TEMPORAL MODELS

FOR MINING, RANKING AND RECOMMENDATION IN THE WEB

Von der Fakultät für Elektrotechnik und Informatik
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des Grades

DOKTOR DER NATURWISSENSCHAFTEN

Dr. rer. nat.

genehmigte Dissertation
von

M.Sc. Tu Nguyen

geboren am 28. September 1986, in Hanoi, Vietnam

2020

Referent: Prof. Dr. techn. Wolfgang Nejdl
Korreferent: Prof. Dr. Eelco Herder
Korreferent: Prof. Dr. Kurt Schneider
Tag der Promotion: 16. March 2020

ABSTRACT

Due to their first-hand, diverse and evolution-aware reflection of nearly all areas of life, heterogeneous temporal datasets i.e., the Web, collaborative knowledge bases and social networks have been emerged as gold-mines for content analytics of many sorts. In those collections, time plays an essential role in many crucial information retrieval and data mining tasks, such as from user intent understanding, document ranking to advanced recommendations. There are two semantically closed and important constituents when modeling along the time dimension, i.e., *entity* and *event*. Time is crucially served as the context for changes driven by happenings and phenomena (events) that related to people, organizations or places (so-called entities) in our social lives. Thus, determining what users expect, or in other words, resolving the uncertainty confounded by temporal changes is a compelling task to support consistent user satisfaction.

In this thesis, we address the aforementioned issues and propose temporal models that capture the temporal dynamics of such entities and events to serve for the end tasks. Specifically, we make the following contributions in this thesis:

- *Query recommendation* and *document ranking* in the Web - we address the issues for (1) suggesting entity-centric queries and (2) ranking effectiveness surrounding the *happening* time period of an associated event. In particular, we propose a multi-criteria optimization framework that facilitates the combination of multiple temporal models to smooth out the abrupt changes when transitioning between event phases in (1) and a probabilistic approach for search result diversification of temporally ambiguous queries for (2).
- *Entity relatedness* in Wikipedia - we study the long-term dynamics of Wikipedia as a *global memory place* for high-impact events, specifically the reviving memories of past events. Additionally, we propose a neural network-based approach to measure the temporal relatedness of entities and events. The model engages different latent representations of an entity (i.e., from time, link-based graph and content) and use the *collective attention* from user navigation as the supervision.
- *Graph-based ranking* and *temporal anchor-text mining* in Web Archives - we tackle the problem of discovering important documents along the time-span of Web Archives, leveraging the link graph. Specifically, we combine the problems of relevance, temporal authority, diversity and time in a unified framework. The model accounts for the incomplete link structure and natural time lagging in Web Archives in mining the temporal authority.
- Methods for *enhancing predictive models* at early-stage in social media and clinical domain - we investigate several methods to control model instability and enrich contexts of predictive models at the “cold-start” period. We demonstrate their effectiveness for the *rumor detection* and *blood glucose prediction* cases respectively.

Overall, the findings presented in this thesis demonstrate the importance of tracking these temporal dynamics surround salient events and entities for IR applications. We show that determining such changes in time-based patterns and trends in prevalent temporal collections can better satisfy user expectations, and boost ranking and recommendation effectiveness over time.

Keywords: *temporal dynamics, ranking, recommendation, events*

ZUSAMMENFASSUNG

Durch ihre eigene, vielfältige und evolutionäre Reflexion nahezu aller Lebensbereiche heterogen Zeitdatensätze, das Web, kollaborative Wissensdatenbanken und soziale Netzwerke haben Als Goldminen für die Inhaltsanalyse vielerlei Arten entstanden. In diesen Sammlungen spielt die Zeit weiter wesentliche Rolle bei vielen wichtigen Informationsabruf- und Data-Mining-Aufgaben, z.B. Verständnis der Benutzerabsicht, Dokument Ranking zu fortgeschrittenen Empfehlungen. Es gibt zwei semantisch geschlossen und wichtige Bestandteile bei der Modellierung entlang der Zeitdimension, d.H. *Entität* und *Ereignis*. Zeit ist entscheidend als Kontext für Veränderungen, der von Ereignissen und Phënomenen (Ereignissen) ausgelöst wurde Menschen, Organisationen, Orten usw. (Entitäten) in unserem sozialen Leben. So bestimmen, welche Benutzer erwarten, oder anders ausgedrückt, die durch zeitliche Änderungen verunsicherte Ungewissheit zu lösen, ist zwingend Aufgabe zur Unterstützung einer konstanten Benutzerzufriedenheit. In dieser Arbeit beziehen wir uns auf die oben genannten Probleme und schlagen Zeitmodelle vor, die einfangen die zeitliche Dynamik solcher Entitäten und Ereignisse erfassen, um für die endgültigen Aufgaben zu dienen. Konkret machen wir die folgenden Beiträge im Rahmen dieser Arbeit:

- Abfrageempfehlung und Dokumentenranking im Web - wir behandeln die Probleme für (1) Vorschläge für entitätszentrierte Abfragen und (2) Ranking-Effektivität rund um das Geschehen Zeitraum eines zugehörigen Ereignisses. Insbesondere schlagen wir eine Optimierung mit mehreren Kriterien vor Rahmen, der die Kombination mehrerer zeitlicher Modelle erleichtert, um abrupte Änderungen beim Übergang zwischen Ereignisphasen in (1) und einem probabilistischen Ansatz zur Suchergebnisdiversifizierung von zeitlich mehrdeutigen Abfragen für (2).
- Entitätenbezug in Wikipedia - Wir untersuchen die Langzeitdynamik von Wikipedia als global Speicherplatz für Ereignisse mit hoher Wirkung, insbesondere die Erinnerungen an vergangene Ereignisse. Zusätzlich, Wir schlagen einen neuronalen netzwerkbasierten Ansatz vor, um die zeitliche Beziehung zu messen von Entitäten und Ereignissen. Das Modell greift auf verschiedene latente Darstellungen einer Entität (d.H. von Zeit zu Zