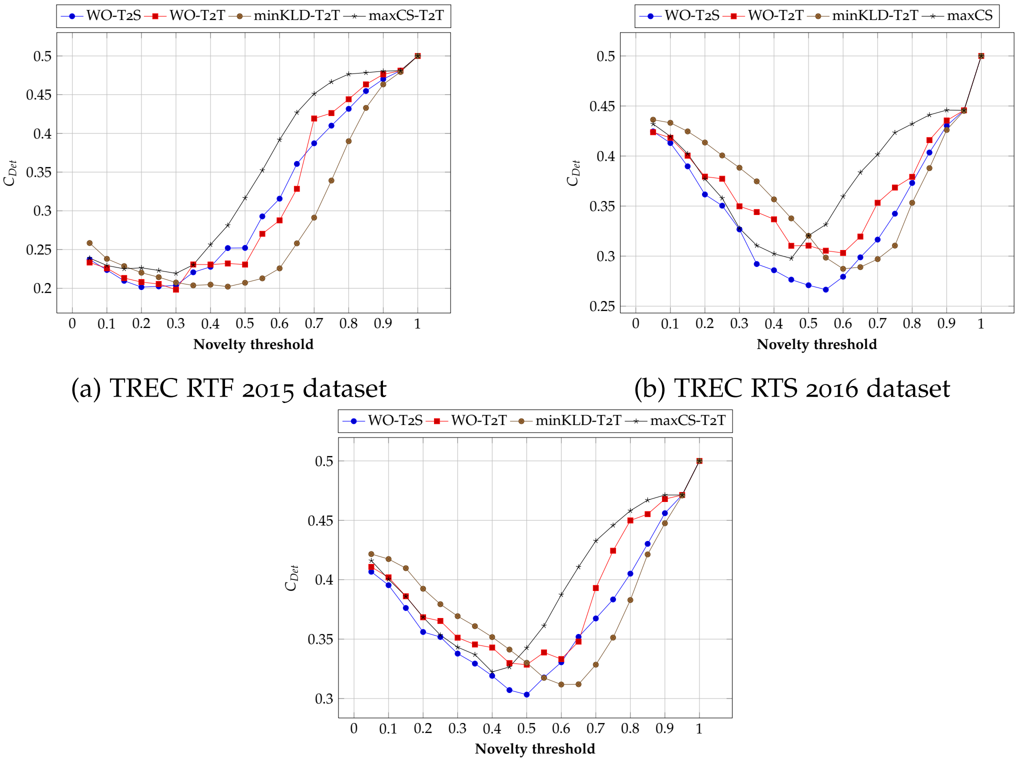
$$P_{Miss} = \frac{Numberofmisseddetections}{Numberofclusters} \tag{5.7}$$

$$P_{Fa} = \frac{Numberofredundanttweetspushedtotheuser}{Numberofredundanttweets} \tag{5.8}$$

These probabilities are inversely correlated. When the false alarm increases the miss detection decreases. Hence a novelty detection system tends to find a tradd-off between these probabilities. To evaluate the performance of novelty detection, the false alarm and miss detection probabilities are linearly combined to a single detection cost as follows [72]:

$$C_{Det} = \frac{1}{2}P_{Miss} + \frac{1}{2}P_{Fa} \tag{5.9}$$

A perfect system would score 0 in the detection cost function. A naive system which is always yielding yes or no scores 1. As the purpose of a novelty detection in the context of the tweet stream filtering task is to minimize both missed novel tweets and



(a) TREC RTF 2015 dataset

(b) TREC RTS 2016 dataset

(c) TREC RTS 2017 dataset

Figure 5.2: Comparison of the detection cost (C_{det}) with state-of-the-art baselines over different values of novelty threshold.

false alarms (pushing a redundant tweet), a novelty detection method should minimize the detection cost.

We present in Figure 5.2 the detection cost on the three tweets datasets over different novelty threshold values. An approach is considered to perform best when it has its curve towards the lower of the graph and the minimum detection cost is used to define the optimum threshold. The lowest detection cost value corresponds to the optimal trade-off between avoiding sending redundant tweets and missing out novel ones. Hence, the threshold value that gets the lowest detection cost is the best value to use.

From Figure 5.2, we notice that among baselines, minKL method achieves very good performances overall datasets and particularly on TREC 2015 dataset where it is the best performing method. This corroborates with previous results which highlighted the high performance of minKL in the novelty detection task [154].

We observe that for TREC 2016 and 2017 datasets our novelty detection method (WO-T2S) obtains the lowest detection cost overall baselines and offers slightly high detection cost than the best performing method on TREC 2015 dataset. It can be noted that WO-T2S which is an aggregate comparison provides little to no difference performance than a pairwise comparison (WoT2T) in the terms of detection cost on TREC 2015 dataset. The best performance of each method in terms of detection cost with the miss detection P_{Miss} and false alarm P_{Fa} probabilities are reported in Table 5.4.

The obtained results are encouraging, our simple method outperforms the baselines on TREC 2017 and 2016 datasets and manages to follow narrowly the best performing baseline in TREC 2015 datasets. We found performance improvements in terms of de-

Table 5.4: The lowest detection cost overall novelty threshold values on TREC 2015, 2016 and 2017 datasets.

	TREC 2015			TREC 2016			TREC 2017		
Method	P_{Miss}	P_{Fa}	C_{Det}	P_{Miss}	P_{Fa}	C_{Det}	P_{Miss}	P_{Fa}	C_{Det}
WO-T2S	0.0763	0.3343	0.2053	0.2472	0.2859	**0.2665**	0.1592	0.4473	**0.3033**
WO-T2T	0.1065	0.2990	0.2032	0.3046	0.3017	0.3032	0.1566	0.5004	0.3285
minKL-T2T	0.1306	0.2734	**0.2020**	0.1290	0.4457	0.2873	0.1265	0.4972	0.3118
maxCS-T2T	0.1619	0.2715	0.2167	0.2589	0.3366	0.2977	0.1672	0.4777	0.3224

tection cost against the minKL up to 7.23% and 2.72% on TREC 2016 and 2017 datasets respectively. One can argue that the improvement of performance is not significant. However, our method is much more computationally efficient than minKL as shown in Section 5.4.1.2.

We can conclude that the aggregate comparison based on word overlap offers better results than the pairwise comparison. This is due to the fact that the aggregate method by merging all previously seen tweets increases the quality of novelty control in the case of a short document such as social media posts. This can be illustrated briefly by the following example:

Assume that the current summary contains two tweets T_1, and T_2 with $T_1 = \{w_1, w_2, w_3, w_4\}$ and $T_2 = \{w_5, w_6, w_7, w_8, w_9\}$ and suppose that the incoming tweet T_3 includes all terms that occur in both tweets T_1, and T_2 ($T_3 = \{w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9\}$). Then, in the aggregate comparison, the novelty score of T_3 is equal to zero whereas in the pairwise comparison, the novelty score is equal to 4/9.

Finally, we choose to set the novelty threshold value to (0.3) for the next experiments. This threshold value gets the lowest cost detection on TREC 2015 dataset (Figure 5.2a) which corresponds to the optimal trade-off between avoiding sending redundant tweets and missing out novel ones.

5.4.1.2 *Performance in terms of execution time*

We compare our approach in terms of run time with the best performing method from the baselines (minKL). We carried out experiments for different values of the number of tweets in the summary. The results are shown in Figure 5.3. The runtime values are in microseconds and correspond to the average time needed to process and assign a novelty score of the incoming tweet with respect to the current summary. the number of tweets in the summary varies from 1 to 1450 tweets.

It is clear that our approach is considerably faster than pairwise based approaches. The difference between these methods increases as the size of the summary increases since pairwise based approaches have to be executed on the entire summary to compute the similarity/divergence between the incoming tweet and all tweets in the summary. In Tweet-to-Tweet comparison, the average execution time per tweet keeps rising as the number of tweets in the summary increases while our method keeps having low execution time regardless of the length of the current summary. In the proposed method,

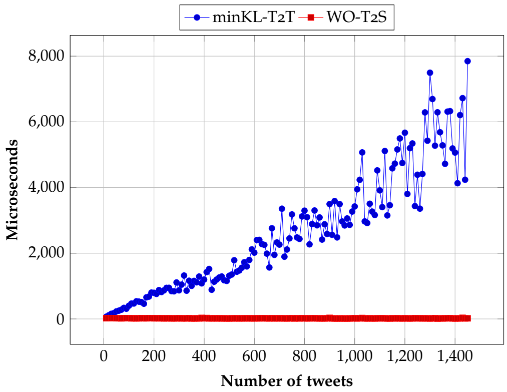
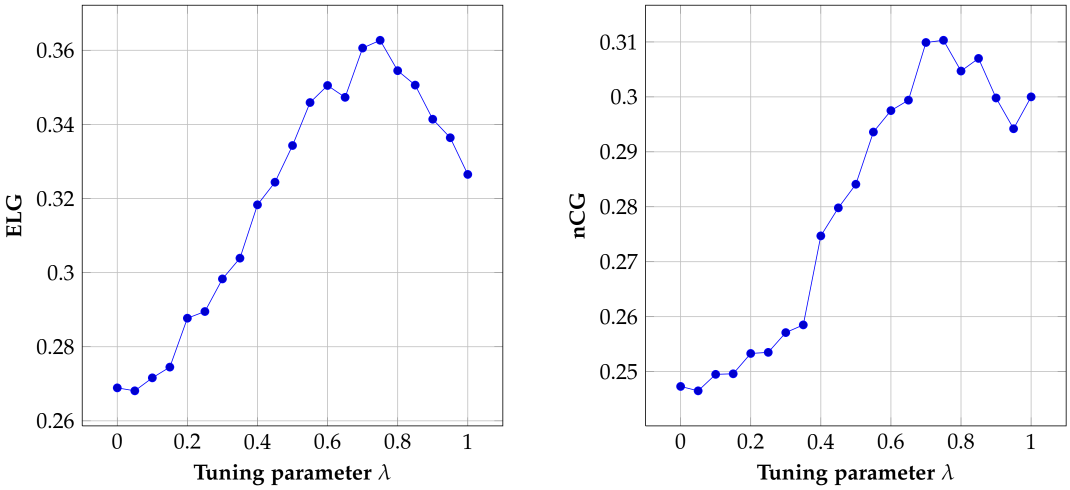


Figure 5.3: Average execution time per tweets, for different size of the summary, for $WO - T2S$ and $minKL - T2T$.



(a) Performance in terms of ELG

(b) Performance in terms of nCG

Figure 5.4: Tuning the parameter λ on TREC MB 2015 dataset

the response time is not affected by the length of the summary which allows to better cope with the velocity of the tweets stream.

5.4.2 *Evaluating retrieval effectiveness of WSEBM*

Before evaluating the effectiveness of our approach WSEBM to estimate the relevance of incoming tweets, we first conducted experiments on TREC MB 2015 dataset to tune the parameter λ of the Equation 5.4 that computes the relevance score of the incoming tweet.

In what follows, we present the impact of the tuning parameter λ followed by a comparison of the effectiveness of the proposed relevance estimation model with the state-of-the-art baselines.

5.4.2.1 *Effect of tuning parameter λ*

Parameter $\lambda \in [0, 1]$ is a balance factor between considering the title and the description of the query to estimate the relevance score. To examine the effect of λ, we vary it from 0 to 1 at the step of 0.1. Figure 5.4 shows the performances in terms of expected latency-discounted gain (ELG) and normalized cumulative gain (nCG) for different value of λ.

As shown in Figure 5.4, the use of the query description words yields better results than considering only the title query words because when using only the title of the query, tweets that contain all terms of the query get the same relevance score. We notice that the extreme emphasis on the title or on the description of the query, which corresponds to ($\lambda = 1$) and ($\lambda = 0$) respectively, causes performance loss. Therefore, we choose to set λ to 0.75 which means that the relevance score of the tweet with respect to the query title has higher importance than the relevance score with respect to the query. This makes sense since the title has few words while the description can be lengthy and contains general terms that are not directly related to the user's information need.

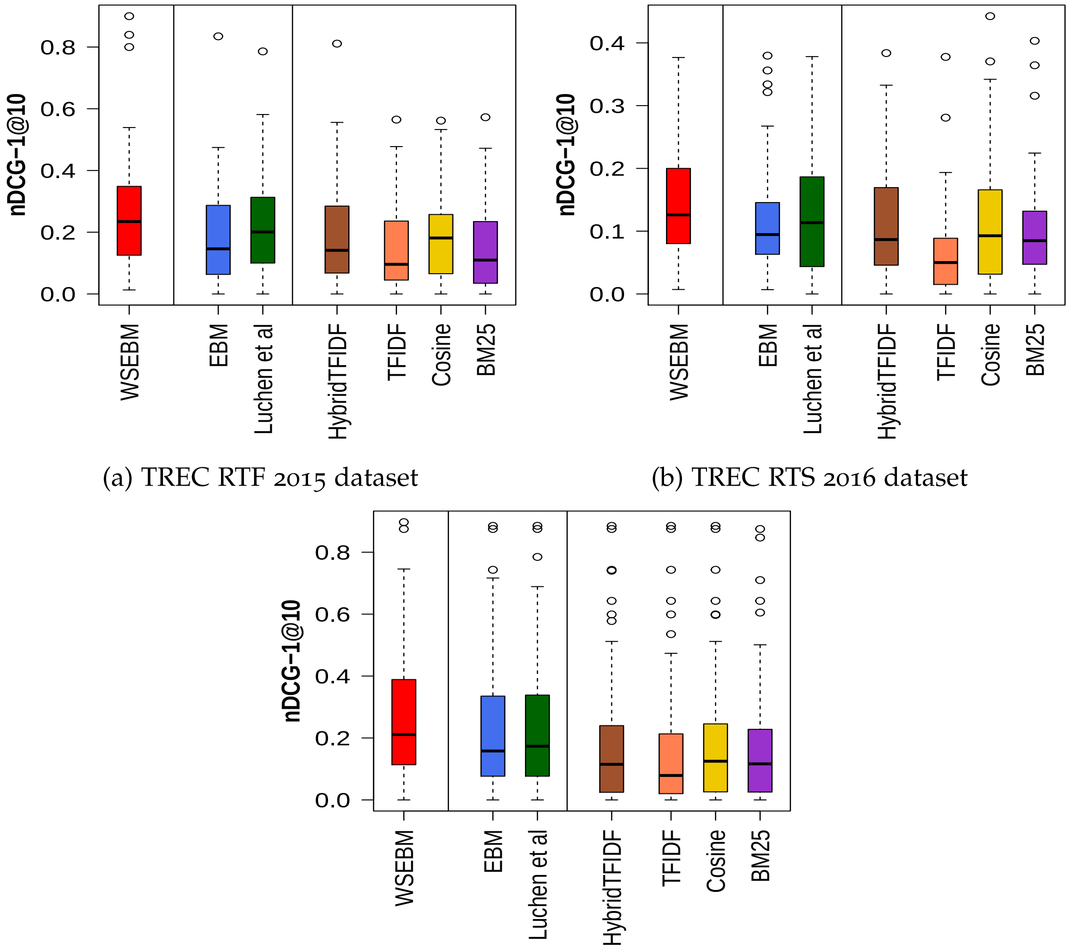
5.4.2.2 *Comparative evaluation with state-of-the-art baselines*

To evaluate the effectiveness of the proposed model to identify relevant content, we carried out a series of experiments according to protocol 2 in which we do not take into consideration the novelty. To do so, the novelty threshold was set to 0 and the system returned,for each day, a ranked list of tweets according to their relevance score. We compare the ranking quality of the proposed relevance model denoted by WSEBM for (Word Similarity Extended Boolean Model) against the six baselines from the-state-of-the-art on the three tweets collection (TREC MB 2015,2016,2017). In this experiment, we ignore topics for which no relevant tweet has been identified in the judgment pool since all systems get a score of zero for these topics in ranking task.

The result of the experiment is provided in the form of box-plots in order to capture the variance of the results. Performances in terms of nDCG@10 metric over the aforementioned tweets datasets are given in Figure 5.5. Error bars denote 95% confidence intervals around the mean which are computed from all topics.

Overall, it seems from these results that baselines based upon the number of query terms occurrence, namely EBM and approach proposed by Luchen et al [91] perform better than models that incorporate more text statistics such as BM25, HybridTF-IDF. This probably has much to do with the specific nature of tweets that have handful terms and in which terms rarely appear more than once. Among state-of-the-art models, the approach proposed by Luchen et al [91] obtained the best results over the three datasets followed nearly by EBM model. These results support our intuition that query term occurrence appears to be the key feature in the context of tweet retrieval and that the extend boolean model is suitable to assess the relevance score of a tweet with respect to other models.

We observe that our proposed model WSEBM outperforms all baselines over the three datasets with significant improvement compared to stream statistics-based methods (TFIDF, HybridTFIDF, BM25, cosine similarity). We found performance improvements in terms of nDCG@10 against the best stream statistics-based method (cosine similarity) up to 45.37%, 30.89% and 31.31% in TREC 2015, 2016 and 2017 datasets respectively.



(a) TREC RTF 2015 dataset

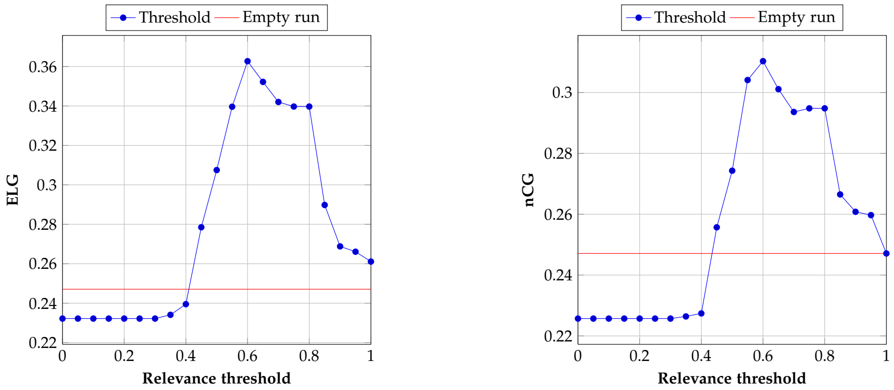
(b) TREC RTS 2016 dataset

(c) TREC RTS 2017 dataset

Figure 5.5: Comparison of the ranking quality with state-of-the-art baselines

By comparing WSEBM with EBM, we notice that the introduction of word similarity based on word embedding enhances the performances overall datasets with an improvement up to 14.48% 26.44% 22.81%. These results can be explained by the use of word embedding similarity which seems to be effective. Additionally, in EBM, two tweets that contain the same number of query terms get the same relevance score regardless of their length and the different terms that may have whereas in WSEBM these two tweets get different relevance score according to their terms that do not appear in the query. WSEBM awards the most score to terms that share the same context with query terms which boosts the relevance score of a tweet that contains such terms. Indeed, in the proposed relevance scoring function, query's terms that do not occur in a tweet but share many contexts with tweet's terms get score different from 0. As a result, it awards a high score to tweets containing query's terms and terms that share the same semantic context with the query's terms. Therefore, these tweets are boosted to the top of the returned summary.

These results confirm our previous claims that word mismatch issue between query and tweet introduces significant confusion and makes it harder to detect salient tweets. The use of the word similarity based on word embedding model allows leveraging the



(a) Performances in terms of ELG

(b) Performances in terms of nCG

Figure 5.6: The impact of the relevance threshold on TREC MB 2015 dataset. The horizontal red line indicates the score of an empty run.

semantic relationship between tweet terms and query terms. This corroborates with previous findings made by Zuccon et al. [172] who showed the positive effect of the use of word embedding for information retrieval.

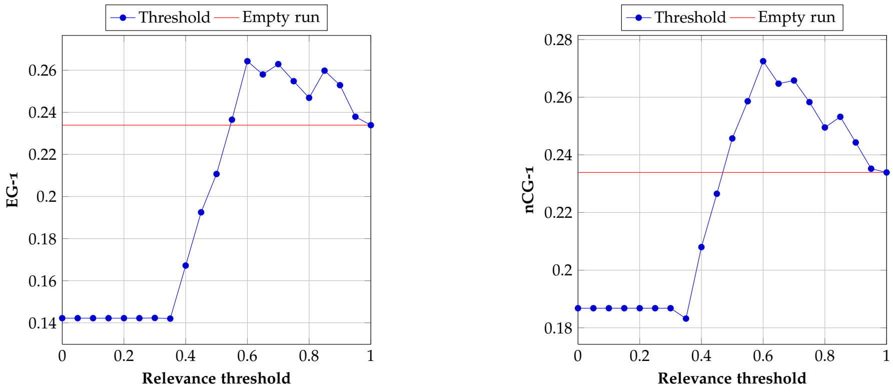
5.4.3 *Effectiveness of the proposed approach*

In this subsection, we report the performance of our approach in real-time tweet summarization at different values of relevance threshold used to filter out irrelevant tweets as described in Section 5.2.4. Then, we compare the performance obtained with the immediate tweet pushing strategy against "*wait& see*" pushing strategy.

5.4.3.1 *Effect of relevance threshold value*

The relevance filter makes a threshold-based decision. In this filter, a tweet is considered as a candidate for notification to the user only if its relevance score falls above of the minimum of either a global threshold value (GT) and the average of the previous relevance scores. To better understand the impact of the relevance threshold value, we conducted experiments in which we gradually vary the relevance threshold from 0 to 1 at the step of 0.05 and we keep the other parameters fixed (novelty threshold = 0.3, and tuning parameter $\lambda = 0.75$). Figures 5.6, 5.7 and 5.8 present the performances in terms of expected gain and normalized cumulative gain for different values of relevance threshold on TREC 2015, 2016 and 2017 datasets respectively. The baseline in these experiments is the empty run which corresponds to a system that never pushes tweets. This baseline receives a non-zero ELG, EG and nCG scores because no relevant tweets appeared for many topics on many days and system is rewarded for pushing nothing on such days by receiving the perfect score (1). Note that the empty run is a challenging baseline that many systems in TREC 2015, 2016 and 2017 tracks failed to beat.

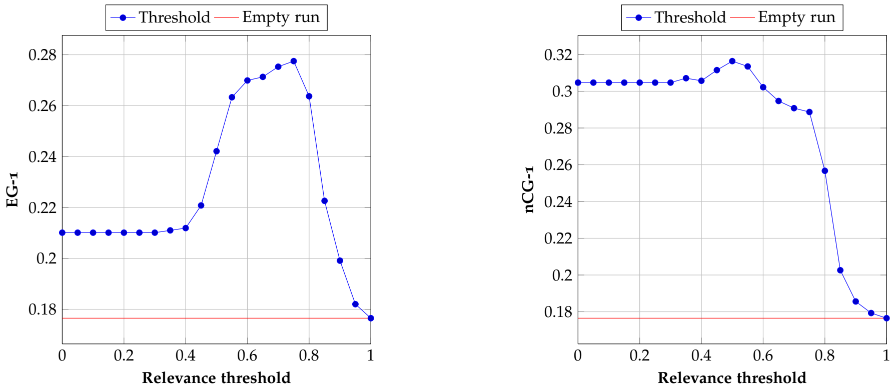
Overall, we note that the threshold has a serious impact on the filtering effectiveness. We observe that the best performance on the three datasets is obtained with different values of relevance threshold. The use of low or high threshold values hinders the performances of the relevance filter. This can be explained by the fact that the former case yields to push a few tweets whereas in the latter case many tweets may be submitted.



(a) Performances in terms of EG-1

(b) Performances in terms of nCG-1

Figure 5.7: The impact of the relevance threshold on TREC RTS 2016 dataset. The horizontal red line indicates the score of an empty run.



(a) Performances in terms of EG-1

(b) Performances in terms of nCG-1

Figure 5.8: The impact of the relevance threshold on TREC RTS 2017 dataset. The horizontal red line indicates the score of an empty run.

Figure 5.6 shows that the best results in terms of both metrics (ELG, nCG) are achieved with the same threshold value (GT = 0.6). In TREC 2016 dataset (Figure 5.7), we find that the best performance in terms of EG is obtained with the threshold value equal to 0.7 whereas the best score in terms of nCG is achieved by the threshold value equal to 0.55. We observe a similar trend for TREC 2017 dataset as shown in Figure 5.8, where different values of threshold yield to the highest performance in terms of EG and nCG. We find that our approach obtains higher nCG than EG on TREC 2016 and 2017 dataset for low relevance threshold. These results were expected because a low threshold value yields to increase the number of delivered tweets. The system that pushes a high volume of tweets gets a high nCG since the nCG is a recall-like metric. What the system with low relevance threshold lacks in terms of quality of individual tweets it is made up in volume, leading to higher nCG than its EG scores.

We do observe that the performances of our approach overpass the empty run baseline in both metrics (EG, nCG) on TREC 2017 dataset overall relevance threshold values but failed to beat this baseline on TREC 2015 and 2016 dataset for low relevance threshold ($<= 0.4$). These results can be explained by the fact that there are more silent days in TREC 2015 and 2016 than in TREC 2017 dataset as it is shown in Table 5.5. For these days no relevant tweets occur and the system should push nothing in this case. On the one hand, the score of an empty run increase with the number of silent days because systems that recognizing that there are no relevant tweets for a particular day and remain silent are rewarded. On the other hand, a system with low relevance threshold is penalized for being "gossipy" and pushing tweets in a silent day which causes degradation on overall performances.

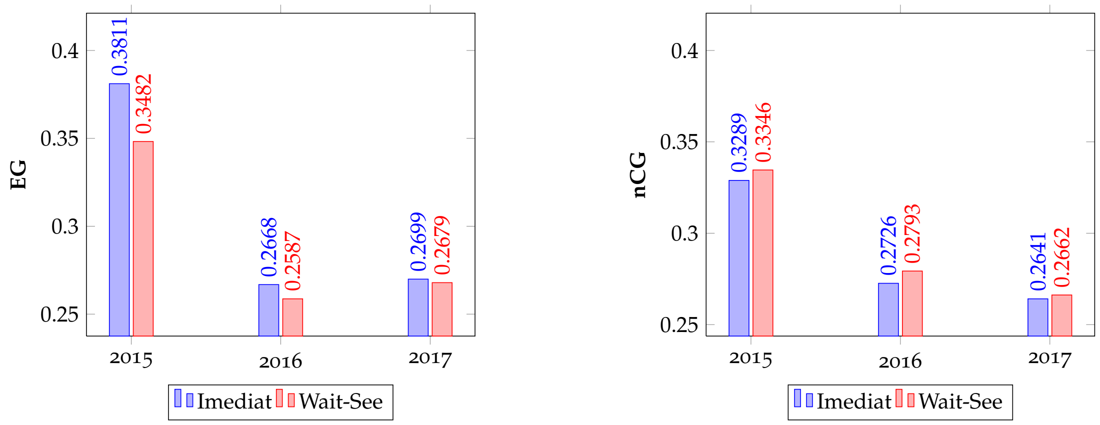
Table 5.5: Silent vs event full days in TREC 2015,2016 and 2017 datasets.

Corpus	TOPICS	Silent days	Full days	Total	Ratio of Silent Day
TREC 2015	51	126	384	510	24.70%
TREC 2016	56	173	387	560	30.89%
TREC 2017	99	137	639	776	17.64%

These experiments highlight the importance of effective threshold setting and demonstrate that the effectiveness of tweet stream filtering relies greatly on identifying the appropriate relevance threshold for pushing updates.

The best performance on TREC 2015 is obtained with the global threshold $GT = 0.6$, which we retain for the remainder of this chapter.

5.4.3.2 *Effectiveness of immediate pushing strategy*



(a) Performances in terms of EG

(b) Performances in terms of nCG

Figure 5.9: Instantly pushing strategy VS high-latency pushing strategy.

In this subsection, we compare the instantly pushing strategy against the high-latency pushing strategy baseline (Wait-See) in which tweets are pushed at the end of the day. Figure 5.9 shows the results in terms of expected gain (EG) and normalized cumulative gain (nCG). Notice here that these metrics are latency-independent. Interestingly, we observe that immediate pushing strategy achieves higher expected gain than high-latency pushing strategy. This result makes sense because expected gain (EG) reward systems for identifying silent day and "stay silent", which is important if we want to shield users from being bombarded with notifications. We find that the high-latency pushing strategy shows better performances than immediate pushing strategy in terms of normalized cumulative gain (nCG). This result was expected since EG and nCG metrics are quite similar to precision and recall respectively and in principle, systems make trade-off along these two metrics. Indeed, delaying the submission of tweets allows the system to better accumulate evidence and achieves higher recall. These observations thus merely provide support to hypothesis made in the introduction which states that "taking time to accumulate evidence before decision making may hurt the quality of the notification".

5.4.4 *Comparative evaluation with official TREC results*

In this section, we report the comparative effectiveness of the proposed approach based on WSEBM with the high performing models from TREC RTF 2015 and TREC RTS 2016 and 2017 Tracks. Notice that a briefly described of these approaches were presented in Table 3.4.

5.4.4.1 *Comparison with TREC MB-RTF 2015 official results*

We compare our approach (denoted by WSEBM for Word Similarity EBM) against the two high-performing official results from the TREC MB-RTF 2015 PKUICSTRunA2 [49] and UWaterlooATDK [149] and against the approach described in [91] in which authors improve their results (UWaterlooATDK) obtained in TREC 2015. In addition, in order to evaluate the impact of using word similarity as weighting technique, we compare our method with standard EBM. Table 5.6 reports the results in terms of ELG and nCG. Notice here that ELG metric handles redundancy and timeliness: a system only receives credit for returning a non-redundant tweet and credit decays linearly with the latency such that after 100 minutes, the system receives no credit even if the tweet was relevant.

Table 5.6: Comparative evaluation with state-of-the-art.

Method	ELG	nCG	%ELG
WSEBM	**0.3811**	**0.3289**	
EBM	0.2583†	0.2544†	+32.22%
Tan et all[91]	0.3678	-	+3.48%
TREC MB RTF 2015 official Results			
PKUICSTRunA2	0.3175†	0.3127	+16.68%
UWaterlooATDK	0.3150†	0.2679†	+17.34%

Note. % indicates improvements in terms of ELG. Symbol † denotes the Student test significance of with $p - value \leq 0.01$.

As shown in this table, WSEBM outperforms all baselines overall metrics. We found performance improvements up to ELG values of about 16% for the best run in TREC

MB 2015 task and of about 3.4% for the approach based on feedback strategy to set the relevance threshold.

5.4.4.2 *Comparison with TREC RTS 2016 official results*

We compare our results with the high-performing official scenario "A" runs from the TREC RTS 2016 and the TREC baseline approach which corresponds to the best performing system from the TREC 2015 Microblog Track [85]. Recall that in this track, systems outputs were assessed using two different methodologies. The first one was the online in-situ evaluation based on judgments made by the mobile assessors. This assessment was carried out during the evaluation period as systems generate output. The second one was the batch evaluation methodology in which pushed tweets were assessed by NIST assessors via pooling after the live evaluation period ended. Notice that the official results in TREC 2016 track reveal that the first and the third best runs are based on the same approach and were tagged as manual which means that they used a human involvement before or during the evaluation period [86]. Unfortunately, we don't find any description of these runs because the team who submitted them did not share their notebook in the TREC conference proceeding. Hence, the second best run [122] turns out to be the high-performing automatic run in this track.

Table 5.7: Comparison with the official TREC 2016 RTS track results.

	in-situ evaluation metrics					**Batch evaluation metrics**					
Method	**Rel**	**Red**	**Not Rel**	P_s	P_l	**EG1**	**EG0**	**nCG1**	**nCG0**	**M**	**L**
Our approach	131	2	103	0.5550	0.5635	0.2668	**0.0589**	0.2725	0.0672	80	660
TREC RTS 2016 official Results											
COMP2016 run3-13	193	4	141	**0.5710**	**0.5828**	**0.2698**	0.0483	**0.2909**	**0.0695**	24	443
QUBaseline-37[122]	56	3	108	0.3353	0.3533	0.2643	0.0321*	0.2479*	0.0157†	62478	169
COMP2016 run1-11	54	1	38	0.51	0.5238	0.2565	0.0244*	0.2515*	0.0194†	7545	128
Performance of TREC RTF 2015 best run in 2016 track											
WaterlooBaseline-50	148	12	286	0.3318	0.3587	0.2289*	0.0253*	0.2330†	0.0295†	8718	576

Note. First three columns show the number of tweets that were judged relevant, redundant, and not relevant. The second to last column shows the median latency with respect to the first tweet in each cluster. The final column shows the length of each run, defined as the number of pushed tweets that were assessed. The official TREC runs rows are sorted by EG-1. The symbols *, †, and ‡ denote the Student test significance of the proposed approach improvement: $* \; 0.01 < p-value \leq 0.05$, $\dagger \; p-value \leq 0.01$, $\ddagger \; 0.05 < p-value \leq 0.1$.

Results of the in-situ evaluation by the mobile assessors are shown in the first five columns of Table 5.7. In this evaluation, systems performances are measured in terms of two variants of precision metric namely strict and lenient precision (P_s and P_l). In P_s systems don't get credit for redundant posts while in P_l redundant posts are considered as relevant. We note that systems vary widely in the volume of pushed tweets. Our approach pushes more tweets than others runs. The performance of our approach in terms of precision metric falls in the middle between the first and the third best runs which get the highest values of "strict" and "lenient" precision. We observe that our approach outperforms the best automatic run [122] in terms of both precision metric.

We found performance improvements up to P_s and P_l values of about 65.54% and 59.51% respectively.

Regarding the quality of notification, results of the batch evaluation by NIST assessors are shown in the second part of Table 5.7. The columns (6-7-8-9) list the metrics used to evaluate the quality of notification namely EG-1, EG-0, nCG-1, and nCG-0. We can see from results shown in Table 5.7 in terms of normalized cumulative gain (nCG) and the expected gain (EG) that performances of our approach are rather promising for the following reasons:

- The performance in terms of EG-1 and nCG-1 compared with the best TREC run are not improved but we notice that our approach has higher EG-0 with an improvement up to 21.94% while the performance in terms of EG-1 is slightly reduced of about 2.08%. This fact holds despite the fact that the best TREC run is manual while our approach is automatic.
- Our approach outperforms the best automatic run [122] overall metrics. We observe that the improvement in performance of our approach with respect to the performance of the second best run in terms of EG-0 and nCG-0 are more important than the improvement in performance in terms of EG-1 and nCG-1.

These results can be explained by the fact that, on the one hand, our approach pushes more tweets than others runs and, on the other hand, TREC RTS 2016 collection is characterized by the high number of the silent days as shown in Table 5.5. Recall that in EG-1 nCG-1, systems are rewarded for "staying quiet" in silent days whereas in EG-0 and nCG-0 all systems receive a gain of zero no matter what they do. Therefore systems that push more relevant tweets tend to score higher in terms of EG-0 and nCG-0 and may actually score lower in terms of EG-1 and nCG-1 than systems that pushed fewer relevant tweets.

For the timeliness of systems output, we report in the second to last column (denoted by M) the median latency (in second) which capture the timeliness of the system output. Note that latency is computed with respect to the first tweet in each cluster, and thus a system may have a high latency even if it submits a tweet immediately after it is identified. We notice that the median latency of our approach (80s) is very small compared to the median latency of the best automatic run in TREC RTS 2016 track (62478). This result shows that our approach trade off summary quality with latency and hence produces good quality output at the cost of low latency. This result is not surprising since the decision to select/ignore an incoming tweet is made in real-time as soon as a tweet is available. In [122] the submission of tweets is delayed until the system overtakes a predefined silence period or it has already identified a certain number of candidate tweets.

Interestingly, we note that the best performing run from TREC 2015 track [91] did not perform well in TREC 2016 track in both evaluation methodologies. In the batch evaluation, this baseline failed to defeat the empty run. In the online in-situ evaluation, it falls in the middle of the pack among 42 runs. In contrast, our approach performs well on TREC 2015 track dataset as shown in the previous subsection Section 5.4.4.1 and achieves nearly the same performance than the high-performing run in TREC 2016 track and it outperforms the best automatic run.

These results confirm that our proposed model which are based on word similarity and a dynamic relevance threshold setting is competitive on the first hand to identify relevant posts that convey novel information and on the second hand to push a timely notification.

5.4.4.3 *Comparison with TREC RTS 2017 official results*

In this subsection, we compare the results obtained by our approach against the official runs of this track [87]. Similar to the previous year track (TREC RTS 2016), the participants' systems were evaluated following the batch and the online in-situ evaluation methodologies.

We recall here that for the batch evaluation, the same two main metrics were used: the expected gain (EG), the normalized cumulative gain (nCG). However, EG-0 and nCG-0 variants of the metrics were replaced by new variants EGp and nCGp (p for proportional) [87]. The reasons for this is that Roegiest et al. [129] showed that EG-0 and nCG-0 are flawed metrics because they correlate with the volume of pushed tweets. The EGp and nCGp variant were introduced to handle the issue of the 0-1 discontinuity on silent days. In these metrics, a linear penalty is applied according to the number of tweets submitted on a silent day. For the online in-situ evaluation, in addition to the precision metric, the online utility is computed from live user judgments.

Table 5.8: Comparison with the official TREC 2017 RTS track results.

	in-situ evaluation metrics				Batch evaluation metrics					
Method	P_s	P_l	$Util_s$	$Util_l$	**EGp**	**EG1**	**nCGp**	**nCG1**	**M**	**L**
Our approach	0.3984	0.4734	-791	-207	0.3030	**0.2668**	**0.2992**	**0.2630**	1	1201
TREC RTS 2017 official Results										
HLJIT testRun2-07[61]	0.3784	0.4446	-654	-298	**0.3630**	0.2088†	0.2808	0.1266†	56744	621
HLJIT testRun1-06[61]	0.3389	0.4082	-805	-459	0.3318	0.1811†	0.2610∗	0.1102†	49154	618
UDInfoBL-run2-34	0.3980	0.4708	-342	-98	0.3226	0.2622	0.2489∗	0.1886†	55781	452
IRIT-Run1-14[66]	**0.4200**	**0.4814**	-198	-46	0.2918	0.2571‡	0.2321†	0.1974†	1	320
Performance of TREC RTS 2016 best run in 2017 track										
QUBaseline[122]	0.3785	0.4386	-562	-284	0.2422∗	0.2146†	0.2260†	0.1984†	1	446

Note: First four columns show the evaluation results by the mobile assessors and the following columns report the evaluation by NIST assessors. The column marked "M" shows the median latency with respect to the first tweet in each cluster. The last column shows the length of each run, defined as the number of pushed tweets that were assessed. The official TREC runs rows are sorted by EGp. The symbols *, †, and ‡ denote the Student test significance of the proposed approach improvement: $* \; 0.01 < p - value \leq 0.05$, † $p - value \leq 0.01$, ‡ $0.05 < p - value \leq 0.1$.

Results of the in situ evaluation by the mobile assessors are shown in the four first columns of Table 5.8. We see that systems vary widely in the volume of pushed tweets. Our approach submits the highest volume of tweets. In terms of quality of pushed tweets, we note that our approach obtains higher online precision than the best official run while the performance in terms of online utility is lower. This is likely due to the fact that our approach pushes 1201 tweets which is about twice as much the number of tweets delivered by the best official run. In fact, the study conducted by Roegiest et, al [129] showed that, on the one hand, there is a strong negative correlation between tweet volume and utility, and in the other hand, there is no correlation between online precision and online utility because systems with the same online precision can vary widely in push volume and hence have different utility. Hence, we believe that from the user perspective, the output with the highest volume is more informative than the shorter one with the same precision.

Regarding the batch evaluation, results are shown in the second party of Table 5.8. First, we note that our run falls in the fourth position in terms of the official track metric (EGp). Our approach outperforms the fourth official best runs overall metrics and overpasses the three high-performing official runs in terms of EG1, nCG1, and nCGp metrics. The improvement of performance in terms of nCGp and nCG1, which are a recall like metrics, makes sense since our approach submits more tweets than other runs. Notice that the fourth best run adopts a basic strategy that consists of pushing at most only one tweet per topic per day which explains that our approach outperforms this run. We observe that our approach achieves the highest score in terms of EG1 under which system is penalized by receiving a score of zero for not remaining silent and pushes at least one tweet on a silent day. It presents an improvement of 27.77% compared to the EG1 score of the best official. Despite that, we notice that our approach failed to improve performance in terms of EGp metric under which the penalty for not remaining "quiet" on a silent day is proportional to the number of delivered tweets up to the limit of ten tweets per day. This result suggests that what our approach lacks in the identification of silent day, it makes up in the number of relevant tweets pushed on "eventful days", leading to higher EG1 than EGp scores.

For the timeliness of the system output, we observe that our system have very low latency compared to the three best runs. The three high-performing official runs exhibit much higher latency. This might explain the high score in terms of EGp achieved by these runs. Delaying the submission of tweets gives the opportunity to systems to accumulate evidence that become available only after a while of the publication tweets. These results reveal that these runs trade latency for higher quality whereas our approach achieves a good balance between latency and quality of notification.

In this track, we observe a similar trend than in track of 2016. We note that the best performing run from the previous year (TREC RTS 2016) did not perform well in TREC RTS 2017 in terms of batch evaluation and it falls in the middle of the pack. This suggests that a progress has been made in the task of real-time filtering of social media stream since the first track (TREC RTF 2015). Interestingly, we see that our approach achieves a good performance in TREC RTS 2017 as well as in the previous two tracks. This result confirms the robustness of the proposed model to identify relevant and novel tweets and to find a trade-off between output quality and timeliness. Our approach shows that it produces a good quality output at the cost of low latency.

5.5 Conclusion

In this chapter, we introduced a novel approach for prospective notification in social media stream. We show that word similarity matching based on word embedding model and a simple thresholding strategy achieves good results in terms of quality and the timeliness. The experiments thereby underlined that leveraging the semantic relationships between terms allows improving the identification of relevant posts. The proposed relevance function enables the use of simple threshold across all topics. We showed that the threshold setting strategy has a great impact on the quality of notification and better results can be achieved if the threshold is appropriately set.

Finally, we presented in this chapter a comparison of the performance of the proposed approach against the official results in the last recent TREC microblog summarization track. The obtained results clearly show the robustness of the proposed method which managed to achieve a good performance overall TREC microblog summarization tracks with low cost of latency.

In accordance to one of the main subjects discussed in this chapter, namely the identification of salient posts in social media stream and the impact of relevance threshold value in filtering tweet stream, we will propose in the next chapter a learning to filter model based on machine learning technique to counter the issue of relevance threshold setting.

Furthermore, and since tweets provide additional information other than text which are related to their social context, in the next chapter, we will investigate the impact of considering social signals in real-time tweet filtering.

6 LEARNING TO FILTER TWEET STREAM

6.1 Introduction

This chapter is mainly about the selection of relevant posts in the social media stream. To identify relevant tweets in a timely fashion, it is common to rely on a threshold-based filter. To shield users from unwanted notifications, systems attempt to find a trade-off between pushing too many or too few tweets. In the case of a high threshold value, a system may miss pushing interesting content to the user. Conversely, in the case of a low threshold value, a system may overwhelm the user with irrelevant tweets. However, as shown in the previous chapter, it is difficult to properly and effectively set the relevance threshold value.

A considerable attention has been paid on threshold setting strategies [91, 49, 33, 122, 34]. Hence, many strategies for setting the relevance threshold were examined. Among them we can mention a single static threshold for all topics [91, 34, 122], an adaptive threshold set manually[49] , and a dynamic threshold set using statistics [33] or set using relevance feedback [91].

To overcome the issue related to relevance threshold setting, we propose, in this chapter, a learn to filter approach based on machine learning to build a binary classifier that predicts the relevance of an incoming tweet with respect to the topic of interest. In addition, we explore and evaluate an adaptive learning strategy in which the live user feedback is used to periodically update the classifier. This allows investigating the gain that can be achieved by taking advantage of an ongoing relevance feedback which is generated by users as tweets are pushed.

It is recognized that features based on social signals are important for relevance[152, 45, 23, 24, 39, 91]. Prior work addressing tweet retrieval consider that tweet relevance depends, on the one hand, on the importance (popularity) of corresponding authors in the social network and, on the other hand, on the content quality such as the occurrence of URLs, mentions, and hashtags. In these work, the proposed features are combined using a Bayesian network model [23, 24], clustering-based approaches or learning to rank methods [45, 109]. These previous work, consider social features that are not available with the coming tweet and/or require the gathering of supplementary information from the social network. These features include for instance the authority and the popularity of the user that post or retweet the incoming tweet. The former is based on how much previous tweets that were posted by the author of the incoming tweet were retweeted [45] while the later is evaluated by computing the popularity score using the PageRank algorithm [116] based on retweet relations. The use of such features is not possible in real-time tweet filtering scenario in which we are limited to use social features that are provided in the tweet itself.

While considerable work has been done in leveraging social features as additional relevance factor in ad-hoc information retrieval, there is still a lack of studies that analyze how effective is the use of social signals in real-time tweet filtering. A key limitation of many social features used in the literature is that they are not suitable for real-time filtering because they either require several API calls for crawling the required information (e. g.the popularity in the social network) or are not yet available (e. g.the number of time a tweet has been retweeted).

To address this issue, we study the impact of taking into account social signals and some non-content features (e. g.the presence of URLs) that are provided in the tweet itself. This is achieved by using a supervised learning based approaches which allows combining social feature with query dependent features. We proposed and evaluated a set of social and other non-content features suitable for real-time tweet filtering. We distinguish two classes of features. The first one consists of tweet specific features that include particular characteristics of tweets, such as the presence of URLs and whether it is a reply to another tweet or a retweet. The second class consists of user account features which refer to the activity and the influence of the author of the post on the social network. To fit the real-time filtering scenario, our method leverages only the available and accessible features in the tweet without retrieving any further information from Twitter's servers. We argue that considering social features enhances the effectiveness of the relevance filter.

The main contributions of this chapter are:

1. To overcome the threshold setting issue, we propose a tweet filtering approach based on machine learning that considers social signals as well as query-dependent features;
2. We propose a set of social features suitable for real-time tweet filtering;
3. We study the impact of social signals in real-time tweet filtering;
4. We show the gain that can be achieved by an adaptive learning strategy that takes advantage of ongoing users assessments.

We next introduce an overview of a learning to filter approach then we describe how we take advantage of the ongoing relevance feedback in an adaptive learning strategy. After that, we focus on different features that we consider to predict in real-time the relevance of an incoming tweet with respect to a given topic. We present an experimental setup followed with results of experiments carried out on TREC RTF 2015 [85], TREC RTS 2016 [86] and, 2017 [87] datasets. We end this chapter with conclusions.

6.2 Tweet Filtering

As previously discussed, to meet requirements of prospective information needs, notifications should be relevant (on topic), timely (provide updates as soon as the event occurs), and novel (avoid pushing multiple tweets that convey the same information). The aforementioned requirements are fulfilled by our approach as follows:

- To reduce the latency between notification time and publication time, the decision of selecting/ignoring an incoming tweet is taken immediately in real-time as soon as a tweet is crawled;
- To enhance the relevance and novelty of pushed tweets, the proposed method relies on three consecutive filters namely: a tweet quality filter, a relevance filter

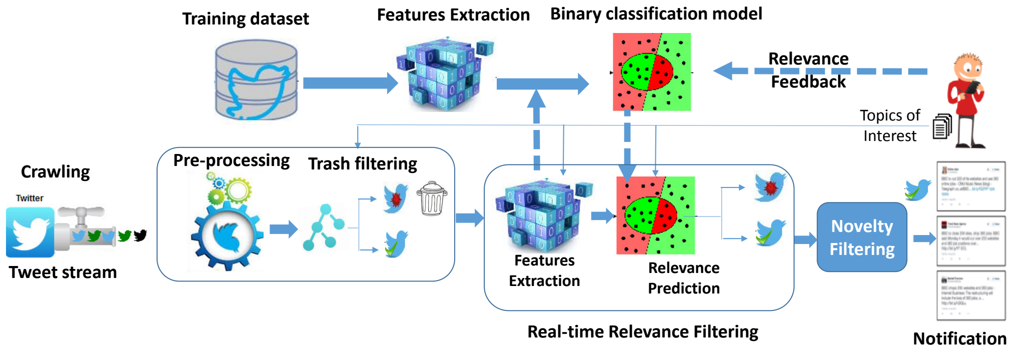


Figure 6.1: Overview of the adaptive learning strategy for real-time tweet filtering.

based on a binary classifier that takes advantage of the ongoing user relevance feedback and a novelty filter.

Figure 6.1 depicts an overview of our approach that monitors the continuous stream of tweets and fetches user relevance judgment to update the binary classifier. In this approach, the tweet quality and novelty filters are practically the same as the one described in the previous chapter. The incoming tweets are preprocessed and the "trash" tweets are discarded according to the rules introduced in Section 5.3.2.2.

For novelty detection, we notice that in the context of prospective tweet summarization, the novelty of a tweet is system-dependent. A tweet is considered novel for system S_i if it is the first tweet returned by S_i that conveys substantive information that is not present in the earlier ones. The same tweet can be considered novel for system S_i and redundant for system S_j. This implies that the idea of using a binary classifier that predicts the novelty of the incoming tweet can not be apply straightforward. Hence, we keep in this approach a threshold-based filter for novelty. The novelty score of the incoming tweet is computed as described in Section 5.2.3 and tweets having a novelty score lower than a predefined threshold are considered as redundant and hence they are ignored.

At a high level, the proposed approach differs from the ones introduced in Section 5.2.1 by replacing the threshold-based relevance filter that leverage only tweet text content with relevance filter based on supervised machine learning technique. This filter consists of a predictive model based on a binary classifier (relevant/not relevant) that predicts the relevance of the incoming tweet. The key advantage of adopting a learning to filter approach are:

- The use of the binary classifier trained on a given annotated dataset allows overcoming the relevance threshold setting issue;
- The use of machine learning method allows to easily combine different types of features in order to enhance the identification of relevant tweets. We exploit social signals as well as content features as evidence to identify relevant tweets in the social media stream. The social signals are query independent features whereas the content text features are query dependent;
- The relevance feedback can be used to update the binary classifier in order to further improve the effectiveness of the binary classifier.

We assume that users interact with notifications by providing relevance judgments that can be fetched by systems. The user can choose to judge each tweet as relevant, relevant but redundant (on topic, but contains information conveyed previously), or not relevant. The availability of an ongoing relevance feedback provides an opportunity to adjust the tweet filter using an adaptive learning strategy. This allows us to examine what gains could be achieved with live assessment feedback. Notice that such scenario was operationalized in the real-time summarization task (scenario A) at TREC 2017 [1].

6.2.1 *Learning to filter tweet in real-time*

The core task of relevance filtering is to determine whether an incoming tweet is relevant or not with respect to a user information need. Intuitively, supervised learning approaches can be applied in tweet filtering. The straightforward way is to consider the tweet filtering task as a binary classification problem in which each tweet is classified as either relevant or irrelevant. Tweets that are considered irrelevant are ignored and no longer considered. Hence, learning to filter tweets is a supervised learning method. It requires a training dataset consisting of queries which represent user interest and labeled tweets. After training data preparation, we need to select and extract features from the training corpus. Afterward, we have to figure out which learning algorithm to use. Indeed, there are several learning algorithms from the state-of-the-art that can be used to train a binary classifier as relevance filter, to mention: Naive Bayes [70], Random Forest [25] and J48 [121].

One of the most important tasks of a machine learning algorithm is the selection of features. The main challenge in this learning to filter approach is the feature design due to the real-time aspect of filtering tweets in social media stream which suggests that :

- Although social signals that correspond to the interaction of the social network users with a post such as the number of "*likes*" and "*retweets*" are probably one of the most interesting sources of evidence for relevance detection [19], these social signals are not available at the time of the decision making. Indeed, since filtering is carried out just after the publication of the incoming tweet, users have not yet reacted to these publications;

- To better fit a real-time filtering scenario, we cannot use features that require retrieving information from Twitter's servers such as the authority and the popularity of the author of the incoming tweet based on social network interaction [45]. We are limited to leverage the available simple meta-data about user accounts and tweets since gathering other features might be time-consuming due to Twitter's API limitations.

In Section 6.3, we introduce a design of "*light*" features suitable for real-time tweet filtering.

6.2.2 *Adaptive Learning strategy*

In adaptive learning, we assume that users interact with the prospective notification system by providing relevance judgments of pushed tweets. The system takes advantage

[1] http://trecrts.github.io/TREC2017-RTS-guidelines.html

of relevance feedback to re-train the classifier. To do so, the classifier is initialized with a training dataset and it is retrained periodically each times new relevance judgments have been made available. The system fetches the relevance judgment of users periodically and uses it to label the new instances that correspond to the features of the pushed tweets. These new labeled instances are added to the current training instances set and the model is retrained. We set this strategy in order to fit a real-world scenario in which the user may choose to judge the pushed tweet immediately or later (if it arrives at an inopportune time) or may choose to not do it.

One major drawback of updating the training instances set each time a new relevance judgment is made available is the risk to obtain an unbalanced training dataset from a class distribution point of view. In this case, a classifier tends to predict samples from the majority class which may correspond to irrelevant tweets. According to machine learning principles, performing learning has to be accomplished on a totally or almost balanced data-set. To ensure that this kind of situation does not happen, we control the number of relevant and irrelevant instances to be added to the training set. For each instance that corresponds to a new relevant tweet, only one instance corresponding to irrelevant tweet is added.

6.3 Features Design

In this section, we detail the different features that we consider to predict the relevance of an incoming tweet to a given topic. We present query dependent features that measure the relevance with respect to the topic and social signal features that do not consider the actual topic.

In the context of real-time tweet filtering, we are limited to use features that are already available in the meta-data of a tweet. This allows us to predict the relevance of the incoming tweet as soon as it is published. Hence, we are not able to use Twitter's REST APIs to collect further features such as the profiles of followers or to crawl external URL webpage text. Among the set of available features extracted from the tweet's meta-data, we used feature selection algorithms to determine the best relevance-dependent signals that can be effectively used in the tweet filtering task. We defined 22 features categorized into three classes: query dependent, tweet specific and user account features.

6.3.1 *Query dependent features*

As in the previous chapter, we assume that the user information need follows the standard TREC topic format which includes a title (Q^t) of the information need and a description (Q^d) that indicates what is and what is not relevant. To capture the relevance of a tweet's content, we used six query dependent features that measure the relevance of the given tweet text with respect to a topic. These features are as follows:

- $|(Q^t \cup Q^d) \cap Hashtag|$: The number of words overlaps between the query's terms and hashtags in the tweet. The rationale behind this feature is that the presence of a query term as a hashtag is a valuable signal of relevance since hashtags are used to draw attention and to label the content of a given tweet;

- The cosine similarity between the query title and the tweet's text vectors using a word embedding model (word2vec [106]). The vectors of the title of the query and the tweet's text are obtained by summing up all vectors of their words. This feature can be considered as a semantic-based relevance score which aims to leverage the probable semantic relationship between terms of the query and the tweet by taking advantage of a word embedding model;
- $RSV(T, Q^t)$: The relevance score of the incoming tweet with respect to the title Q^t. This feature is a retrieval score calculated according to Equation 5.1 of the proposed relevance estimation model introduced previously in Section 5.2.2;
- $RSV(T, Q^d)$: The relevance score of the incoming tweet with respect to the description Q^d. This feature is measured according to Equation 5.2 in Section 5.2.2;
- $|(Q^t \cap T|)$: The number of words that overlap between the text of the tweet and the query's title $|Q^t|$;
- $|(Q^d \cap T|)$: The number of words that overlap between the text of the tweet and the query's description $|Q^d|$.

Notice here that all the aforementioned features do require neither external knowledge nor stream statistic and can be estimated at the time a given tweet arrives.

6.3.2 *Tweet specific features*

These features describe elements that are mentioned in a tweet text and the nature of the tweet itself which can be a retweet of another tweet or a reply to an old tweet of another user. We leverage seven (07) tweet-specific features that are defined as follows:

1. **RateHashtag**: This feature is defined as the ratio of the number of hashtags and the number of tokens in the tweet. A hashtag is used to highlight a topic of a tweet and it is a way of making it easier for users to find, follow, and contribute to a conversation. Therefore, the presence of hashtag may be an indicator of relevance. However, due to the shortness of tweet, the presence of many hashtags may indicate the opposite. For that reason, we suggest using the ratio of the number of hashtags and the tweet's length instead of using a boolean value that indicates whether a tweet contains at least one hashtag or not as proposed by [152].
2. **HasURL**: [31] showed that people often exchange URLs via Twitter and considering this information improves the ranking of recently discussed URLs on Web search. Therefore, the presence of a URL can be considered as an indicator of relevance as suggested in [39].
3. **isReply**: Whether a tweet is a reply to another tweet. A reply tweet is a tweet sent in direct response to another tweet which can be a comment on a previous post or simply a chat with other users. For a user seeking news, this type of post appears to be less interesting. Therefore, we argue that this feature is a valuable signal for relevance detection. We suppose that reply tweets are less likely to be relevant than other tweets;

4. **TimeofPublication**: This feature describes at which hour during the day a tweet was published. We use the "*utc_offset*" property of the user object (which is returned with the tweet) to calculate the time relative to the user's timezone. The intuition behind this feature is that tweets posted at an inappropriate hour of the day (e. g.02:00 pm) are likely to be irrelevant;

5. **HasEntity**: This feature is a boolean property which indicates whether an entity (PERSON, ORGANIZATION, LOCATION) is mentioned in the tweet. For this, we use the Stanford Named Entity Recognizer [2]. An empirical study on relevance factors conducted by [152] shows that these three types of entities occur more often in relevant tweets than in irrelevant ones. Therefore, we assume that tweets in which at least one entity is mentioned are more likely to be relevant than tweets that do not contain any entity;

6. **NbUser**: This feature is defined by the number of user-names mentioned in the tweet. We assume that the more user-names are mentioned in a tweet, the more likely this tweet is irrelevant. The idea put forward in this hypothesis is that tweets in which many user-names appear are likely conversational and/or personal posts and hence are not very meaningful;

7. The length of the tweet: The number of tokens that a tweet's text contains after removing stop words, URLs and user-names. We argue that a longer tweet is more informative than a shorter one. For this reason, we consider the length of a tweet as an indicator of relevance.

6.3.3 *User account features*

In addition to tweet specific signals, we also investigate features that describe the author of the post in the social network. Note that in the case that the given tweet is a retweet, we consider the features of the user that published the original tweet, and not the one who retweeted it. The rationale behind the consideration of user features being that the importance of a tweet's content is related to the authority of the user who posts the tweet. The authority of a user can be captured through social features that are available in the meta-data of tweets. These features are time-sensitive. The importance of a signal depends on the account age. An old account may have much more followers than a recent one. Therefore, in user account features, we implicitly consider the age of the account (in days) at the time of the tweet publication. In this work we investigate the following user account features:

- Follower: Number of followers the account of the author of the given tweet currently has. This feature directly indicates the size of the audience for that user;

- Friend: The number of users the account of the author of the given tweet is following;

- Follower/day: Ratio of the number of user's followers and the age of the account. Considering the number of followers as evidence of relevance may promote old account since in general, they may have much more followers than a recent one.

[2] http://nlp.stanford.edu/software/CRF-NER.shtml

To cope with this issue, we propose to normalize the number of followers by the age of the account in days;

- Friend/day: Ratio of user friends and the age of the account. The number of friends is normalized by the age of the account for the same reasons aforementioned in the previous feature;
- Fol/Fr: Ratio of the numbers of followers and friends (followees) of the user. This ratio is a common metric to measure influence on Twitter [45, 79, 63]. It may communicate the intended purpose or practices of a user. If the ratio approaches 1 (means that the number of followers and friends are near-equal), the user account might be considered as conversationalist since in this case the user most likely follows back a majority of his followers. If this ratio approaches infinity (high number of followers with a low number of friends), the user might be considered as a valuable source of information. Finally, if the ratio approaches zero (a low number of followers and a high number of friends), the user account might be categorized as a spammer. Therefore, we hypothesize that tweets issued by users having a higher ratio of followers/friends are more likely to be relevant than tweets posted by users having a smaller ratio of followers/friends;
- List/day: Ratio of the number of lists a user appears in and the age of the account. Note that, users are allowed to classify their followings into several lists based on topics;
- $(List + Fol/Fr)/day$: A combination of followers/friends ration and the number of lists the user appears in;
- Tweet/day: Ratio of the number of tweets (including retweets) issued by the user and the age of the account. This feature indicates the activity of the author of the given tweet. We assume that tweets posted by a user who has regular and "reasonable" activity are more likely to be relevant than tweets published by users with irregular activity or those who publish a huge number of tweets each day;
- Verified: This feature is a boolean property that indicates, when true, that the user has a verified account.

6.4 Experimental Evaluation

In order to validate our approaches based on machine learning techniques and to show the impact of social features to filter tweets, we conduct a series of experiments on datasets of the TREC RTS 2016 [86] and 2017 [**?**] tracks. The main goals of these experiments are:

- To evaluate the effectiveness of the proposed features to predict the relevance of tweets for a given topic;
- To evaluate the impact of considering social signal features in tweet stream filtering. To do so, we compare the performance of the classifier based solely on query-dependent features (QF) against the classifier that leverages one class of

signal signals features (Tweet specific (TF) or user account (UF)) with the query dependent features as well as the classifier that combines all features (both classes of social signals and query-dependent features);

- To compare the effectiveness of the classifier with an adaptive learning strategy as well as the classifier without adaptive learning with those obtained in TREC RTS 2016 and 2017 tracks.

In these experiments, we aim to provide empirical evidence supporting the following hypotheses:

- (**H1**) Considering social signals increases the performance of learning to filter approach.
- (**H2**) The learning to filter approach outperforms the threshold-based filtering approach.
- (**H3**) Taking advantage of ongoing relevance feedback in the adaptive learning strategy improves the quality of predicting the relevancy of incoming tweets for a given topic.
- (**H4**) Learning to filter tweet stream in real-time is topic-independent.

In the remainder of this section, we first introduce the experimental setup. Then, we detail training dataset used to build the classifier that predicts the relevance of tweets.

6.4.1 *Experimental setup*

In order to build a binary classifier and to evaluate the effectiveness of the proposed features on identifying relevant tweets regardless of the novelty, we conduct a series of experiments on a subset of TREC RTF 2015 dataset. In particular, we evaluate the performance of three well-known supervised learning algorithms, namely Naive Bayes [70], Random Forest [25] and, J48 [121]. Accordingly, we choose the best performing learning algorithm to build the classifier. Then, we evaluate the impact of the proposed features for correctly classifying tweets. In these experiments, we rely on the machine learning toolkit weka [60]. To select features, we use an information gain algorithm implemented in Weka tool [60]. We used a 10-fold cross-validation algorithm to measure the performance of our features and the learning classification algorithms.

To evaluate the performance of the proposed approach in tweet real-time filtering, we carried out experiments that correspond to post hoc runs using a replay mechanism of scenario "A" over tweets captured during the evaluation period of TREC RTS 2016 and 2017 tracks. We recall here that in this scenario, a system is allowed to return a maximum of 10 tweets per day and per topic. We used the binary classifier trained on a subset of TREC RTF 2015 dataset (as described above). In these experiments, the fetching of relevance judgment, the novelty threshold value and the word embedding models were set as follows:

- **Fetching of relevance judgments:** Without direct users interaction, the usual approach to model an adaptive learning setting is to consider the relevance judgments set of the test collection. We follow this approach to simulate an ongoing relevance feedback used in the adaptive learning strategy. In our experiments, we use mobile assessor assessment from in-situ evaluation in which tweets submitted by participant systems were judged in an online fashion during the evaluation period as described in [129, 128]. We assume that relevance feedback is provided after each notification and available for immediate use. In the adaptive learning strategy, the system fetches the relevance judgment of pushed tweet periodically. Each time new relevance judgments have been made available, the system retrains the binary classifier after including new instances to the initial training data-set.

- **Novelty threshold:** In these experiments the novelty threshold value was set experimentally based on pilot experiments conducted on TREC 2015 data-set in Section 5.4.1. We set this threshold to 0.3 which corresponds to the value that gets the lowest detection cost as shown in Figure 5.2a.
- **Word embedding models:** The word embedding models (used in the evaluation of the relevance score of incoming tweets) are the same as the ones used in experiments conducted in the previous chapter as described in Section 5.3.2.1.

In the following subsection, we describe the dataset used to train the binary classifier.

6.4.2 *Binary classifier training dataset*

The binary classifier is built using a learning algorithm trained on a TREC 2015 RTF dataset as follows: we extract for each topic (51 topics) tweets from the judgment pool of TREC 2015 RTF dataset. We obtain 94,068 tweets, among them, 8,164 tweets were labeled as relevant. We notice that the classes of these sets are unbalanced. In this case, a classifier tends to predict samples from the majority class which corresponds to irrelevant tweets. To get a balanced training dataset, we filter out all tweets that do not contain at least two query terms. Thus, we obtain a training dataset that contains 6663 tweets in which the distribution of relevant and irrelevant tweet is 50.18% and 49.82% respectively. The trained classifier is then tested on TREC RTS 2016 and 2017 datasets.

Notice here that the topic set of the TREC RTS 2016 track contains a mix of topics from the TREC RTF 2015 (36 topics) and new topics (20) unseen in the judgment pool of TREC 2015 track. In TREC RTS 2017 track all topics (188) were new and were specifically developed from scratch for this year's track. In other words, the TREC RTS 2017 track concerns topics unseen in the training dataset and the TREC RTS 2016 includes topics partially covered in the training dataset. This fact allows investigating whether the learning to filter approach is topic-independent or topic-dependent. Also, we do observe that the size of the topic set (51) of the training data-set is smaller than the size of the topic set in the test datasets (56 and 188 topics in 2016 and 2017 tracks respectively).

6.5 Results and Discussion

In this section, we divided our results into two parts. In the first part, we present results that attempt to answer the following questions:

- Which one of the three learning algorithms (Naive Bayes [70], Random Forest [25] and J48 [121]) performs better in a tweet filtering task? Section 6.5.1 describes the results obtained by the aforementioned learning algorithms;
- What is the impact of each category of adopted features in enhancing the performance of the classifier? Section 6.5.2 reports the effectiveness of each category of proposed features.

In the second part, we compare the performance of the proposed approach based on the binary classifier with the approach proposed in the previous chapter (WSEBM) which relies on a threshold-based filter and we report the impact of considering social signals in real-time tweet filtering. Next, we describe the impact of the adaptive learning strategy denoted by the Adaptive Binary classifier (ABC) which is compared with the passive binary classifier (PBC). Finally, we compare our results with the high-performing official results from the TREC RTS 2016 and 2017 tracks.

Methods are labeled using the following convention: the first part indicates whether the binary classifier is adaptive (ABC) or not (PBC), the second part indicates which categories of features were used to predict the relevance of tweets (Query-dependent features: (QF), tweet specific features: (TF) and user account features: (UF)).

6.5.1 *Performance of different learning algorithms*

In this subsection, we compare three well-known supervised learning methods in terms of predicting the relevance of tweet with respect to user information need. These methods are Naive Bayes [70], Random Forest [25] and, J48 [121]. In this experiment, we used a 10-fold cross validation algorithm on TREC RTF 2015 dataset. To assess the effectiveness of the classification model, we adopt the standard existing information retrieval metrics of precision, recall, and F-measure. Note that in this experiment redundant post are considered as relevant.

We report in Table 6.1 the performance results of the aforementioned learning methods in terms of precision (P) recall (R) and F-measure. We notice that Random Forest has shown the best results overall evaluation metrics. To provide an in-depth understanding of the performance of the adopted learning methods, we plot, in Figure 6.2, the Receiver Operating Characteristic curves (ROC) showing the evolution of the performance of each classifier. This curve shows the trade-off between the probability of identifying relevant tweets as relevant (true positive rate) and the probability of considering an irrelevant post as relevant (false positive rate). On the x-axis is the false positive rate and on the y-axis is the true positive rate. The area under a ROC curve quantifies the overall ability of the classifier to correctly identify relevant and irrelevant tweets. A classifier is considered to perform best when it has its curve towards the top-left of the graph. The closer the curve follows the left-hand border, the more accurate the classifier.

Results reported in Figure 6.2 reveal that Random Forest has a higher accuracy than the J48 and Naive Bayes learning algorithms. These results motivated our choice to adopt the Random Forest learning method in our approach.

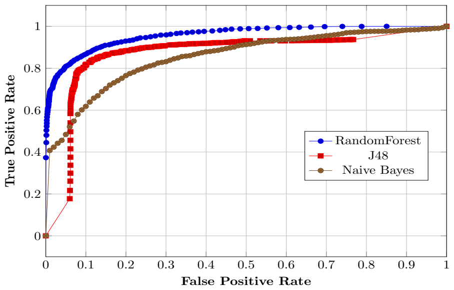


Figure 6.2: Roc curve for different binary classifier.

Classifier	P	R	F-Measure
RF	**0.908**	**0.903**	**0.905**
J48	0,861	0,861	0.861
NB	0.749	0.693	0.678

Table 6.1: Comparison of performance of different binary classifier on TREC 2015 dataset.

6.5.2 *Effectiveness of different categories of features*

To assess the effectiveness of our proposed features, we used a 10-fold cross-validation on the TREC RTF 2015 data-set and we measure the accuracy together with the standard information retrieval metrics, namely precision, recall, and F-measure. These metrics were computed for the relevant class only since our main purpose is relevant tweet detection. We compare the performance of the binary classifier that considers all features (query-dependent, (QF), tweet specific features (TF) and user account feature(UF) against degenerate versions of our model. The degenerate versions are defined to evaluate the impact of each class of features as follows:

- In QDF, only query-depended features are used. This version is considered as baseline and allows us to show the impact of leveraging social signals in tweet stream filtering task.
- QDF-UF takes into account the query-dependent features (QDF) as well as user account features. Comparing our model with this version allows to evaluate the impact of user account features independently of tweet specific features;
- QDF-TF combines query-dependent features with tweet specific features. This version is defined to study the impact of considering tweet specific features.

As shown in Figure 6.3, the use of social signal features improves the quality of the classifier overall metrics. We claim that considering social signals improves the ability of the classifier to correctly identify relevant tweets. The results plot on Figure 6.3 support this hypothesis (H1). The performance improvements of considering social signals compared to query-dependent features are about 4.12%, 14.44%, and 14.46% in terms of the precision, the recall and the accuracy respectively.

The comparison of the social signal features also reveals some differences between tweet specific and user account features. Overall, it appears that learning the classification model solely based on query and tweet specific features leads to achieve a higher accuracy than the model based on query and user account features. The use of tweet specific features allows enhancing the precision whereas considering user account features improves the recall of the classifier. Note that the impact of tweet specific features is in line with the finding made by [152, 39] who showed that the length, URL, and the replies based features are a valuable indicator of the relevance of tweets.

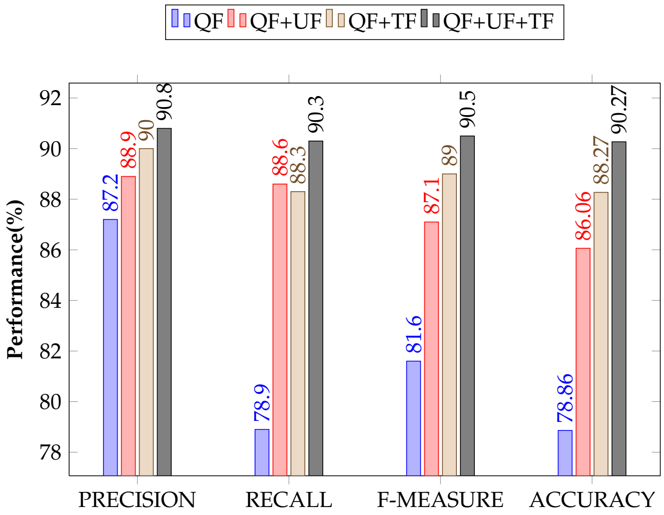


Figure 6.3: Performance of our features on TREC RTF 2015 dataset using different evaluation metrics.

6.5.3 *Impact of social signals in real-time tweet filtering*

In order to examine the performance of the learning to filter approach that considers social signal features in real-time tweet stream filtering, we compare results obtained when all features are combined (PBC(QF+TF+UF)) against:

- The three degenerate versions of the classification model (PBC(QF), PBC(QF+TF) and PBC(QF+UF)) as defined in the previous subsection. The comparison with these degenerate versions allows on the one hand to verify hypothesis (H1) and on the other hand to evaluate the effect of each category of social signal features;
- The approach proposed in the previous chapter (WSEBM)[34] in which the relevance filter is threshold-based. This approach is considered as a baseline and enables to verify hypothesis (H2).

We recall that the classifier is trained using TREC RTF 2015 dataset. Table 6.2 reports performances in terms of the in-situ and the batch evaluation metrics on TREC RTS 21016 and 2017 dataset. In the in-situ evaluation, the performances are measured in terms of lenient and strict precision (P_s and P_l) while performances in the batch evaluation are computed in terms of gain-oriented metrics namely, the normalized cumulative gain (nCG) and the expected gain (EG).

From Table 6.2, we can see that the best performance is achieved when all features are combined. The use of a classification model based on social signals and query-dependent features (PBC(QF+TF+UF)) improves the performance of real-time tweet filtering compared to the approach that relies on a threshold-based filter (WSEBM) in terms of the batch evaluation on the TREC RTS 2016 as well as on the TREC RTS 2017 dataset. The leaning to filter approach (PBC(QF+TF+UF)) outperforms the WSEBM model with an improvement of the EG-1 and nCG-1 metrics of about 7.50% and 5.68% respectively on the TREC RTS 2016 dataset and of about 3.78% and 4.06% on the TREC RTS 2017 dataset. It is interesting to observe that classification model based on social

Table 6.2: Performance of social signals in real-time tweet filtering using TREC RTS 2016 and 2017 datasets.

	In-situ evaluation metrics				**Batch evaluation metrics**					
	2016		**2017**		**2016**			**2017**		
Method	P_s	P_l	P_s	P_l	**EG1**	**nCG1**	**L**	**EG1**	**nCG1**	**L**
PBC-QF-UF-TF	0.4903	0.5096	0.3771	0.4451	**0.2793**	**0.2828**	436	**0.2769**	**0.2737**	1117
PBC-QF-UF	0.4911	0.5088	0.3698	0.4337	0.2705‡	0.2750‡	445	0.2720	0.2697	1147
PBC-QF-TF	0.4765	0.4966	0.3659	0.4316	0.2647∗	0.2701∗	420	0.2667	0.2551∗	1030
PBC-QF	0.4662	0.4932	0.3577	0.4197	0.2613∗	0.2641∗	423	0.2562∗	0.2450†	1025
WSEBM	**0.5550**	**0.5635**	**0.3984**	**0.4734**	0.2668‡	0.2725‡	662	0.2668	0.2630	1201

Note: The first four columns show the results of the in-situ evaluation conducted by mobile assessors and the following columns report the evaluation by NIST assessors. The column marked "L" shows the length of each run, defined as the number of pushed tweets that were assessed. The symbols *, †, and ‡ denote the Student test significance of adaptive learning improvement: $* \ 0.01 < p - value \leq 0.05$, † $p - value \leq 0.01$, ‡ $0.05 < p - value \leq 0.1$.

signals improves both EG-1 and nCG-1 since these metrics are quite similar to precision and recall and in general systems attempt to make a trade-off between them.

Interestingly, although the proposed learning model improves performance in terms of the bath evaluation (EG1 and nCG1 metrics), we observe that it failed to beat threshold-based model in terms of the in-situ evaluation metrics (P_l, P_s) in which assessors provided judgments online while systems pushed tweets during the evaluation period. These results can be explained by the volume of pushed tweets and the number of silent days in which the system breaks the silence. A study conducted by [129] on the in-situ evaluation metrics reveals that there is a little correlation between the online precision and the volume of pushed tweets. For a system that pushes a high volume of tweets, what it lacks in terms of the quality of tweets, it makes up in volume. We note that (PBC(QF+TF+UF)) pushes 436 and 1117 tweets on TREC 2016 and 2017 datasets respectively whereas WSEBM pushes more tweets (662 and 1201 tweets on TREC 2016 and 2017 datasets respectively). Regarding breaking the silence in silent days, a system is penalized by receiving a score of 0 for such days otherwise it is rewarded by receiving a perfect score 1 for identifying a silent day. We observe that on TREC 2016 dataset that includes 173 silent days, (PBC(QF+TF+UF)) breaks silences on 10 days while WSEBM breaks silences on 30 days. A similar trend is observed on TREC 2017 dataset that contains 137 silent days, (PBC(QF+TF+UF)) breaks silences on 32 days while WSEBM breaks silences on 37 days. This observation thus merely provides support to hypothesis H1.

Comparing the two classes of social signals, there is thus no notable difference between the two categories of social signal features. However, we notice that the tweet features contribute to enhancing performance in terms of EG-1 whereas considering the user account features yield to improve performance in terms of nCG-1. By combining both features in (PBC(QF+TF+UF)), the expected gain (EG1) increases which is caused by a higher recall (nCG1). It appears that the user account and tweet specific features complement each other. These results reveal that the user account features allows identifying more relevant information whereas the use of tweet specific features yields to recognize that there is no relevant information and stay silent. User account features can be interpreted as a means of evaluating the "reliability" of the information source and tweet specific features can be considered as an assessment of the information quality.

6.5.4 *Impact of adaptive learning strategy*

In this subsection, we explore the effectiveness of taking advantage of the ongoing relevance feedback to improve the ability to identify relevant tweets. Recall that first the classifier was trained using TREC RTF 2015 dataset then during the filtering process, relevance feedback is used to re-train the classifier. In Table 6.3, we compare the performance of adaptive learning classification model (ABC) against a passive learning classification model (PBC) in terms of the in-situ as well as the batch evaluation metrics on both the TREC RTS 2016 and 2017 dataset tracks.

Note that in the experiment carried out on the TREC RTS 2016 dataset, our approach (ABC) pushed 480 tweets and among them, only 140 were judged by assessors. So the classifier was re-trained using the 140 available relevance assessment. In the experiment conducted on TREC RTS 2017, the system pushed 3182 tweets among them 2138 were judged by assessors. Hence, the classification model was retrained incrementally with 2138 new annotated instances. The training of the model is fast (a few seconds on a workstation with 3 cores, 1.8 GHz, and 8 GB of RAM). Therefore, we decided to add the annotated instances to the retain set and to retrain the classification model with every 10 freshly labeled instances (which correspond to a pushed tweets assessed by mobile assessor).

Table 6.3: Adaptive learning VS passive learning performances on TREC RTS 2016 and 2017 datasets.

	In-situ evaluation metrics						**Batch evaluation metrics**			
	2016			**2017**			**2016**		**2017**	
Method	P_s	P_l	**L**	P_s	P_l	**L**	**EG1**	**nCG1**	**EG1**	**nCG1**
ABC-QF-UF-TF	0.4931	0.5136	140	0.3931	0.4640	2138	**0.2989**	**0.2954**	**0.2956**	**0.2861**
PBC-QF-UF-TF	0.4903	0.5096	149	0.3771	0.4451	2123	0.2793‡	0.2828	0.2769‡	0.2737
% change	0.57%	0.78%		4.24%	4.24%		7.01%	4.45%	6.75%	4.53%

Note: The first four columns show results of in-situ evaluation conducted by mobile assessors and the following columns report the evaluation by NIST assessors. The column marked "L" shows the length of each run, defined as the number of pushed tweets that were judged by mobile assessors. The last row shows the adaptive learning improvement.

As we can see from Table 6.3, the adaptive learning strategy outperforms the passive learning classification model in which the initial model is not retrained overall in-situ and batch evaluation metrics on both tweets collection (TREC RTS 2016 and 2017). The results of this experiment support hypothesis (H3). It indicates that taking advantage of an ongoing assessment is very useful for identifying correctly relevant tweets for a given topic in the social media stream. We observe that on both datasets, the performance improvement in terms of precision-oriented metric (EG1) obtained by the adaptive classification model is higher than the improvement in terms of recall-oriented metrics (nCG1).

This result shows that adaptive learning using ongoing feedback has the potential to learn when to "stay silent" and to produce notification with significant gain. It is interesting to note that no significant improvement in terms of the in-situ evaluation was found between adaptive and passive learning model on TREC RTS 2016 dataset. This can be explained by the low number of tweets submitted by adaptive learning approach and were judged in TREC RTS 2016 which means that the classification model was retrained with relatively a small number of new instances.

To provide an in-depth understanding of the impact of taking advantage of an ongoing relevance feedback in real-time tweet filtering, we compare in Figure 6.4 the evolution of performance obtained when the adaptive learning (ABC) is used against the performance of the passive learning model (PBC) in terms of online "lenient" precision and utility metrics across each day of TREC RTS 2017 evaluation period. Note that, we adopt the "lenient" version of precision and utility metrics because in these measures system gets credit for redundant judgments and the purpose of this experiment is to examine the ability of correctly identifying relevant tweets regardless of the redundancy. We carried out this experiment on TREC RTS 2017 dataset because it includes more relevance judgment than TREC RTS 2016 dataset.

From Figure 6.4, we can see that in the first two days the impact of the use of relevance feedback is not significant. The adaptive learning strategy improves the "lenient" precision by 1.05% and 1.53% in the first two days respectively. At the beginning of the filtering process, there is thus no notable difference between the adaptive and passive learning models. However, after the third day, we do observe that the adaptive classification model is showing signs of improvements in performance in terms of both metrics ("lenient" precision and utility). This trend is confirmed until the last day of the evaluation period and this despite the fact that the performances of both approaches have been declined compared to the performance obtained at the beginning of the evaluation period. We notice that in the last day the adaptive learning outperforms passive learning by 15.28% in terms of "lenient" precision. This observation validates the hypothesis (H3). It also underlines the key importance of the amount of relevance feedback to enhance the quality of real-time relevance tweets filtering.

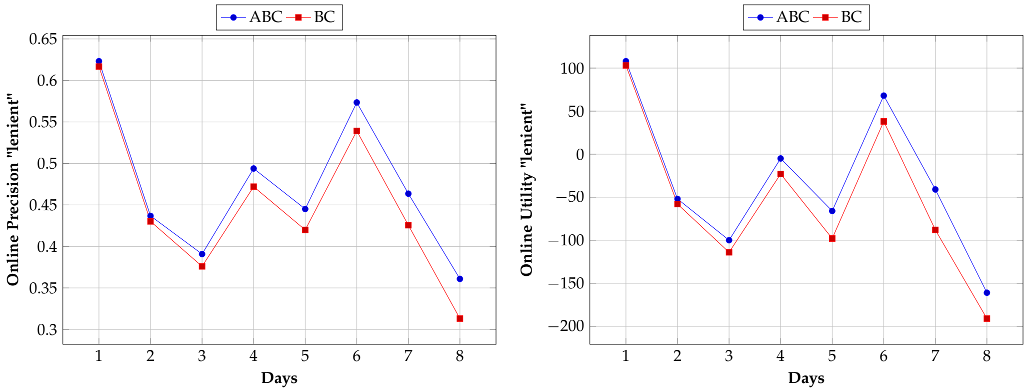
6.5.5 *Comparative evaluation with the official TREC results*

In this subsection, we compare the effectiveness of the proposed approach with the high performing models in terms of the batch and the in-situ evaluation methodologies from TREC RTS 2016 and 2017 Tracks. Notice that a brief description of these models was presented in Table 3.4.

6.5.5.1 *Comparison with the official results of TREC RTS 2016*

In Table 6.4, we compare our approach based on adaptive learning strategy with the best performing runs in TREC RTS 2016 tracks in terms of official metrics in-situ and batch evaluation.

Regarding batch evaluation, Table 6.4 shows that the adaptive learning method presented in this chapter achieves the best performance in terms of EG1 and nCG1 metrics, and the passive learning approach also gets the improvement compared to the high performing runs in TREC RTS 2016 track. We can observe that our method outperforms the



(a) Performances in terms of Online Precision "lenient"

(b) Performances in terms of Online Utility "lenient"

Figure 6.4: Comparison of performance in terms of in-situ evaluation between adaptive learning strategy (ABC) and a passive binary classifier (PBC) on TREC RTS 2017 dataset.

Table 6.4: Comparison with the official TREC 2016 RTS track results.

	In-situ evaluation metrics					Batch evaluation metrics					
Method	**Rel**	**Red**	**Not Rel**	P_s	P_l	**EG1**	**EG0**	**nCG1**	**nCG0**	**ML**	**L**
ABC-QF-UF-TF	72	3	71	0.4931	0.5136	**0.2989**	**0.0561**	**0.2954**	0.0525	39	375
PBC-QF-UF-TF	131	2	103	0.5550	0.5635	0.2793‡	0.0493	0.2828	0.0524	80	436
TREC RTS 2016 official Results: Best run in terms of the batch evaluation											
COMP2016 run3-13[59]	193	4	141	0.5710	0.5828	0.2698*	0.0483	0.2909	**0.0695**	24	443
QUBaseline-37[122]	56	3	108	0.3353†	0.3533†	0.2643*	0.0321†	0.2479†	0.0157†	62478	169
COMP2016 run1-11	54	1	38	0.51	0.5238	0.2565†	0.0244†	0.2515†	0.0194†	7545	128
TREC RTS 2016 official Results: Best run in terms of in-situ evaluation											
COMP2016 run3-13	193	4	141	0.5710	0.5828	0.2698*	0.0483	0.2909	0.0695	24	443
COMP2016 run2-12	47	1	38	0.5465	0.5581	0.2559*	0.0220†	0.2483†	0.0143†	10055	169
COMP2016 run1-11	54	1	38	0.51	0.5238	0.2565*	0.0244†	0.2515†	0.0194†	7545	128
CLIP-A-1-08 [108]	91	1	89	0.5028	0.5083	0.2366†	0.0206†	0.2254†	0.0093†	178997	113

Note: The first four columns show the evaluation results by the mobile assessors and the following columns report the evaluation by NIST assessors. The column marked "ML" shows the median latency with respect to the first tweet in each cluster. The last column shows the length of each run, defined as the number of pushed tweets that were assessed. Rows of best runs in terms of the batch and the in-situ evaluation are sorted by EG1 and P_s respectively. The symbols *, †, and ‡ denote the Student test significance of adaptive learning improvement: $*\ 0.01 < p - value \leq 0.05$, † $p - value \leq 0.01$, ‡ $0.05 < p - value \leq 0.1$.

best TREC 2016 run in batch evaluation (COMP2016 run3-13)[59] overall metrics except for the nCG0 metric that corresponds to a recall-oriented measure in which systems are not penalized for pushing tweets on silent days. This observation holds despite the fact that the best TREC run is manual whereas our approach is automatic [86]. The fact that COMP2016-run3-13 run pushes more tweets than our approach may explain this result. We note that the performance improvements of our method are more important in terms of the precision-oriented metric (EG1) than in terms of recall-oriented metric (nCG). We found performance improvements of our approach up to EG-1 and nCG-1 measures of about 10.78% and 1.54% respectively compared to the best TREC 2016 run (COMP2016 run3-13). The performance improvement in terms of EG1 was found to be statistically significant with p-value < 0.05. The improvement in terms of EG1 shows that adaptive learning strategy that uses ongoing feedback has the potential to identify silent day. We also note that our approach outperforms run QUBaseline-37[122] which is the best performing automatic system at TREC 2016 track [122] overall batch evaluation metrics. The performance improvements of adaptive learning strategy compared to run QUBaseline-37 were found to be statistically significant with p-value < 0.05 in terms of EG1 and with p-value < 0.01 in terms of EG0, nCG1, and nCG0 metrics.

Regarding performance in terms of the in-situ evaluation, we notice that the best performing run (COMP2016 run3-13) in terms of the batch evaluation achieves the highest precision score whereas scores of the best automatic run QUBaseline-37 [122] falls in the 17 position among 42 runs that participated in TREC RTS 2016 track. Our approach failed to defeat the manual run (COMP2016 run3-13) in terms of in-situ evaluation. However, we note that our approach significantly outperforms QUBaseline-37 at significance level p< 0.01. In addition, our approach obtained similar performance as run CLIP-A-1-08 [108] which is the best automatic run in terms of in-situ evaluation. However, CLIP-A-1-08 [108] did not perform well in terms of the batch evaluation whereas our approach shows good performances in terms of the in-situ as well as the batch evaluation metrics.

6.5.5.2 *Comparison with the official results of TREC RTS 2017*

We compare in what follows the effectiveness of our approaches based on adaptive and passive classification model against the high performing run in TREC RTS 2017 track. Table 6.5 show performances in terms of the in-situ and the batch evaluation metrics. Recall that the set of topics in this track are new and unseen in TREC RTF 2015 track which is used to train the classifier that predicts the relevance of incoming tweets.

Overall, we note that our approaches were shown to be comparably effective to the best performing runs in each evaluation methodology whereas the best performing runs in terms of the batch evaluation did not perform well in terms of in-situ evaluation and vice-versa. It appears that the best performing runs (HLJIT testRun2-07[61]) and (WuWien-Run1-39and) in terms of the batch and the in-situ evaluations respectively trade latency for a higher quality which is pronounced in the high EGp score obtained by HLJIT testRun2-07[61] run. We observe that these runs have a very high latency whereas our approaches have a very low latency. This means that our proposed models based on machine learning present a good balance between latency and quality. In HLJIT testRun2-07[61] and (WuWien-Run1-39and) the decision to push a selected tweet is delayed to the end of the day which allows the system to accumulate better evidence and achieves higher EGp and precision respectively.

Results in terms of the batch evaluation show that the adaptive learning approach achieves the best performance overall metrics expect in terms of EGp compared to the best run in TREC RTS 2017 (HLJIT testRun2-07[61]. Also, the approach based on the passive learning also gets improvements. Our approach based on adaptive learning significantly outperforms the best TREC RTS 2017 run in terms of EG1, nCGp, and

Table 6.5: Comparison with the official TREC 2017 RTS track results.

	In-situ evaluation metrics				**Batch evaluation metrics**					
Method	P_s	P_l	$Util_s$	$Util_l$	**EGp**	**EG1**	**nCGp**	**nCG1**	**ML**	**L**
ABC-QF-UF-TF	0.3931	0.4640	-802	-207	0.3227	**0.2956**	**0.3032**	**0.2861**	1	1117
PBC-QF-UF-TF	0.3771	0.4451	-965	-431	0.3079‡	0.2769‡	0.2992	0.2737	1	1201
TREC RTS 2017 official Results: Best run in terms of the batch evaluation										
HLJIT testRun2-07[61]	0.3784	0.4446	-654	-298	**0.3630**	0.2088†	0.2808∗	0.1266†	56744	621
HLJIT testRun1-06[61]	0.3389	0.4082	-805	-459	0.3318	0.1811†	0.2610∗	0.1102†	49154	618
UDInfoBL-run2-34	0.3980	0.4708	-342	-98	0.3226	0.2622∗	0.2489†	0.1886†	55781	452
IRIT-Run1-14[66]	**0.4200**	**0.4814**	-198	-46	0.2918∗	0.2571†	0.2321†	0.1974†	1	320
TREC RTS 2017 official Results: Best run in terms of in-situ evaluation										
WuWien-Run1-39	**0.4337**	**0.4822**	-93	-25	0.2018†	0.1873†	0.1912†	0.1767†	19872	122
IRIT-Run1-14	0.4200	0.4814	-198	-46	0.2918	0.2571∗	0.2321†	0.1974†	1	320
PRNA-A1-21[80]	0.4140	0.4783	-262	-66	0.2090†	0.1951†	0.2052†	0.1913†	69	295
UDInfoSDWR-run1-35	0.4096	0.4941	-199	-13	0.2907	0.2571∗	0.2285†	0.1949†	60685	308
Performance of TREC RTS 2016 best run in 2017 track										
QUBaseline	0.3785	0.4386	-562	-284	0.2422†	0.2146†	0.2260†	0.1984†	1	446

Note: The first four columns show the evaluation results by the mobile assessors and the following columns report the evaluation by NIST assessors. The column marked "ML" shows the median latency with respect to the first tweet in each cluster. The last column shows the length of each run, defined as the number of pushed tweets that were assessed. Rows of best run in terms of the batch and the in-situ evaluation are sorted by EGp and P_s respectively. The symbols *, †, and ‡ denote the Student test significance of adaptive learning improvement: $* \; 0.01 < p - value \leq 0.05$, † $p - value \leq 0.01$, ‡ $0.05 < p - value \leq 0.1$.

nCG1 metrics. In this context, a number of observations are worth making. First, the performance of the approach based on the passive classification model (PBC) in terms of recall-oriented metrics (nCGp and nCG1) implies that our approach discovers more relevant tweets than HLJIT testRun2-07. This was expected since in HLJIT testRun2-07 only one tweet was pushed per topic and per day. Second, we note that run HLJIT testRun2-07 is a learning to rank based approach that relies only on query dependent features whereas our model combines query depend and social features. This highlights again the positive impact of considering social signals in real-time tweets filtering which provides another support to the hypothesis (H1). Third, taking into account that our approaches submitted more tweets than the best performing runs, the performance of our approaches in terms of the precision-oriented metric EG1 reveal that the classification model that leverages social signals is able to detect silent days and to keep silence in such days. Recall here that the EG1 metric penalizes systems for breaking the silence in a silent day by receiving a score of zero. Last but not least, the high score of the first three runs in terms of EGp metric can be explained on the one hand by the high latency and on the other hand the low number of pushed tweets. The former suggests that the submission of tweets is delayed in order to accumulate evidence while the latter implies that systems are less penalized if they push a tweet in a silent day. This finding is confirmed by the results obtained by run IRIT-Run1-14 [66] which is a simple baseline that consists of submitting at most one tweet per day per topic (the first tweet on a day having all the query terms). This run was ranked second in the in-situ evaluation and fourth in the batch evaluation.

Regarding performance in terms of the in-situ evaluation, we observer that our approach has lower utility compared to the best performing runs in TREC RTS 2017 track. This is may be explained by the high volume of tweet submitted by our approaches and the fact that there is a strong negative correlation between tweet volume and utility [129].

Finally, we note that the performance of the approach based on the passive classification model (PBC) obtained on TREC 2017 dataset clearly reveal that it is topic independent since topics considered on TREC 2017 dataset are unseen on the training dataset (based TREC 2015). This result provides a support for the hypothesis (H4). The learning-based approach can be applied to filter tweets with respect to unseen topics in the training dataset used to build the binary classifier.

6.6 Conclusions

To tackle the task of prospective notification in social media streams, we introduced a new approach that considers content and social signals and uses a supervised learning approach to filter in real-time the tweet streams. The main contribution of the proposed method is the introduction of an adaptive learning strategy that allows countering the threshold setting issue and take advantage of an ongoing assessment feedback. We proposed a set of social and other non-content features suitable for real-time tweet filtering.

Experimental results based on a real-world dataset revealed that the proposed approach outperforms the best automatic TREC RTS 2016 systems and was shown to be comparably effective to the best performing runs in the TREC RTS 2017 track. The proposed approach based on learning to filter constitutes a good trade-off between timeliness (latency) and quality (relevance and novelty) whereas the state-of-the-art approaches tend to trade latency for higher quality. We highlight the importance of social signals in tweet stream filtering tasks. It appears that user account features can be interpreted as a means of evaluating the "reliability" of the information source and tweet specific features can be considered as an evidence of the information quality. The learning based filter achieves a good balance between pushing too many or too few tweets. We showed that a learning-based approach is topic independent. Results also revealed that more improvements are achieved by taking advantage of ongoing relevance feedback.

Part IV

TWEET AGGREGATION

Brevity is the Soul of Wit.

– William Shakespeare

7 OPTIMIZATION FRAMEWORK MODEL FOR RETROSPECTIVE TWEET SUMMARIZATION

7.1 Introduction

In this chapter, we tackle the retrospective summarization task in social media stream which refers to the task of automatic summarization of long-ongoing events. The goal is to produce a concise summary that captures key aspects of the information need to help the user to make up for what he would have missed regarding the event of interest. In this task, the timeliness is not important since the events have already taken place and the user is looking for what happened (the main developments that have occurred until now). However, to be effective such summaries are expected to fulfill some important properties such as:

- Relevance: summaries should contain informative units that are relevant to the user interest;
- Redundancy: summaries should not contain multiple posts that convey the same information;
- Coverage: summaries should cover as many important aspects of the user's interest as possible and should also have diversity among them;
- Diversity: summaries should have diversity among the selected information because important information may be spread out over the lifetime of the event. To capture the developmental of an event over time, it is desired that summaries include information from different time windows over the lifetime of the given event;
- Length: the summary should be concise. Its length (number of tweets) is, in general, bounded so it would fit "a real world" scenario where summaries are generated for mobile push notifications.

Optimizing all these criteria jointly is a challenging task especially for long-running events. In [142] the tweet summarization problem is proven to be NP-hard. This is because the inclusion of relevant tweets relies not only on properties of tweets themselves but also on the properties of every other tweet in the summary.

Several approaches have been proposed to tackle this issue [67, 142, 168, 141, 123, 171, 97, 63, 7]. As discussed at the end of Chapter 3, most of these approaches generate summaries by iteratively selecting the most relevant tweets and discarding those having their similarity with respect to the current summary above a certain threshold. Such approaches ignore the mutual relation among tweets. In addition, these approaches do not consider the fact that important information may be spread out over the lifetime of the event of interest.

To overcome these issues, we propose a novel approach [32] that follows a different paradigm with the goal of increasing the coverage of different subtopics and time windows of a long ongoing event by considering the mutual relation between tweets. We propose to formulate the summary generation as an optimization problem modeled using Integer Linear Programming (ILP)[64]. An ILP problem is a constrained optimization problem, where both the cost function and constraints are linear in a set of integer variables. The summary generation is considered as an optimization problem that consists of selecting a subset of tweets that maximizes the global summary relevance and fulfills constraints related to non-redundancy, coverage, temporal diversity and summary length.

More precisely, the proposed method is designed for online retrospective summarization. It relies on a three-stage approach. First, tweets that do not have sufficient quality and word overlap with the query are discarded. Second, two incremental clusters of posts are determined, namely topical cluster, and temporal cluster. The former is based on tweet content while the latter is based on publication times. To measure tweet-tweet similarity, we make use of word embedding, which counters the shortness of tweets as well as the term mismatch issue. Third and last, a subset of posts is selected so as to maximize their overall relevance to the query subject to constraints related to, summary length, temporal diversity, coverage, and redundancy. In order to handle this selection, we formulate the tweet summary generation as integer linear problem in which unknowns variables are binaries, the objective function is to be maximized and constraints ensure that at most one post per cluster from the two categories of clusters (topical and temporal) is selected with respect to the defined summary length limit.

Integer Linear Programming (ILP) techniques have been used in multi-document summarization [104, 81] and in microblog summarization [88]. The optimization problem proposed in our work differs from those proposed in state of the art by:

- It takes into account the temporal dimension which is not the case in the related works.
- The coverage and redundancy requirement are represented in the same constraint while in [88] a redundancy constraint is created for each pair of tweets which increases the computational complexity of the generated ILP.

The main contributions of the proposed approach are:

- We adopt Integer Linear Programming technique to periodically generate a summary in order to optimize all the aforementioned criteria. To reduce the computational complexity and handle the coverage issue, the tweet stream is filtered and clustered in real-time.
- In order to capture the development of the event over its lifetime, we take into account the temporal diversity of tweets as one criterion that needs to be fulfilled in the summary generation process.
- We do not rely on statistics to evaluate the relevance score of an incoming tweet which allows to estimate it at the time the new tweet arrives independently of the previously seen tweets and without the need for indexing tweet stream.

Notice that, we focus in this chapter on the tweets incremental clustering and summary generation. The relevance score of the incoming tweet is computed using WSEBM that we describe previously in Section 5.2.2 [34]. To filter out irrelevant tweets, we use learning to filter approach proposed in the previous chapter.

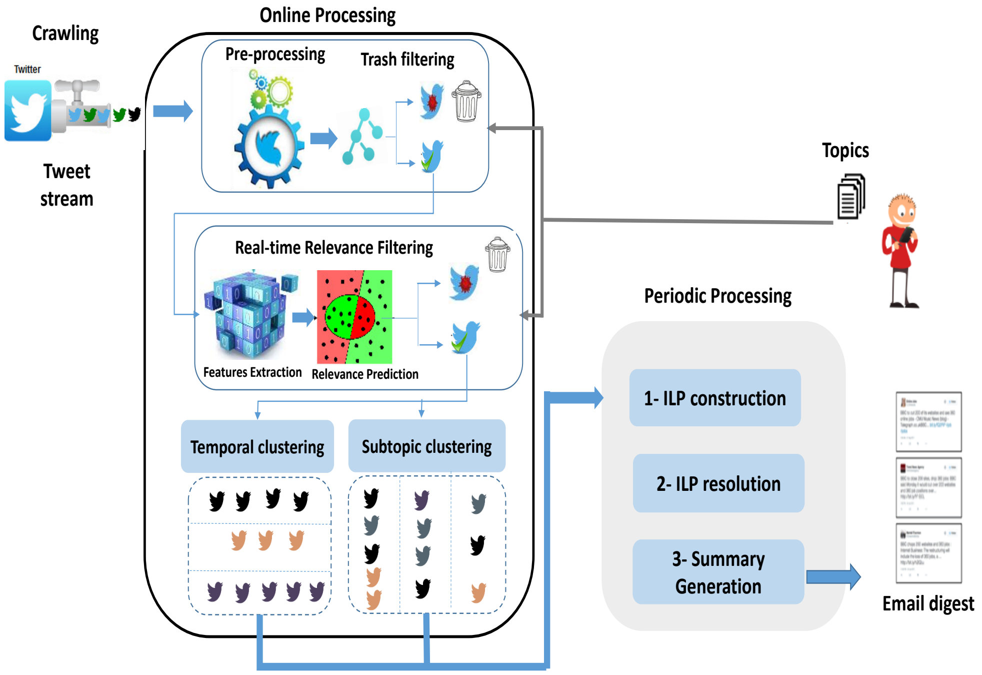


Figure 7.1: Overview of the tweet summary generation approach based on ILP.

The remainder of this chapter is organized as follows: First, we introduce an overview of our approach. Afterward, we put forward the incremental tweet clustering method. Then, we describe the proposed integer linear programming model that formulates the summary generation in tweets stream. Finally, we conduct a series of experiments based on TREC 2015, 2016 and 2017 tracks corpus in order to evaluate the effectiveness of our model.

7.2 Retrospective tweet summarization

Our goal is to periodically generate the summary that can best convey the main ideas of the user information need within the length limit and a minimum of redundancy. To do so, we propose a new approach that filters and clusters tweet stream and then periodically selects relevant tweets to be pushed to the user as a summary of a long-ongoing event.

To achieve this purpose, the proposed approach includes two main components as depicted in Figure 7.1: (i) **An online component** that filters and clusters crawled tweets in real time after prepossessing step, and (ii) **an off-line component** that generates a summary periodically after the end of a predefined time window for instance one day.

The tweet stream filtering and clustering component consists of three main steps as listed below:

1. *Pre-possessing and quality filtering:* This step filters out trash tweets and those that do not have enough overlap with the query after prepossessing (stop-words removal and stemming). This filter is the same as the one used in the previous chapters (Section 5.3.2.2);

2. *Relevance estimation and filtering:* In this step, first a relevance score of the incoming tweet with respect to the query is evaluated. This score is computed using the proposed model (WSEBM) according to the Equation 5.4 described in Section 5.2.2 [34]. Then potential irrelevant tweets are discarded. To filter tweets in this stage, we rely on the binary classification model build on Random Forest algorithm with features proposed in the previous chapter (Section 6.3). This filter contributes to reducing the number of candidates tweets and decreases the computational complexity. By doing this, we make feasible the use of ILP.
3. *Incremental tweet clustering:* The purpose of this step is to identify the different subtopics (aspects) of an event and to gather tweets in different time windows over the lifetime of the given event. Tweets are clustered in real-time while they arrive. This makes possible to generate and to issue the summary at any time to a user tracking the development of an event over time.

The summary generation component selects a subset of tweets from a set of candidate tweets that pass the filtering step. The goal is to select tweets that fulfill requirements related to the non-redundancy, the topical coverage, temporal diversity, and the summary length. To achieve this goal, we propose to use an Integer Linear Programming (ILP) model which selects tweets that optimize a global objective function under certain constraints. This step is executed periodically within a predefined time window.

We present in what follows the proposed model to generate a periodic tweet summary based on ILP. Before discussing this, we will describe the incremental tweets clustering method in which incoming tweets that pass relevance filter are gathered in two cluster types, namely topical and temporal clusters. The first one is based on topical similarity and the second one is based on tweet timestamp.

7.2.1 *Incremental tweet clustering*

The summary should cover all the aspects users are interested in. For example, a summary of a natural disaster should include aspects of what happened, when/where it happened, damages, rescue efforts, etc., and these aspects are provided by different tweets. We assume that an effective summary should also contain information nugget from different time window in order to give an overview of the development of the event. Hence, we propose to consider both dimensions, topical similarity and temporal distance between tweets in order to enhance the coverage and diversity in the summary. Given a tweet stream, we automatically cluster tweets into two types of clusters namely topical and timeline clusters. In the former, tweets sharing similar terms are absorbed into the same cluster and in the latter, tweets published in the same time window are gathered in the same timeline cluster.

7.2.1.1 *Subtopic clustering*

The subtopic clustering is based on a pairwise similarity comparison between an incoming tweet and centroids of existing clusters. For an incoming tweet T the key problem is to decide whether to absorb it into an existing cluster or to upgrade it as a new cluster. We first find the cluster whose centroid is the nearest to T. Tweet T is added to the closest cluster if its similarity score is greater than a predefined threshold γ; otherwise,

T is upgraded to a new cluster with T as the centroid. Each time an incoming tweet is added to an existing cluster its centroid is updated. We choose as new centroid the tweet that has the highest value of the sum of similarity scores with all other tweets in the cluster. To overcome the issue of word mismatch when measuring the tweet-tweet similarity, we use word embedding model to estimate the similarity between tweet's terms. The similarity between two tweets T and T' is computed as follows:

$$Sim(T, T') = \frac{\sum_{t_i \in T} \max_{t_j \in T'} w2vsim(t_i, t_j)}{|T \cup T'|} \tag{7.1}$$

Where $w2vsim(t_i, t_j)$ is the cosine similarity between vectors of terms t_i and t_j which are generated by the word2vec model[106].

The use of maximum instead of average allows getting a similarity score equal to 1 if the term t_i occurs in both tweets. In the other case, where term t_i of tweet T does not occur in T', the maximum will return the similarity score of the most similar term in T' to t_i whereas the average may return a small score if terms that occur in T' are very different from t_i. This fact holds even if tweet T' contains term t_i. In the case that a tweet term is out of the vocabulary in the word embedding model, the similarity score is set to zero.

7.2.1.2 *Timeline clustering*

The aim of timeline clustering is to capture the development of the event over the time. We would like to avoid that the summary contains tweets published in the same time window. Indeed, we believe that all tweets that are published in the same time window are more likely to be related to each other. In timeline clustering, tweets posted in the same time window are absorbed in the same timeline cluster. The decision to whether the incoming tweet is added to the current cluster is based on the delay (in seconds) between its timestamp and the timestamp of the first tweet used to create the actual cluster. If the delay is higher than a certain time window size, a new time cluster is created; otherwise, the incoming tweet is added to the current time cluster.

7.2.2 *Summary generation*

After filtering and clustering steps, the final step is the generation of the summary. We propose to formulate the tweet summarization as an Integer Linear Programming (ILP) problem in which both the objective function and constraints are linear in a set of integer variables. More specifically, we would like to select from M candidate tweets (those that pass the filter) N tweets that maximize the relevance score with respect to the query and fulfill a series of constraints related to redundancy, coverage, temporal diversity, and length limit. To find the optimal solution, we use the branch and bound algorithm [64].

Assume that there is a total of M candidate tweets that are clustered in A subtopic clusters (denoted C_j) among them there are s clusters that contain at least two tweets. In the same way, assume that there is a total of W timeline clusters (denoted TW_l) that contain at least two tweets. The tweet summarization problem can be formulated as the following ILP problem:

We include a binary variable X_i which is set to 1 when tweet T_i is added to the summary and 0 otherwise. The goal of the ILP is to set these indicators variables to maximize the payoff subject to the set of constraints that guarantee the validity of the solution. Notice here that the first constraint states that the indicator variables are binary.

$$\forall i \in [1, M], X_i \in \{0, 1\}$$

7.2.2.1 *Objective function*

Top-ranked tweets are the most relevant tweets corresponding to the related aspects which we want to include in the final summary. Thus, the goal is to maximize the global relevance score of selected tweets that optimize the overall coverage, temporal diversity and relevance of the final summary. The objective function is defined as follows:

$$\max(\sum_{i=1}^{M} X_i \times RSV(T_i, Q))$$

Where $RSV(T_i, Q)$ is the relevance score of tweet T_i with respect to query Q which is computed according to the Equation 5.4 described in Section 5.2.2.

7.2.2.2 *Coverage and redundancy constraints*

These constraints fulfill both redundancy and coverage requirements. In order to avoid redundancy, we just choose at most one tweet from each topical cluster. Indeed, the limitation of the number of tweets from each cluster guarantees that a maximum of sub-topics (aspects) will be presented in the summary such that the summary can cover most information of the whole tweet set. These constraints are formulated as follows:

$$\forall C_j \in \{C_1, ..., C_s\} \sum_{i;T_i \in C_j} X_i \leqslant 1$$

Where C_j is the j^{th} subtopic cluster and s is the number of subtopic clusters that contain at least two tweets.

7.2.2.3 *Temporal diversity constraints*

To guarantee that the summary contains tweets from different time windows, we choose in maximum one tweet from each time window cluster. These constraints are formulated as follows:

$$\forall TW_l \in \{TW_1, ..., TW_w\} \sum_{i;T_i \in TW_l} X_i \leqslant 1$$

Where TW_l is the l^{th} temporal cluster and w is the number of temporal clusters that contain at least two tweets.

7.2.2.4 *Length Constraint*

We add this constraint to ensure that the length of the final summary is limited to the minimum of either a predefined constant N (i.e. the maximum length) or $M - 1$ where M is the number of candidate tweets.

$$\sum_{i=1}^{M} X_i \leqslant min(N, M - 1)$$

7.3 Experimental evaluation

To evaluate our approach, we carried out twofold objectives experiments: First we conducted a series of experiments on TREC RTF 2015 dataset to set parameters used in our approach. Second, we compare our approach with the state-of-the-art methods and with the three best performing runs in TREC RTS 2016 and 2017 tracks. In particular, we evaluate the impact of the use of ILP for generating a retrospective summary in social media which is compared to Top-k based approaches.

7.3.1 *Experimental setup*

Experiments were conducted by using replay mechanism of scenario "B" over tweets captured during the evaluation period of the TREC 2015 Microblog Real-Time Filtering (MB RTF)[85], TREC Real-Time summarization (RTS) 2016[86] and 2017 [87] tracks. Recall that the scenario "B" in these tracks is more like a top-100 retrieval task based on a one-day. It consists of identifying a batch of up to 100 ranked tweets per day and per topic which are delivered to the user daily (at the end of the day).

Performance of systems was evaluated in terms of the normalized Discounted Cumulative Gain (nDCG). This metric gives higher value to the well-ranked list. Note that in these tracks, the redundancy is implicitly handled by computing the gain of returned tweets with respect to the semantic clusters. Indeed, a semantic clustering where conducted by NIST assessors in which relevant tweets that share substantively similar content were clustered into the same semantic cluster. Systems receive a gain for only the first relevant tweet from each cluster. Hence, a system that submits more than one relevant tweets but "saying the same thing" is penalized by receiving a null score for all returned tweets except for the first one.

As baselines, we use the three approaches that were recommended by [97] to be considered as baselines since it turned out to be the best one among 11 different tweet summarization approaches. These approaches are TF-IDF, HybridTF-IDF[141] and sumbasic [110].

In addition, to evaluate the impact of ILP, we consider as a baseline a variant of our approach in which we disable ILP. In this baseline denoted by WSEBM-TOP10, we select iteratively the TOP-10 tweets but with discarding those having a similarity score above the predefined threshold (the same value of the one used for the subtopic clustering). Tweets that pass the relevance filter are sorted according to their relevance score computed by WSEBM model proposed in Equation 5.4 [34]. We choose to select TOP-10 tweets because the evaluation metrics are computed on top-10 tweets.

There are several parameters in our method namely the similarity threshold and the size of time windows that control the subtopic (Section 7.2.1.1) and timeline clustering (Section 7.2.1.2) respectively. We describe in the following subsection how we tune these parameters.

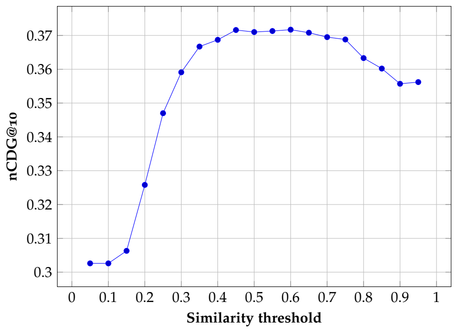


Figure 7.2: Impact of topical clustering on TREC RTF 2015 dataset

7.3.2 *Parameter Setting*

In our experiments, we use TREC RTF 2015 dataset to tune the similarity threshold and the size of time windows used in the subtopic and temporal clustering respectively as follows:

7.3.2.1 *Effect of subtopic clustering*

The topical clustering is controlled by the tweet-tweet similarity measurement and the similarity threshold γ. Figure 7.2 shows the effect of the similarity threshold in terms of nDCG@10 obtained by the proposed similarity function denoted by (Jaccard-w2v). In this experiment, we gradually vary the similarity threshold γ from 0.05 to 1 at the step of 0.05 and we disable the temporal clustering by setting the time window size to zero($\tau = 0s$). From Figure 7.2, we can see that the performance improves when γ increases but it decreases when γ comes near to 1. These results were expected for the following reasons: In the first case (γ small) the number of clusters that contain at least two tweets decreases (all tweet may be gathered in the same cluster) and only one or few tweets (with the highest relevance score) are selected causing damage in terms of coverage. In the second case (γ near to one), there are no clusters with at least two tweets. This means that there are no constraints related to the topical coverage. In this case, the ILP selects the top-k tweets without discarding the redundant ones leading to hinder the quality of the summary.

These results reveal that $\gamma = 0.6$ appears as a good choice as it gives a good balance between the number of clusters and the number of tweets in each cluster. Hence, for the next experiments, we set γ to 0.6.

7.3.2.2 *Effect of timeline clustering*

The timeline clustering is based on the size of the time window. Hence to test the effect of the use of the timeline clustering, we conducted experiments in which we vary the time window size (τ) from 0 to 1800 seconds and we keep others parameters fixed. The obtained results are shown in Figure 7.3. We notice that the performance decreases when τ, increases. On the one hand, when τ is large, we obtain clusters that contain a lot of tweets causing to discard many tweets which damage the quality of the summary. On the other hand, when τ is very small, no time cluster is created which means that we do not have any constraint related to temporal diversity. Thus, trying to maximize

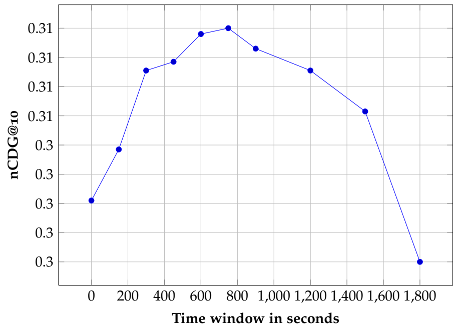


Figure 7.3: Impact of temporal clustering on TREC RTF 2015 dataset

the temporal diversity will probably lead to a result degradation. It seems that $\tau = 600s$ is a good value that leads to a good balance between the number of timeline clusters and the number of tweets in each cluster. Hence, for the next experiments, we set τ to $600s$.

7.4 Results and Discussion

7.4.1 *Impact of the use of ILP*

We compare the impact of the use of ILP to generate the summary against the TOP-10 selection strategy within TREC RTF 2015. In [92] authors show that the treatment of silent days has a large impact on system scores in TREC MB RTF 2015. For this reason and to better perceive the impact of the use of the ILP, we present the obtained results over both all 51 topics and over only the 14 eventful days topics (for which there is no silent day). In this experiment, we gradually vary the similarity threshold γ from 0.5 to 0.95 at the step of 0.05. Recall that in TOP-10 selection strategy, we discard tweets that have a similarity score regarding already selected tweets above the predefined threshold. Figure 7.4 reports the results obtained overall judged topics (51) in terms of nDCG@10 by varying the similarity threshold used in subtopic clustering γ gradually. As shown in this Figure, the use of ILP yields better performances overall similarity threshold. The positive improvements are statistically significant with p values between 0.01 and 0.05 for the similarity threshold $\gamma <= 0.55$ and between 0.05 and 0.1 for the similarity threshold $\gamma >= 0.6$. We found performance improvements of about 3.48% for the similarity threshold $\gamma = 0.5$ and of about 6.20% for the similarity threshold $\gamma = 0.6$. From Figure 7.5, we can see that the performance improvements of ILP compared to the TOP-10 approach in terms of nDCG@10 are better over eventful days topics than overall 51 topics and overall the similarity threshold values. When only the eventful topics are considered, the obtained performance improvements of ILP vary between 7.92% and 8.94% for the similarity threshold $\gamma = 0.5$ and $\gamma = 0.6$ respectively. Whereas when considering all topics, the use of ILP improves the performance with about 3.48% and 6.20% for the same similarity thresholds. These results reveal that the proposed method is more effective for events that raise a lot of reactions in social media. In fact, the impact of tweets clustering and the use of ILP to generate a summary is more significant when the number of candidate tweets M is greater than the desired length limit of the summary N (set to 10 in our experiments). In the case of $M \leq N$, the ILP component acts almost like top-K ranking methods since it selects all candidate tweets with discarding the redundant tweets.

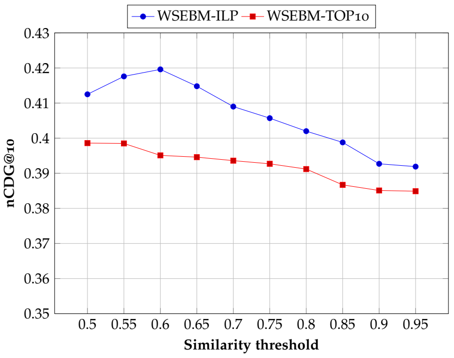


Figure 7.4: ILP vs TOP10 over all topics.

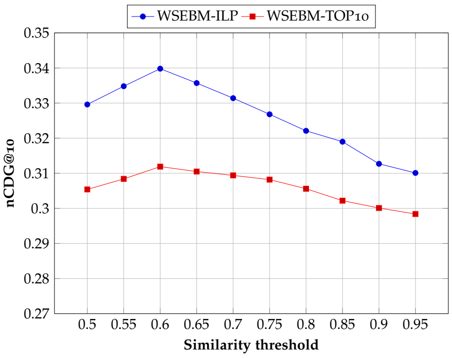


Figure 7.5: ILP vs TOP10 over eventful topics.

7.4.2 *Impact of subtopic and timeline clustering*

In this experiment, we compare the performance of the ILP that leverages subtopic and timeline clustering against the ILP based solely on subtopic clustering (SC) as well as the ILP with solely the timeline clustering (TC). Figure 7.6 shows the performance in terms of nDCG1@10 of each variant of our approach on TREC 2015, 2016, and 2017 datasets.

From Figure 7.6, we can notice that the ILP based on subtopic clustering outperforms the ILP based on the timeline clustering on both datasets. This result was expected since the use of timeline clustering without subtopic clustering may lead to select redundant tweets that were published in two different time windows. The inclusion of such kind of tweets degrades the quality of the summary. In contrast, considering the timeline with subtopic clustering improves the performances. The performance improvement of the ILP with the two types of clustering compared to the ILP based solely on subtopic clustering are about 4.71%, 9.25% and 4.78% on TREC 2015, 2016, and 2017 respectively. The positive improvement of the use of the two types of clustering is statistically significant with p-value <0.05 but there is no statistically significant difference between ILP-SC and ILP-TC. This result suggests that timeline clustering which introduces a diversity in the summary is useful. The proposed framework simultaneously reduces the redundancy and adds the diversity in the summary.

7.4.3 *Comparative evaluation with state-of-the-art baselines and the official TREC results*

In this subsection, we compare the effectiveness of the proposed approach against the adopted baselines from the literature and the high performing runs on TREC RTS 2016 and 2017 tracks. Notice that a brief description of approaches from TREC RTS track runs was presented in Table 3.3. In these experiments, we set the similarity threshold that controls the subtopic clustering γ to 0.6 and the size of the time windows τ to $600s$.

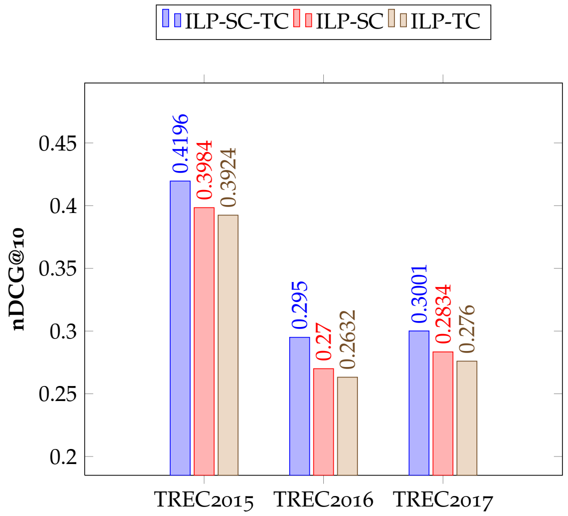


Figure 7.6: Impact of the timeline and the subtopic clustering.

7.4.3.1 *Comparative evaluation on TREC RTS 2016 results*

Table 7.1: Comparative of effectiveness on TREC RTS 2016 dataset.

Method	nDCG1@10	nDCG0@10	length	%
WSEBM-ILP	**0.2950**	**0.1201**	786	
WSEBM-TOP10	0.2708‡	0.0583†	687	+8.93%
HybridTF-IDF	0.1678 †	0.0767‡	1237	+75.85%
TF-IDF	0.1745†	0.0834‡	1173	+69.10%
SUMBASIC	0.1655†	0.0536‡	1158	+78.30%
TREC RTS 2016 official Results				
PolyURunB3	0.2898	0.0684†	444	+1.79%
nudtsna	0.2708‡	0.0529†	1481	+6.31%
QUJM16	0.2621‡	0.0301†	350	+9.84%

Note. % indicates the proposed method improvements in terms of nDCG-1@10. The symbols *, †, and ‡ denote the Student test significance: $*$ $0.01 < t \leq 0.05$, † $t \leq 0.01$, ‡ $0.05 < t \leq 0.1$.

In Table 7.1, we compare our model (WSEBM-ILP) as well as the degenerate version (WSEBM-TOP10) with the three high-performing runs (PolyURunB3 [59], nudtsna, QUJM16 [122]) from the TREC RTS 2016 track [86] and against state-of-the-art baselines within TREC RTS 2016 dataset. Notice that there is no paper in the TREC 2016 proceeding that describes how the run *nudtsna* were produced.

To get a deeper understanding of the effectiveness of the proposed method, we show in Table 7.1 the obtained results in terms of nDCG-1@10 as well as in terms of nDCG0@10 metrics. We recall that in the latter metric, systems are not penalized for pushing tweets in a silent day. First, we notice that our approach outperforms the state-of-the-art methods overall metrics with an improvement up to 75.85%, for the HybridTF-IDF and up to 69.10% for TF-IDF. We also notice that our model (WSEBM-ILP) slightly outperforms the best performing run (PolyURunB3) in TREC 2016 in terms of nDCG1@-10 with a significant improvement of the performance in terms of nDCG0@10 in which systems are not penalized for pushing tweets for a silent day. The performance improvements are up to nDCG-1@10 and nDCG-0@10 values of about 1.79% and 75.58% respectively. The positive improvement in terms of nDCG0@10 is statistically significant with (p-value <0.01). Notice here that these performances are achieved despite the fact that our method is automatic, while in the best TREC runs (PolyURunB3) [59] the threshold used in tweet filtering stage is based on the observation on the Tweet Stream for days before evaluation period. To improve performance in terms of nDCG-1, the system needs to identify silent day which can be achieved with better tweet filter setting. These results show that both approaches (ours and PolyURunB3) perform well when it comes to not pushing tweets for the silent day which improves performance in terms of nDCG1@10. However, for the eventful day, our method pushes more relevant and not redundant tweets than PolyURunB3 which explains the improvement of the performance in terms of nDCG0@10. These results are consistent with previous findings that our approach is more efficient for the event that catches a lot of attention in social media. In addition, we observer that our approach outperforms the best automatic TREC 2016 run (nudtsna) overall metrics. We found the performance improvements up to nDCG-1@10 and nDCG0@10 values of about 6.31% and 127.03% respectively.

7.4.3.2 *Comparative evaluation on TREC RTS 2017 results*

Table 7.2: Comparative of effectiveness on TREC RTS 2017 dataset.

Method	nDCGp@10	nDCG1@10	length	%
WSEBM-ILP	0.3457	0.3001	1846	
WSEBM-TOP10	0.3269‡	0.2824	1764	+5.75%
HybridTF-IDF	0.3095∗	0.2497∗	3226	+11.69%
TFIDF	0.3166‡	0.2570∗	3237	+9.19%
SUMBASIC	0.3022∗	0.2424∗	3206	+14.39%
TREC RTS 2017 official Results				
HLJIT qFB_url[61]	**0.3656**	0.2910	4574	-5.44%
PKUICSTRunB1 [151]	0.3483	**0.3003**	2409	-0.74%
HLJIT HLJIT_l2r[61]	0.3274‡	0.2778‡	3946	+5.58%

Note. % indicates the proposed method improvements in terms of nDCGp@10. The symbols *, †, and ‡ denote the Student test significance: $* \; 0.01 < t \leq 0.05$, † $t \leq 0.01$, ‡ $0.05 < t \leq 0.1$.

In this section, we describe the results obtained by baselines and the proposed model (WSEBM-ILP) as well as the degenerate model (WESEBM-TOP10) on TREC RTS 2017 dataset. These results are compared against the high performing runs in scenario B of TREC RTS 2017 track namely *qFB_url*, *HLJIT_l2r* [61] and *PKUICSTRunB1* [151]. We recall here that all these run are based on top-k selection. Run *qFB_url* is based on language model in which the tweet text is extended with terms of the linked URL webpage. *HLJIT_l2r* run is Learn to Rank model based on listNet algorithm [28]. The run *PKUICSTRunB1* is based on negative KL-divergence language model.

Table 7.2 shows performance in terms of two variants of nDCG metric (nDCG1@10 and nDCGp@10). Note that the nDCGp@10 metric is the official metrics in the TREC RTS 2017. The unique difference between these metrics lies in the way in which systems are penalized for pushing tweets on a silent day. On such day, nDCG1 variant is binary whereas nDCGp is based on a linear penalty. In the former metric, a system receives a perfect score (1) if it does not push any tweet, or zero otherwise, while in the latter metric the penalty is gradually increased from 0 to 1 according to the number of pushed tweets.

Results shown in Table 7.2 are rather promising for the following reasons:

- WSEBM-ILP shows better results than WSEBM-TOP10 which is consistent with previous results on TREC RTS 2016 dataset. Given the same set of candidates tweets, the use of ILP yields to generate a summary with higher quality than the one builds using the traditional TOP-K selection approach.

- We observe that both of our models outperform the considered baselines overall metrics. This confirms that the use of word similarity based on word embedding when computing the relevance score of a tweet leads to better identify relevant tweets since it is able to consider different words with the same semantic meaning.

- We note that our model WSEBM-ILP overpasses the third best run (*HLJIT_l2r*) [61] in TREC RTS 2017 overall metrics. In fact, WSEBM-ILP outperforms the best run (*qFB_url*) [61] in terms of nDCG1@10 metric. However, (*qFB_url*) show higher nDCGp@10 value which indicates that (*qFB_url*) submitted more relevant tweets than WSEBM-ILP. This result can be explained by (i) the high number of tweets (4574) returned by this run and (ii) the fact that in (*qFB_url*), the tweet content is extended with terms of the web page linked by URL mentioned in the tweet. It appears that external evidence is helpful to improve the overall system performance. It is also interesting to observe that WSEBM-ILP overpasses the (*HLJIT_l2r*) run that uses the relevance feedback of mobile assessors (which occurred during the evaluation period as the systems pushed tweets for scenario A). These results confirm that our proposed model based on ILP and word similarity is competitive on the one hand to Learning to Rank model with relevance feedback (*qFB_url*) and on the other hand to language models based approaches (*PKUICSTRunB1*, *qFB_url*)

These results reveal that our approach achieves a good balance between pushing too many tweets and pushing a few tweets. These trends can be explained by first, constraints related to the temporal and the topical coverage allow taking into consideration the mutual relation between tweets which is not the case in the state-of-the-art approaches based on the selection of the top-k tweets. Second, the use of word embedding in computing the tweet-query relevance score leads to boost tweets that contain different terms but sharing the same semantic context with query terms whereas the state-of-the-art baselines are based on stream statistics. Third and last, our approach acts as a top-k selection method when the number of candidate tweets is less than the summary length or when there are no clusters that contain more than one tweet. Somehow, the top-k method can be considered as a particular case of the proposed ILP.

7.4.4 *Scalability*

The scalability experiment evaluates the time-cost of running the summary generation component. It simulates the generation of a daily summary. Figure 7.7 presents the

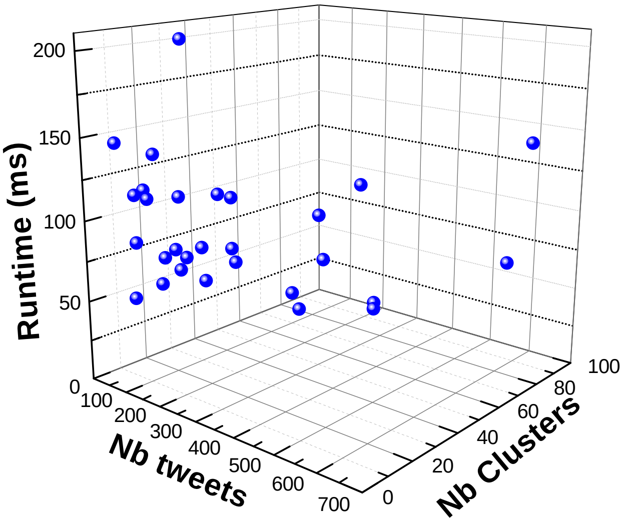


Figure 7.7: Run time of summary generation.

running time of the summary generation according to the number of candidate tweets and topical clusters. We can see that the time cost is less than 1 second. The average of the runtime per topic is 93 ms. We observe that the run-time is proportional to the number of candidate tweets and it is inversely related to a number of subtopic clusters.

7.5 Conclusions

In this chapter, we addressed the task of tweet summarization for a long ongoing event in the social media stream. We introduced a new approach based on an optimization framework to generate a periodic summary of the tweets stream. The main contribution of the proposed method is that tweet selection problem is formulated as ILP that maximizes objective function based on the tweet's relevance score subject to a series of constraints related to redundancy, coverage, temporal diversity, and length limit. To enhance summary coverage, we take into account the different subtopic that may occur. To capture the development of the given event over its lifetime, we consider the temporal context of tweets. In order to overcome the word mismatch issue in the computation of tweet-tweet similarity, we take advantage of the word embedding model.

Experimental results based on two real-world datasets revealed that the proposed approach outperforms the baseline methods as well as the best TREC RTS 2016 systems and shows promising results on TREC RTS 2017 dataset since it falls in the third position. The results also showed that more improvements are achieved on the queries with eventful days in a tweet stream which correspond to events that attract a lot of interest in social media.

Part V

CONCLUSION

It's more fun to arrive a conclusion than to justify it.

— Malcolm Forbes

8 CONCLUSION

8.1 Synthesis of contributions

The work presented in this thesis addresses tweet summarization for a long ongoing event in which we distinguish two complementary scenarios, namely retrospective and prospective summarization. In this task, it is required that a system monitors the live stream of tweets. In the prospective summarization, tweets are processed in real-time whereas in retrospective summarization tweets are treated on a batch.

To tackle this issue, we started by reviewing the state-of-the-art work through which we spotted the limitations that hinder their performances. Hence, we noticed that in the majority of the proposed approaches, (i) the relevance models rely on stream statistics and often do not consider the social context of tweets, (ii) the novelty of the incoming tweet is based on a pairwise comparison with all previously selected tweets on the summary. Additionally, in prospective tweet summarization, the decision to select/ignore an incoming tweet is threshold-based and systems tend to trade latency for a higher quality of tweet. The retrospective summary is generated by selecting iteratively the top-k relevant tweet with discarding redundant ones. Accordingly, we identified four different research questions that deal with (i) the definition of relevance model able to overcome the word mismatch issue and does not rely on tweet stream statistics, (ii) the threshold setting issue in relevance filter, (iii) the consideration of the social context of tweet in real-time tweet filtering (iv) optimizing jointly all criteria required in the summary and considering the fact that important tweets may be spread out over the lifetime of the given event. Thus, our contribution can be summarized in the following main points:

1. We introduce a novel relevance model based on word similarity matching in which we take advantage of word embedding representation to evaluate term-term similarity. The proposed model is an adaptation of the Extend Boolean Model that, instead of using TF-IDF weighting scheme, use the word embedding similarity between query and tweets terms. The intuition behind this proposition is that tweets that contain words sharing many contexts with the query words are more likely to be relevant. The proposed model overcomes the word mismatch issue since a query term that does not occur in a tweet but shares the same context with the given tweet's terms gets a nonzero weight. Additionally, the relevance score of the incoming tweet is estimated at the time the new tweet arrives independently of the previously seen tweets and without the need to maintain an index of tweet stream. The experiments underlined that leveraging the semantic relationships between terms allows improving the retrieval of relevant posts. The proposed relevance function enables the use of simple threshold across all topics. We highlighted the impact of the threshold setting strategy on the quality of real-time tweet summarization. We showed that better results can be achieved if the threshold value is appropriately set.

2. We propose a simple but efficient novelty detection method that does not rely on either a pairwise comparison or the tweet stream statistics. The novelty score of the candidate tweet is evaluated regarding all words of tweets already selected on the summary. We used a modified version of word overlap to compute the similarity between the incoming tweet and the summary. Our approach scales better than state-of-the-art methods and allows reducing the computational complexity. To evaluate the effectiveness of the proposed novelty detection method, we conducted extensive experiments on a real-world dataset (from a recent tweets stream) independently of the relevance filtering step. The obtained results show that our method outperforms the baseline approaches. Moreover, as our method does not use a pairwise comparison, it is much faster and scalable than the others.

3. We put forward a Learn to Filter approach that uses a machine learning technique to build a binary classifier that predicts the relevancy of the incoming tweets. This proposition has a twofold objective. On the one hand, it overcomes the relevance threshold setting issue. On the other hand, it allows considering the social context of tweets in addition to the content provided by the tweet. Hence, we proposed and evaluated a set of social and other non-content features suitable for real-time tweet filtering. We partitioned social features into two categories, namely tweet specific and user-account features. Our experiments highlight the importance of social signals in tweet stream filtering tasks. The comparison between the two categories of features reveals some differences in their effect. It appears that user account features can be interpreted as a means of evaluating the "reliability" of the information source and tweet specific features can be considered as an evidence of the information quality.

   We also extend the Learn to Filter approach to an adaptive learning approach in which the binary classifier is periodically re-trained. To do so, we take advantage of an ongoing relevance feedback.

   Experimental results revealed that the proposed approach based on learning to filter presents a good trade-off between timeliness (latency) and quality (relevance and novelty). The learning based filter achieves a good balance between pushing too many or too few tweets. Results also showed that more improvements are achieved by taken advantage of ongoing relevance feedback.

4. To optimize all criteria required on a retrospective summary, namely the relevancy, the low redundancy, the coverage, the temporal diversity, and the conciseness, we proposed to model the summary generation as Integer Linear Programming problem. To ensure that the summary includes tweets that convoy various aspects and published during different time periods over the lifetime of the given event, tweets are incrementally clustered into two types of clusters. The first type of clusters is based on the content similarity between tweets whereas the second type relies on the temporal context of tweets. The proposed ILP model guarantees that at most one post per aspect and time period is selected with respect to the defined summary length limit. Experimental results based on two real-world datasets revealed that the proposed approach outperforms the baseline methods as well as the best TREC RTS 2016 systems and shows promising results on TREC RTS 2017 dataset

since it falls in the third position. The results also showed that more improvements are achieved on the queries with eventful days in a tweet stream.

8.2 Perspectives

As we have discussed in this thesis, various approaches have been proposed for event summarization in tweet streams. However, there are still lots of problems that have not been addressed yet, which can be important as future research directions. Here, we list from our perspective some important future research:

1. **Evaluation of the source of the information:** A user in social media becomes an individual news media that produces/propagates information about what is happening in the world. The current tweet summarization approaches consider that the user-accounts have the same importance independently of the association that may be between the category of user-accounts and the event of interest. As source of an information, a user-account should be evaluated. We believe that the reliability of a user-accounts depends not only on its importance in the social network but also on its category (official account of an organization, account of a celebrity or personal account) and the type of the event of interests. For instance, in the case of an airplane crash, it obvious that any information published by the official account of the airline company is more reliable (more relevant) than information published by an unknown user-account.

2. **Cross validation of the information:** There is an increasing need to verify and determine the accuracy of the information provided in social media. The state-of-the-art tweet summarization approaches do not take into consideration the credibility of the information conveyed by a tweet. Indeed, all tweets are assumed to be similarly credible and reliable. We argue that the evaluation of the credibility of an information conveyed by a tweet, for instance on a scale of 0 to 5, can enhance not only the relevance of a retrospective summary but also the effectiveness of the real-time tweet filtering. However, the evaluation of the credibility of an information on social media is challenging. One solution is to rely on another social media such as Facebook to corroborated the information. Additionally, the location of the users who posted the tweet (whether they were in the same place where the event of interest took place when the tweet had been posted) can be considered as an evidence of the reliability of the information. Also, the quality of the user-account discussed above can be useful to estimate the credibility of the information.

3. **Considering other dimensions of diversity in the summary:** To generate the summary that covers different aspects and time periods, we focused on the content similarity and temporal context of tweets. Nevertheless, there are other important aspects that are worth studying, such as:

   - *Geospatial diversity:* Twitter provides a location information about a tweet. Leveraging geospatial dimension may enhance the diversity in the summary. For instance, in a presidential election, a summary that includes tweets posted from different regions of the country will provide a better overview about the general trend than a summary that contains tweets posted from the same region. Additionally, the geospatial information can also be considered as an evidence of the relevance of tweets. In the case of a located event such as natural disaster, a tweet posted inside of the area impacted by the catastrophe is likely more relevant than tweet posted outside the area of interest.

The main issue that challenges the introduction of geospatial diversity in tweet summarization is the localization of a tweet. Indeed, the location information of a tweet can be identified using two different sources: (1) Accurately through the geotagging feature available on Twitter if the user has chosen to provide location information for the tweets he publishes using a smartphone with GPS capabilities. (2) Approximately using the location in the user's profile. However, only a very small portion of the tweets is geolocated (approximately 1% of all tweets published on Twitter). Hence, we are often settled to use the location information provided in the profile of the user. This information is supposed to be the name of the city where the user lives. To be able to exploit the location filed in the user's profile, it is necessary to translate it into geographic coordinates. While this information can be useful in some scenarios (such as a summarization of the reaction of people during a political event), it can be less useful in the case of monitoring tweets related to an emergency in a specific geographic area, to aid situational awareness. The use of location filed in the user's profile does not guarantee that the tweet was posted in the geographical area of the interest.

- *User diversity:* The user may be more inserted by tweets posted by various sources which may include an on the one hand official accounts of organizations, traditional media, and celebrates and on the other hand accounts of simple users (ordinary people). The inclusion of tweets posted by different sources may allow representing different viewpoints in the summary. This can be achieved by two different strategies. The first one is a simple user-account oriented strategy that consists on limiting the number of tweets posted by the same user-account in the summary. The second one is a user-account class oriented strategy in which user-accounts are classified into different classes. Then the restriction of the number of tweets is applied to each class such as the number of tweets published by user-accounts belonging to the same class is bounded. In this context, both supervised and unsupervised learning techniques may be investigated. The main issue challenging this task is the fact that classification should be done in real-time which means that we are limited to use features that are provided with the metadata of tweets.
- *Viewpoint diversity:* In many scenarios such as political issue, it is desired that the summary includes tweets that express different viewpoints and/or sentiments. The use of the viewpoint discovery model and sentiment analyses could enhance the quality of the summary. It will be interesting to consider the use of sentiment analyses to classify tweets to one of the following classes: tweets that express positive emotion, negative emotions or that express a fact or not express any emotions. This leads to ensure that the summary includes tweets that express different sentiments or viewpoints.

BIBLIOGRAPHY

[1] Nitin Agarwal and Yusuf Yiliyasi. Information quality challenges in social media. In *Proceedings of the 15th International Conference on Information Quality, ICIQ 2010, Little Rock, AR, USA, November 12-14, 2010.*, pages 234–248, 2010.

[2] Thomas Aichner and Frank Jacob. Measuring the degree of corporate social media use. *International Journal of Market Research*, 57(2):257–276, 2015. doi: 10.2501/IJMR-2015-018. URL https://doi.org/10.2501/IJMR-2015-018.

[3] Arifah Che Alhadi, Thomas Gottron, Jérôme Kunegis, and Nasir Naveed. Livetweet: Monitoring and predicting interesting microblog posts. In *Advances in Information Retrieval - 34th European Conference on IR Research, ECIR 2012, Barcelona, Spain, April 1-5, 2012. Proceedings*, pages 569–570, 2012. doi: 10.1007/978-3-642-28997-2_66. URL https://doi.org/10.1007/978-3-642-28997-2_66.

[4] James Allan, editor. *Topic Detection and Tracking: Event-based Information Organization*. Kluwer Academic Publishers, Norwell, MA, USA, 2002. ISBN 0-7923-7664-1.

[5] James Allan, Stephen Harding, David Fisher, Alvaro Bolivar, Sergio Guzman-Lara, and Peter Amstutz. Taking topic detection from evaluation to practice. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 4 - Volume 04*, HICSS '05, pages 101.1–, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2268-8-4. doi: 10.1109/HICSS.2005.576. URL http://dx.doi.org/10.1109/HICSS.2005.576.

[6] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. 31:211–236, 05 2017.

[7] Nasser Alsaedi, Pete Burnap, and Omer F. Rana. Automatic summarization of real world events using twitter. In *Proceedings of the Tenth International Conference on Web and Social Media, Cologne, Germany, May 17-20, 2016.*, pages 511–514, 2016. URL http://www.aaai.org/ocs/index.php/ICWSM/ICWSM16/paper/view/13017.

[8] Hughes Amanda Lee and Palen Leysia. Twitter adoption and use in mass convergence and emergency events. *International Journal of Emergency Management*, 6(3): 248–260, 2009. ISSN 0824-7935.

[9] Gianni Amati, Giuseppe Amodeo, Marco Bianchi, Giuseppe Marcone, Fondazione Ugo Bordoni, Carlo Gaibisso, Giorgio Gambosi, Alessandro Celi, Cesidio Di Nicola, and Michele Flammini. Fub, iasi-cnr, UNIVAQ at TREC 2011 microblog track. In *Proceedings of The Twentieth Text REtrieval Conference, TREC 2011, Gaithersburg, Maryland, USA, November 15-18, 2011*, 2011. URL http://trec.nist.gov/pubs/trec20/papers/FUB.microblog.update.pdf.

[10] Sihem Amer-Yahia, Michael Benedikt, and Philip Bohannon. Challenges in searching online communities. *IEEE Data Eng. Bull.*, 30(2):23–31, 2007. URL http://sites.computer.org/debull/A07june/sihem.pdf.

[11] Enrique Amigó, Adolfo Corujo, Julio Gonzalo, Edgar Meij, and Maarten de Rijke. Overview of RepLab 2012: Evaluating online reputation management systems. In *CLEF 2012: working notes for CLEF 2012 Conference: Rome, Italy, September 17-20, 2012*. CEUR, 2012.

[12] Enrique Amigó, Jorge Carrillo de Albornoz, Irina Chugur, Adolfo Corujo, Julio Gonzalo, Tamara Martín, Edgar Meij, Maarten de Rijke, editor="Forner Pamela Spina, Damiano", Henning Müller, Roberto Paredes, Paolo Rosso, and Benno Stein. Overview of replab 2013: Evaluating online reputation monitoring systems. In *Information Access Evaluation. Multilinguality, Multimodality, and Visualization*, pages 333–352, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[13] Nima Asadi and Jimmy Lin. Fast candidate generation for real-time tweet search with bloom filter chains. *ACM Trans. Inf. Syst.*, 31(3):13:1–13:36, August 2013. ISSN 1046-8188. doi: 10.1145/2493175.2493178. URL http://doi.acm.org/10.1145/2493175.2493178.

[14] Javed A. Aslam, Matthew Ekstrand-Abueg, Virgil Pavlu, Fernando Diaz, Richard McCreadie, and Tetsuya Sakai. Trec 2014 temporal summarization track overview. In *Proceedings of The Twenty-Third Text REtrieval Conference, TREC 2014, Gaithersburg, Maryland, USA, November 19-21, 2014*, 2014. URL http://trec.nist.gov/pubs/trec23/papers/overview-tempsumm.pdf.

[15] Javed A. Aslam, Fernando Diaz, Matthew Ekstrand-Abueg, Richard McCreadie, Virgil Pavlu, and Tetsuya Sakai. Trec 2015 temporal summarization track overview. In *Proceedings of The Twenty-Fourth Text REtrieval Conference, TREC 2015, Gaithersburg, Maryland, USA, November 17-20, 2015*, 2015. URL http://trec.nist.gov/pubs/trec24/papers/Overview-TS.pdf.

[16] Farzindar Atefeh and Wael Khreich. A survey of techniques for event detection in twitter. *Comput. Intell.*, 31(1):132–164, February 2015. ISSN 0824-7935. doi: 10.1111/coin.12017. URL http://dx.doi.org/10.1111/coin.12017.

[17] Moshe Babaioff, Michael Dinitz, Anupam Gupta, Nicole Immorlica, and Kunal Talwar. Secretary problems: Weights and discounts. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '09, pages 1245–1254, Philadelphia, PA, USA, 2009. Society for Industrial and Applied Mathematics. URL http://dl.acm.org/citation.cfm?id=1496770.1496905.

[18] Ismail Badache and Mohand Boughanem. A priori relevance based on quality and diversity of social signals. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, pages 731–734, 2015. doi: 10.1145/2766462.2767807. URL http://doi.acm.org/10.1145/2766462.2767807.

[19] Ismail Badache and Mohand Boughanem. Fresh and diverse social signals: Any impacts on search? In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*, CHIIR '17, pages 155–164, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4677-1. doi: 10.1145/3020165.3020177. URL http://doi.acm.org/10.1145/3020165.3020177.

[20] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval: The Concepts and Technology Behind Search*. Addison-Wesley Publishing Company, USA, 2nd edition, 2008. ISBN 9780321416919.

[21] Mossaab Bagdouri and Douglas W. Oard. CLIP at TREC 2015: Microblog and liveqa. In *Proceedings of The Twenty-Fourth Text REtrieval Conference, TREC 2015, Gaithersburg, Maryland, USA, November 17-20, 2015*, 2015. URL http://trec.nist.gov/pubs/trec24/papers/CLIP-MBQA.pdf.

[22] Nicholas J. Belkin and W. Bruce Croft. Information filtering and information retrieval: Two sides of the same coin? *Commun. ACM*, 35(12):29–38, December 1992. ISSN 0001-0782. doi: 10.1145/138859.138861. URL http://doi.acm.org/10.1145/138859.138861.

[23] Lamjed Ben Jabeur, Lynda Tamine, and Mohand Boughanem. Featured tweet search: Modeling time and social influence for microblog retrieval. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '12, pages 166–173, Washington, DC, USA, 2012. IEEE Computer Society. ISBN 978-0-7695-4880-7. URL http://dl.acm.org/citation.cfm?id=2457524.2457678.

[24] Lamjed Ben Jabeur, Lynda Tamine, and Mohand Boughanem. Active microbloggers: Identifying influencers, leaders and discussers in microblogging networks (short paper). In *Symposium on String Processing and Information Retrieval (SPIRE), Cartagena, Colombia, 21/10/2012-25/10/2012*, volume 7608, pages 111–117, http://www.springer.com, octobre 2012. Springer Berlin / Heidelberg.

[25] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324. URL http://dx.doi.org/10.1023/A:1010933404324.

[26] Cody Buntain and Jimmy Lin. Burst detection in social media streams for tracking interest profiles in real time. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 777–780, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4069-4. doi: 10.1145/2911451.2914733. URL http://doi.acm.org/10.1145/2911451.2914733.

[27] H. Cai, Z. Huang, D. Srivastava, and Q. Zhang. Indexing evolving events from tweet streams. *IEEE Transactions on Knowledge and Data Engineering*, 27(11):3001–3015, Nov 2015. ISSN 1041-4347. doi: 10.1109/TKDE.2015.2445773.

[28] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*, pages 129–136, 2007. doi: 10.1145/1273496.1273513. URL http://doi.acm.org/10.1145/1273496.1273513.

[29] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 335–336, New York, NY, USA, 1998. ACM. ISBN

1-58113-015-5. doi: 10.1145/290941.291025. URL http://doi.acm.org/10.1145/290941.291025.

[30] Deepayan Chakrabarti and Kunal Punera. Event summarization using tweets. In *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011*, 2011. URL http://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/view/2885.

[31] Yi Chang, Anlei Dong, Pranam Kolari, Ruiqiang Zhang, Yoshiyuki Inagaki, Fernanodo Diaz, Hongyuan Zha, and Yan Liu. Improving recency ranking using twitter data. *ACM Trans. Intell. Syst. Technol.*, 4(1):4:1–4:24, February 2013. ISSN 2157-6904. doi: 10.1145/2414425.2414429. URL http://doi.acm.org/10.1145/2414425.2414429.

[32] Abdelhamid Chellal and Mohand Boughanem. Optimization framework model for retrospective tweet summarization. In *Proceedings of the ACM Symposium on Applied Computing (SAC), Pau, France, 09/04/18-13/04/18*.

[33] Abdelhamid Chellal, Mohand Boughanem, and Bernard Dousset. Multi-criterion real time tweet summarization based upon adaptive threshold. In *2016 IEEE/WIC/ACM , WI 2016, Omaha, NE, USA, October 13-16, 2016*, pages 264–271, 2016. doi: 10.1109/WI.2016.0045. URL http://dx.doi.org/10.1109/WI.2016.0045.

[34] Abdelhamid Chellal, Mohand Boughanem, and Bernard Dousset. Word similarity based model for tweet stream prospective notification. In *Advances in Information Retrieval - 39th European Conference on IR Research, ECIR 2017, Aberdeen, UK, April 8-13, 2017, Proceedings*, pages 655–661, 2017. doi: 10.1007/978-3-319-56608-5_62.

[35] Fuxing Cheng, Xin Zhang, Ben He, Tiejian Luo, and Wenjie Wang. A survey of learning to rank for real-time twitter search. In Qiaohong Zu, Bo Hu, and Atilla Elçi, editors, *Pervasive Computing and the Networked World*, pages 150–164, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[36] Tao Cheng and Thomas Wicks. Event detection using twitter: A spatio-temporal approach. *PLOS ONE*, 9(6):1–10, 06 2014. doi: 10.1371/journal.pone.0097807. URL https://doi.org/10.1371/journal.pone.0097807.

[37] Robert Collier. Robert collier quotes. URL http://www.brainyquote.com/quotes/quotes/r/robertcoll108959.html.

[38] Firas Damak, Lamjed Ben Jabeur, Guillaume Cabanac, Karen Pinel-Sauvagnat, Lynda Tamine, and Mohand Boughanem. IRIT at TREC microblog 2011. In *Proceedings of The Twentieth Text REtrieval Conference, TREC 2011, Gaithersburg, Maryland, USA, November 15-18, 2011*, 2011. URL http://trec.nist.gov/pubs/trec20/papers/IRIT_SIG.microblog.update.pdf.

[39] Firas Damak, Karen Pinel-Sauvagnat, Guillaume Cabanac, and Mohand Boughanem. Effectiveness of state-of-the-art features for microblog search. In *SAC-IAR 2013*, pages 914–919, G, April 2013. ACM.

[40] Dipanjan Das and André F. T. Martins. A survey on automatic text summarization. *Literature Survey for the Language and Statistics II course at CMU*, 3:192–195, 2007.

[41] Felicia Day. Felicia day quotes. URL http://www.brainyquote.com/quotes/quotes/f/feliciaday561466.html.

[42] Fernando Diaz, Bhaskar Mitra, and Nick Craswell. Query expansion with locally-trained word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, pages 367—-377, 2016. URL http://aclweb.org/anthology/P/P16/P16-1035.pdf.

[43] Anlei Dong, Ruiqiang Zhang, Pranam Kolari, Jing Bai, Fernando Diaz, Yi Chang, Zhaohui Zheng, and Hongyuan Zha. Time is of the essence: improving recency ranking using twitter data. In *Proceedings of the 19th international conference on World wide web, WWW '10*, pages 331–340, New York, NY, USA, 2010. ACM.

[44] Jack Dorsey. Quotes about twitter. URL http://www.wordstream.com/blog/ws/2016/02/26/social-media-quote.

[45] Yajuan Duan, Long Jiang, Tao Qin, Ming Zhou, and Heung-Yeung Shum. An empirical study on learning to rank of tweets. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 295–303, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1873781.1873815.

[46] Yajuan Duan, Zhimin Chen, Furu Wei, Ming Zhou, and Heung-Yeung Shum. Twitter topic summarization by ranking tweets using social influence and content quality. In *COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 8-15 December 2012, Mumbai, India*, pages 763–780, 2012. URL http://aclweb.org/anthology/C/C12/C12-1047.pdf.

[47] Paul S. Earle, Daniel C. Bowden, and Guy Michelle. Twitter earthquake detection: Earthquake monitoring in a social world. 54, 01 2012.

[48] Günes Erkan and Dragomir R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479, December 2004. ISSN 1076-9757. URL http://dl.acm.org/citation.cfm?id=1622487.1622501.

[49] Feifan Fan, Yue Fei, Chao Lv, Lili Yao, Jianwu Yang, and Dongyan Zhao. Pkuicst at trec 2015 microblog track: Query-biased adaptive filtering in real-time microblog stream. In *Text REtrieval Conference, TREC, Gaithersburg, USA, November 17-20, 2015.*

[50] Anjie Fang, Iadh Ounis, Philip Habel, Craig Macdonald, and Nut Limsopatham. Topic-centric classification of twitter user's political orientation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 791–794, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3621-5. doi: 10.1145/2766462.2767833. URL http://doi.acm.org/10.1145/2766462.2767833.

[51] Paul Ferguson, Neil O'Hare, James Lanagan, Owen Phelan, and Kevin McCarthy. An investigation of term weighting approaches for microblog retrieval. In *Proceedings of the 34th European Conference on Advances in Information Retrieval*, ECIR'12, pages 552–555, Berlin, Heidelberg, 2012. Springer-Verlag. ISBN 978-3-642-28996-5. doi: 10.1007/978-3-642-28997-2_62. URL http://dx.doi.org/10.1007/978-3-642-28997-2_62.

[52] Ronald T. Fernández. The effect of smoothing in language models for novelty detection. In *Proceedings of the BCS IRSG Symposium: Future Directions in Information Access (FDIA 2007)*, FDIA 2007, pages 11–16, 2007. URL http://www.bcs.org/upload/pdf/ewic_fd07_paper2.pdf.

[53] J. G Fiscus and G. R. Doddington. Topic detection and tracking evaluation overview. In James Allan, editor, *Topic detection and tracking*, chapter 1, pages 17–31. Springer US, 2002.

[54] Malcolm Forbes. Conclusion quotes. URL https://www.brainyquote.com/quotes/keywords/conclusion.html.

[55] Ophélie Fraisier, Guillaume Cabanac, Yoann Pitarch, Romaric Besancon, and Mohand Boughanem. Stance classification through proximity-based community detection. In *ACM Conference on Hypertext and Social Media, Baltimore, Maryland, USA, 09/07/2018-12/07/2018*, http://www.acm.org/, juillet 2018. ACM.

[56] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. Quantifying controversy on social media. *Trans. Soc. Comput.*, 1 (1):3:1–3:27, January 2018. ISSN 2469-7818. doi: 10.1145/3140565. URL http://doi.acm.org/10.1145/3140565.

[57] Thibault Gisselbrecht, Sylvain Lamprier, and Patrick Gallinari. Dynamic data capture from social media streams: A contextual bandit approach. In *Proceedings of the Tenth International Conference on Web and Social Media, Cologne, Germany, May 17-20, 2016.*, pages 131–140, 2016. URL http://www.aaai.org/ocs/index.php/ICWSM/ICWSM16/paper/view/13084.

[58] Qi Guo, Fernando Diaz, and Elad Yom-Tov. Updating users about time critical events. In *Proceedings of the 35th European Conference on Advances in Information Retrieval*, ECIR'13, pages 483–494, Berlin, Heidelberg, 2013. Springer-Verlag. ISBN 978-3-642-36972-8. doi: 10.1007/978-3-642-36973-5_41. URL http://dx.doi.org/10.1007/978-3-642-36973-5_41.

[59] Wenjie Li Haihui Tan, Dajun Luo. Polyu at trec 2016 real-time summarization. In *Proceedings of The Twenty-Five Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*, 2016.

[60] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009. ISSN 1931-0145. doi: 10.1145/1656274.1656278. URL http://doi.acm.org/10.1145/1656274.1656278.

[61] Zhongyuan Han, Song Li, Leilei Kong, Liuyang Tian, and Haoliang Qi. Hljit at trec 2017 real-time summarization. In *Proceedings of The Twenty-Six Text REtrieval Conference, TREC 2017, Gaithersburg, Maryland, USA, November 15-17, 2017*, 2017.

[62] Mahmud Hasan, Mehmet A Orgun, and Rolf Schwitter. A survey on real-time event detection from the twitter data stream. *Journal of Information Science*, 2017. doi: 10.1177/0165551517698564. URL https://doi.org/10.1177/0165551517698564.

[63] Ruifang He, Yang Liu, Guangchuan Yu, Jiliang Tang, Qinghua Hu, and Jianwu Dang. Twitter summarization with social-temporal context. *World Wide Web*, 20 (2):267–290, March 2017. ISSN 1386-145X. doi: 10.1007/s11280-016-0386-0. URL https://doi.org/10.1007/s11280-016-0386-0.

[64] Juraj Hromkovic and Waldyr M. Oliva. *Algorithmics for Hard Problems*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2nd edition, 2002. ISBN 3540441344.

[65] Mengdie Hu, Shixia Liu, Furu Wei, Yingcai Wu, John Stasko, and Kwan-Liu Ma. Breaking news on twitter. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2751–2754, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2208672. URL http://doi.acm.org/10.1145/2207676.2208672.

[66] Gilles Hubert, José Moreno, Karen Pinel-Sauvagnat, and Yoann Pitarch. Everything You Always Wanted to Know About TREC RTS* (*But Were Afraid to Ask). Diffusion scientifique, décembre 2017. URL https://arxiv.org/abs/1712.04671.

[67] David Inouye and Jugal K. Kalita. Comparing twitter summarization algorithms for multiple post summaries. In *PASSAT/SocialCom 2011, Privacy, Security, Risk and Trust (PASSAT), 2011 IEEE Third International Conference on Social Computing (SocialCom), Boston, MA, USA, 9-11 Oct., 2011*, pages 298–306, 2011. doi: 10.1109/PASSAT/SocialCom.2011.31. URL http://dx.doi.org/10.1109/PASSAT/SocialCom.2011.31.

[68] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, October 2002. ISSN 1046-8188. doi: 10.1145/582415.582418. URL http://doi.acm.org/10.1145/582415.582418.

[69] Jimmy Lin Jiaul H. Paik. Do multiple listeners to the public twitter sample stream receive the same tweets? In *Proceedings of the SIGIR 2015 Workshop on Temporal, Social and Spatially-Aware Information Access*, SIGIR '15, 2015.

[70] George H. John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, pages 338–345, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1-55860-385-9. URL http://dl.acm.org/citation.cfm?id=2074158.2074196.

[71] Joel Judd and Jugal Kalita. Better twitter summaries? In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 445–449. Association for Computational Linguistics, 2013. URL http://www.aclweb.org/anthology/N13-1047.

[72] Margarita Karkali, François Rousseau, Alexandros Ntoulas, and Michalis Vazirgiannis. Using temporal IDF for efficient novelty detection in text streams. *CoRR*, abs/1401.1456, 2014. URL http://arxiv.org/abs/1401.1456.

[73] Muhammad Asif Hossain Khan, Danushka Bollegala, Guangwen Liu, and Kaoru Sezaki. Multi-tweet summarization of real-time events. In *2013 International Conference on Social Computing*, pages 128–133, Sept 2013. doi: 10.1109/SocialCom.2013.26.

[74] Lars Kirchhoff, Katarina Stanoevska-Slabeva, Thomas Nicolai, and Matthes Fleck. Using social network analysis to enhance information retrieval systems. In *5th Applications of Social Network Analysis (ASNA)*, September 2008. URL https://www.alexandria.unisg.ch/46444/.

[75] Marijn Koolen, Gabriella Kazai, and Nick Craswell. Wikipedia pages as entry points for book search. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 44–53, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-390-7. doi: 10.1145/1498759.1498807. URL http://doi.acm.org/10.1145/1498759.1498807.

[76] Eleni Koutrouli and Aphrodite Tsalgatidou. Reputation systems evaluation survey. *ACM Comput. Surv.*, 48(3):35:1–35:28, December 2015. ISSN 0360-0300. doi: 10.1145/2835373. URL http://doi.acm.org/10.1145/2835373.

[77] S. Kullback and R. A. Leibler. *The Annals of Mathematical Statistics*, (1):79–86, 03 .

[78] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 591–600, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-799-8. doi: 10.1145/1772690.1772751. URL http://doi.acm.org/10.1145/1772690.1772751.

[79] Alex Leavitt, Evan Burchard, David Fisher, and Sam Gilbert. The influentials: New approaches for analyzing influence on twitter. *Webecology Project*, September 2009.

[80] Kathy Lee, Ashequl Qadir, Yuan Ling, Joey Liu, Sadid A. Hasan, Vivek Datla, Aaditya Prakash, and Oladimeji Farri. Recognizing tweet relevance with profile-specific and profile-independent supervised models. In *Proceedings of The Twenty-Six Text REtrieval Conference, TREC 2017, Gaithersburg, Maryland, USA, November 15-17, 2017*, 2017.

[81] Peng Li, Yinglin Wang, Wei Gao, and Jing Jiang. Generating aspect-oriented multi-document summarization with event-aspect model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1137–1146, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-11-4. URL http://dl.acm.org/citation.cfm?id=2145432.2145553.

[82] Chin-Yew Lin. Rouge: a package for automatic evaluation of summaries. July 2004. URL https://www.microsoft.com/en-us/research/publication/rouge-a-package-for-automatic-evaluation-of-summaries/.

[83] Chin-Yew Lin and Eduard Hovy. Manual and automatic evaluation of summaries. In *Proceedings of the ACL-02 Workshop on Automatic Summarization - Volume 4*, AS '02, pages 45–51, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118162.1118168. URL https://doi.org/10.3115/1118162.1118168.

[84] Jimmy Lin, Miles Efron, Yulu Wang, and Garrick Sherman. Overview of the trec-2014 microblog track. In *Proceedings of The Twenty-Third Text REtrieval Conference, TREC 2014, Gaithersburg, Maryland, USA, November 19-21, 2014*, 2014.

[85] Jimmy Lin, Miles Efron, Yulu Wang, Garrick Sherman, Richard McCreadie, and Tetsuya Sakai. Overview of the trec 2015 microblog track. In *Text REtrieval Conference, TREC, Gaithersburg, USA, November 17-20*, 2015.

[86] Jimmy Lin, Adam Roegiest, Luchen Tan, Richard McCreadie, Ellen Voorhees, and Fernando Diaz. Overview of the trec 2016 realtime summarization. In *Proceedings of The Twenty-Five Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*, 2016.

[87] Jimmy Lin, Salman Mohammed, Royal Sequiera, Luchen Tan, Nimesh Ghelani, Richard McCreadie Mustafa Abualsaud, Dmitrijs Milajevs, and Ellen Voorhees. Overview of the trec 2017 real-time summarization track. In *Proceedings of The Twenty-Six Text REtrieval Conference, TREC 2017, Gaithersburg, Maryland, USA, November 15-17, 2017*, 2017.

[88] Fei Liu, Yang Liu, and Fuliang Weng. Why is "sxsw" trending?: Exploring multiple text sources for twitter topic summarization. In *Proceedings of the Workshop on Languages in Social Media*, LSM '11, pages 66–75, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-96-1. URL http://dl.acm.org/citation.cfm?id=2021109.2021118.

[89] Kun-Lin Liu, Wu-Jun Li, and Minyi Guo. Emoticon smoothed language models for twitter sentiment analysis. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, pages 1678–1684. AAAI Press, 2012. URL http://dl.acm.org/citation.cfm?id=2900929.2900966.

[90] Xiaohua Liu, Yitong Li, Furu Wei, and Ming Zhou. Graph-based multi-tweet summarization using social signals. In *Proceedings of COLING 2012*, pages 1699–1714, Mumbai, India, December 2012. The COLING 2012 Organizing Committee. URL http://www.aclweb.org/anthology/C12-1104.

[91] Charles L. A. Clarke Jimmy Lin Luchen Tan, Adam Roegiest. Simple dynamic emission strategies for microblog filtering. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 745–754, 2016. ISBN 978-1-4503-4069-4. doi: 2911451. URL http://dx.doi.org/10.1145/2911451.2914694.

[92] Jimmy Lin Charles L. A. Clarke Luchen Tan, Adam Roegiest. An exploration of evaluation metrics for mobile push notifications. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*,

SIGIR '16, pages 745–754, 2016. ISBN 978-1-4503-4069-4. doi: 2911451. URL http://dx.doi.org/10.1145/2911451.2914694.

[93] H. P. Luhn. The automatic creation of literature abstracts. *IBM J. Res. Dev.*, 2 (2):159–165, April 1958. ISSN 0018-8646. doi: 10.1147/rd.22.0159. URL http://dx.doi.org/10.1147/rd.22.0159.

[94] Zhunchen Luo, Miles Osbornez, Saša Petrovic, and Ting Wang. Improving twitter retrieval by exploiting structural information. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, pages 648–654. AAAI Press, 2012. URL http://dl.acm.org/citation.cfm?id=2900728.2900821.

[95] Zhunchen Luo, Miles Osborne, and Ting Wang. An effective approach to tweets opinion retrieval. *World Wide Web*, 18(3):545–566, May 2015. ISSN 1386-145X. doi: 10.1007/s11280-013-0268-7. URL http://dx.doi.org/10.1007/s11280-013-0268-7.

[96] Craig Macdonald and Iadh Ounis. Voting techniques for expert search. *Knowl. Inf. Syst.*, 16(3):259–280, August 2008. ISSN 0219-1377. doi: 10.1007/s10115-007-0105-3. URL http://dx.doi.org/10.1007/s10115-007-0105-3.

[97] Stuart Mackie, Richard McCreadie, Craig Macdonald, and Iadh Ounis. Comparing algorithms for microblog summarisation. In *Information Access Evaluation. Multilinguality, Multimodality, and Interaction - 5th International Conference of the CLEF Initiative, CLEF 2014, Sheffield, UK, September 15-18, 2014. Proceedings*, pages 153–159, 2014. doi: 10.1007/978-3-319-11382-1_15.

[98] Matteo Magnani, Danilo Montesi, and Luca Rossi. Conversation retrieval for microblogging sites. *Inf. Retr.*, 15(3-4):354–372, June 2012. ISSN 1386-4564. doi: 10.1007/s10791-012-9189-9. URL http://dx.doi.org/10.1007/s10791-012-9189-9.

[99] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. ISBN 0521865719, 9780521865715.

[100] A. H. Maslow and K. J. Lewis. Maslow's hierarchy of needs. In *Salenger Incorporated*, 1987.

[101] Kamran Massoudi, Manos Tsagkias, Maarten de Rijke, and Wouter Weerkamp. Incorporating query expansion and quality indicators in searching microblog posts. In Paul Clough, Colum Foley, Cathal Gurrin, Gareth J. F. Jones, Wessel Kraaij, Hyowon Lee, and Vanessa Mudoch, editors, *Advances in Information Retrieval*, pages 362–367, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[102] Richard McCreadie and Craig Macdonald. Relevance in microblogs: Enhancing tweet retrieval using hyperlinked documents. In *Proceedings of the 10th Conference on Open Research Areas in Information Retrieval*, OAIR '13, pages 189–196, Paris, France, France, 2013. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE. ISBN 978-2-905450-09-8. URL http://dl.acm.org/citation.cfm?id=2491748.2491787.

[103] Richard McCreadie, Craig Macdonald, and Iadh Ounis. Incremental update summarization: Adaptive sentence selection based on prevalence and novelty. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 301–310, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2598-1. doi: 10.1145/2661829.2661951. URL http://doi.acm.org/10.1145/2661829.2661951.

[104] Ryan T. McDonald. A study of global inference algorithms in multi-document summarization. In *Advances in Information Retrieval, 29th European Conference on IR Research, ECIR 2007, Rome, Italy, April 2-5, 2007, Proceedings*, pages 557–564, 2007. doi: 10.1007/978-3-540-71496-5_51. URL http://dx.doi.org/10.1007/978-3-540-71496-5_51.

[105] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *EMNLP ACL Processing*, page 404–411, 07 2004.

[106] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[107] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA, 2013. Curran Associates Inc. URL http://dl.acm.org/citation.cfm?id=2999792.2999959.

[108] Douglas W. Oard Mossaab Bagdouri. Clip at trec 2016: Liveqa and rts. In *Proceedings of The Twenty-Five Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*, 2016.

[109] R. Nagmoti, A. Teredesai, and M. De Cock. Ranking approaches for microblog search. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 153–157, 2010. doi: 10.1109/WI-IAT.2010.170.

[110] A. Nenkova and L. Vanderwende. The impact of frequency on summarization. Technical Report MSR-TR-2005-101, MSR-TR-2005-101, January 2005.

[111] Ani Nenkova and Kathleen McKeown. Automatic summarization. *Foundations and Trends in Information Retrieval*, 5(2–3):103–233, 2011. ISSN 1554-0669. doi: 10.1561/1500000015. URL http://dx.doi.org/10.1561/1500000015.

[112] Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. In William W. Cohen and Samuel Gosling, editors, *ICWSM*. The AAAI Press, 2010. URL http://dblp.uni-trier.de/db/conf/icwsm/icwsm2010.html#OConnorBRS10.

[113] Andrei Olariu. Hierarchical clustering in improving microblog stream summarization. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[114] Andrei Olariu. Efficient online summarization of microblogging streams. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 236–240, Gothenburg, Sweden, April 2014. Association for Computational Linguistics.

[115] Iadh Ounis, Craig Macdonald, Jimmy Lin, and Ian Soboroff. Overview of the trec 2011 microblog track. In *Proceedings of The Twentieth Text REtrieval Conference, TREC 2011, Gaithersburg, Maryland, USA, November 15-18, 2011*. National Institute of Standards and Technology (NIST), 2011.

[116] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. URL http://ilpubs.stanford.edu:8090/422/. Previous number = SIDL-WP-1999-0120.

[117] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 275–281, New York, NY, USA, 1998. ACM. ISBN 1-58113-015-5. doi: 10.1145/290941.291008. URL http://doi.acm.org/10.1145/290941.291008.

[118] Ana-Maria Popescu and Marco Pennacchiotti. Detecting controversial events from twitter. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1873–1876, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0099-5. doi: 10.1145/1871437.1871751. URL http://doi.acm.org/10.1145/1871437.1871751.

[119] M. F. Porter. Readings in information retrieval. chapter An Algorithm for Suffix Stripping, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997. ISBN 1-55860-454-5. URL http://dl.acm.org/citation.cfm?id=275537.275705.

[120] Xin Qian, Jimmy Lin, and Adam Roegiest. Interleaved evaluation for retrospective summarization and prospective notification on document streams. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 175–184, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4069-4. doi: 10.1145/2911451.2911494. URL http://doi.acm.org/10.1145/2911451.2911494.

[121] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN 1-55860-238-0.

[122] Tamer Elsayed Reem Suwaileh, Maram Hasanain. Light-weight, conservative, yet effective: Scalable real-time tweet summarization. In *Proceedings of The Twenty-Five Text REtrieval Conference, TREC 2016, Gaithersburg, Maryland, USA, November 15-18, 2016*, 2016.

[123] Zhaochun Ren, Shangsong Liang, Edgar Meij, and Maarten de Rijke. Personalized time-aware tweets summarization. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13,

pages 513–522, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2034-4. doi: 10.1145/2484028.2484052. URL http://doi.acm.org/10.1145/2484028.2484052.

[124] Zhaochun Ren, Oana Inel, Lora Aroyo, and Maarten de Rijke. Time-aware multiviewpoint summarization of multilingual social text streams. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 387–396, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4073-1. doi: 10.1145/2983323.2983710. URL http://doi.acm.org/10.1145/2983323.2983710.

[125] Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. Named entity recognition in tweets: An experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1524–1534, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-11-4. URL http://dl.acm.org/citation.cfm?id=2145432.2145595.

[126] Stephen E. Robertson and Ian Soboroff. The TREC 2002 filtering track report. In *Proceedings of The Eleventh Text REtrieval Conference, TREC 2002, Gaithersburg, Maryland, USA, November 19-22, 2002*, 2002. URL http://trec.nist.gov/pubs/trec11/papers/OVER.FILTERING.pdf.

[127] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. In *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*, pages 109–126, 1994. URL http://trec.nist.gov/pubs/trec3/papers/city.ps.gz.

[128] Adam Roegiest, Luchen Tan, Jimmy Lin, and Charles L.A. Clarke. A platform for streaming push notifications to mobile assessors. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 1077–1080, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4069-4. doi: 10.1145/2911451.2911463. URL http://doi.acm.org/10.1145/2911451.2911463.

[129] Adam Roegiest, Luchen Tan, and Jimmy Lin. Online in-situ interleaved evaluation of real-time push notification systems. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, pages 415–424, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5022-8. doi: 10.1145/3077136.3080808. URL http://doi.acm.org/10.1145/3077136.3080808.

[130] Kevin Dela Rosa, Rushin Shah, Bo Lin, Anatole Gershman, and Robert E. Frederking. Topical clustering of tweets. 2011.

[131] Karankumar Sabhnani and Ben Carterette. University of delaware at trec 2017 real-time summarization track. In *Proceedings of The Twenty-Six Text REtrieval Conference, TREC 2017, Gaithersburg, Maryland, USA, November 15-17, 2017*, 2017.

[132] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, November 1975. ISSN 0001-0782. doi: 10.1145/361219.361220. URL http://doi.acm.org/10.1145/361219.361220.

[133] Gerard Salton. A comparison between manual and automatic indexing methods. Technical report, Ithaca, NY, USA, 1968.

[134] Gerard Salton and YANG C.S. On the specification of term values in automatic indexing. *Journal of Documentation*, 29(4):351–372, December 1973. doi: 10.1108/eb026562.

[135] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986. ISBN 0070544840.

[136] Gerard Salton, Edward A. Fox, and Harry Wu. Extended boolean information retrieval. *Commun. ACM*, 26(11):1022–1036, November 1983. ISSN 0001-0782. doi: 10.1145/182.358466. URL http://doi.acm.org/10.1145/182.358466.

[137] William Shakespeare. play hamlet, in the second act.

[138] David A. Shamma, Lyndon Kennedy, and Elizabeth F. Churchill. Peaks and persistence: Modeling the shape of microblog conversations. In *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work*, CSCW '11, pages 355–358, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0556-3. doi: 10.1145/1958824.1958878. URL http://doi.acm.org/10.1145/1958824.1958878.

[139] Beaux Sharifi, Mark anthony Hutton, and Jugal Kalita. Automatic summarization of twitter topics. In *in National Workshop on Design and Analysis of Algorithm*, 2010.

[140] Beaux Sharifi, Mark-Anthony Hutton, and Jugal Kalita. Summarizing microblogs automatically. In *Human Language Technologies: The Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 685–688, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 1-932432-65-5. URL http://dl.acm.org/citation.cfm?id=1857999.1858099.

[141] Beaux Sharifi, Mark-Anthony Hutton, and Jugal K. Kalita. Experiments in microblog summarization. In *Proceedings of the 2010 IEEE Second International Conference on Social Computing*, SOCIALCOM '10, pages 49–56, Washington, DC, USA, 2010. IEEE Computer Society. ISBN 978-0-7695-4211-9. doi: 10.1109/SocialCom.2010.17. URL http://dx.doi.org/10.1109/SocialCom.2010.17.

[142] Lidan Shou, Zhenhua Wang, Ke Chen, and Gang Chen. Sumblr: Continuous summarization of evolving tweet streams. In *the 36th International ACM SIGIR Conference*, SIGIR '13, pages 533–542, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2034-4. doi: 10.1145/2484028.2484045. URL http://doi.acm.org/10.1145/2484028.2484045.

[143] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. *SIGKDD Explor. Newsl.*, 19 (1):22–36, September 2017. ISSN 1931-0145. doi: 10.1145/3137597.3137600. URL http://doi.acm.org/10.1145/3137597.3137600.

[144] Reem Suwaileh and Tamer Elsayed. Exploiting live feedback for tweet real-time push notifications. In *Proceedings of The Twenty-Six Text REtrieval Conference, TREC 2017, Gaithersburg, Maryland, USA, November 15-17, 2017*, 2017.

[145] Reem Suwaileh, Maram Hasanain, Marwan Torki, and Tamer Elsayed. QU at TREC-2015: building real-time systems for tweet filtering and question answering. In *Proceedings of The Twenty-Fourth Text REtrieval Conference, TREC 2015, Gaithersburg, Maryland, USA, November 17-20, 2015*, 2015. URL http://trec.nist.gov/pubs/trec24/papers/QU-MBQA.pdf.

[146] Hiroya Takamura and Manabu Okumura. Text summarization model based on the budgeted median problem. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 1589–1592, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-512-3. doi: 10.1145/1645953.1646179. URL http://doi.acm.org/10.1145/1645953.1646179.

[147] Hiroya Takamura, Hikaru Yokono, and Manabu Okumura. Summarizing a document stream. In Paul Clough, Colum Foley, Cathal Gurrin, Gareth J. F. Jones, Wessel Kraaij, Hyowon Lee, and Vanessa Mudoch, editors, *Advances in Information Retrieval*, pages 177–188, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-20161-5.

[148] Sakaki Takeshi, Okazaki Makoto, and Matsuo Yutaka. Earthquake shakes twitter users: Real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 851–860, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-799-8. doi: 10.1145/1772690.1772777. URL http://doi.acm.org/10.1145/1772690.1772777.

[149] Luchen Tan, Adam Roegiest, and Charles L.A. Clarke. University of waterloo at trec 2015 microblog track. In *Text REtrieval Conference, TREC, Gaithersburg, USA, November 17-20*, 2015.

[150] Luchen Tan, Jimmy J. Lin, Adam Roegiest, and Charles L. A. Clarke. The effects of latency penalties in evaluating push notification systems. *CoRR*, abs/1606.03066, 2016. URL http://arxiv.org/abs/1606.03066.

[151] Jizhi Tang, Chao Lv, Lili Yao, and Dongyan Zhao. Pkuicst at trec 2017 real-time summarization track: Push notifications and email digest. In *Proceedings of The Twenty-Six Text REtrieval Conference, TREC 2017, Gaithersburg, Maryland, USA, November 15-17, 2017*, 2017.

[152] Ke Tao, Fabian Abel, Claudia Hauff, and Geert-Jan Houben. What makes a tweet relevant for a topic? volume 838 of *CEUR Workshop Proceedings*, pages 49–56. CEUR-WS.org, 2012.

[153] Thibaut Thonet, Guillaume Cabanac, Mohand Boughanem, and Karen Pinel-Sauvagnat. Users are known by the company they keep: Topic models for viewpoint discovery in social networks. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, pages 87–96, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4918-5. doi: 10.1145/3132847.3132897. URL http://doi.acm.org/10.1145/3132847.3132897.

[154] Arnout Verheij, Allard Kleijn, Flavius Frasincar, and Frederik Hogenboom. A comparison study for novelty control mechanisms applied to web news stories. In

2012 IEEE/WIC/ACM International Conferences on Web Intelligence, WI 2012, Macau, China, December 4-7, 2012, pages 431–436, 2012. doi: 10.1109/WI-IAT.2012.128. URL http://dx.doi.org/10.1109/WI-IAT.2012.128.

[155] Yana Volkovich and Andreas Kaltenbrunner. Evaluation of valuable user generated content on social news web sites. In *Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011 (Companion Volume)*, pages 139–140, 2011. doi: 10.1145/1963192.1963263. URL http://doi.acm.org/10.1145/1963192.1963263.

[156] Ellen M. Voorhees and Lori P. Buckland, editors. *Proceedings of The Twenty-First Text REtrieval Conference, TREC 2012, Gaithersburg, Maryland, USA, November 6-9, 2012*, volume Special Publication 500-298, 2012. National Institute of Standards and Technology (NIST). URL http://trec.nist.gov/pubs/trec21/t21.proceedings.html.

[157] Yulu Wang and Jimmy Lin. The impact of future term statistics in real-time tweet search. In *ECIR'2014*, pages 567–572. Springer International Publishing, April 2014.

[158] Mahdi Washha, Aziz Qaroush, Manel Mezghani, and Florence Sèdes. Information quality in social networks: Predicting spammy naming patterns for retrieving twitter spam accounts. In *ICEIS 2017 - Proceedings of the 19th International Conference on Enterprise Information Systems, Volume 2, Porto, Portugal, April 26-29, 2017*, pages 610–622, 2017. doi: 10.5220/0006314006100622. URL https://doi.org/10.5220/0006314006100622.

[159] Stanley Wasserman and Katherine Faust. *Social network analysis: Methods and applications*. Cambridge University Press, 1994. URL https://doi.org/10.1017/CBO9780511815478.

[160] W. Xie, F. Zhu, J. Jiang, E. P. Lim, and K. Wang. Topicsketch: Real-time bursty topic detection from twitter. *IEEE Transactions on Knowledge and Data Engineering*, 28(8):2216–2229, Aug 2016. ISSN 1041-4347. doi: 10.1109/TKDE.2016.2556661.

[161] Wei Xu, Ralph Grishman, Adam Meyers, and Alan Ritter. A preliminary study of tweet summarization using information extraction. In *Proceedings of the Workshop on Language Analysis in Social Media*, pages 20–29, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W13-1103.

[162] Yuto Yamaguchi, Tsubasa Takahashi, Toshiyuki Amagasa, and Hiroyuki Kitagawa. Turank: Twitter user ranking based on user-tweet graph analysis. In *Proceedings of the 11th International Conference on Web Information Systems Engineering*, WISE'10, pages 240–253, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-17615-1, 978-3-642-17615-9. URL http://dl.acm.org/citation.cfm?id=1991336.1991364.

[163] Jin-Ge Yao, Xiaojun Wan, and Jianguo Xiao. Recent advances in document summarization. *Knowl. Inf. Syst.*, 53(2):297–336, November 2017. ISSN 0219-1377. doi: 10.1007/s10115-017-1042-4. URL https://doi.org/10.1007/s10115-017-1042-4.

[164] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 334–342, New York, NY, USA, 2001. ACM. ISBN 1-58113-331-6. doi: 10.1145/383952.384019. URL http://doi.acm.org/10.1145/383952.384019.

[165] Lulin Zhao, Yi Zeng, and Ning Zhong. A weighted multi-factor algorithm for microblog search. In *Proceedings of the 7th International Conference on Active Media Technology*, AMT'11, pages 153–161, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-23619-8. URL http://dl.acm.org/citation.cfm?id=2033896.2033919.

[166] Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. Comparing twitter and traditional media using topic models. In *Advances in Information Retrieval - 33rd European Conference on IR Research, ECIR 2011, Dublin, Ireland, April 18-21, 2011. Proceedings*, pages 338–349, 2011. doi: 10.1007/978-3-642-20161-5_34. URL https://doi.org/10.1007/978-3-642-20161-5_34.

[167] Lei Zheng and Kai Han. Multi topic distribution model for topic discovery in twitter. In *2013 IEEE Seventh International Conference on Semantic Computing*, pages 420–425, Sept 2013. doi: 10.1109/ICSC.2013.81.

[168] Wang Zhenhua, Shou Lidan, Chen Ke, Chen Gang, and Mehrotra Sharad. On summarization and timeline generation for evolutionary tweet streams. *IEEE Trans. Knowl. Data Eng.*, 27(5):1301–1315, 2015.

[169] Bolong Zhu, Jinghua Gao, Xiao Han, Cunhui Shi, Shenghua Liu, Yue Liu, and Xueqi Cheng. ICTNET at microblog track TREC 2012. In *Proceedings of The Twenty-First Text REtrieval Conference, TREC 2012, Gaithersburg, Maryland, USA, November 6-9, 2012*, 2012. URL http://trec.nist.gov/pubs/trec21/papers/ICTNET.microblog.final.pdf.

[170] Xiang Zhu, Jiuming Huang, Sheng Zhu, Ming Chen, Chenlu Zhang, Zhenzhen Li, Huang Dongchuan, Zhao Chengliang, Aiping Li, and Yan Jia. NUDTSNA at TREC 2015 microblog track: A live retrieval system framework for social network based on semantic expansion and quality model. In *Proceedings of The Twenty-Fourth Text REtrieval Conference, TREC 2015, Gaithersburg, Maryland, USA, November 17-20, 2015*, 2015. URL http://trec.nist.gov/pubs/trec24/papers/NUDTSNA-MB.pdf.

[171] Arkaitz Zubiaga, Damiano Spina, Enrique Amigó, and Julio Gonzalo. Towards real-time summarization of scheduled events from twitter streams. In *Proceedings of the 23rd ACM Conference on Hypertext and Social Media*, HT '12, pages 319–320, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1335-3. doi: 10.1145/2309996.2310053. URL http://doi.acm.org/10.1145/2309996.2310053.

[172] Guido Zuccon, Bevan Koopman, Peter Bruza, and Leif Azzopardi. Integrating and evaluating neural word embeddings in information retrieval. In *Proceedings of the 20th Australasian Document Computing Symposium*, ADCS'15, pages 12:1–12:8,

New York, NY, USA, 2015. ACM. ISBN 978-1-4503-4040-3. doi: 10.1145/2838931.2838936. URL http://doi.acm.org/10.1145/2838931.2838936.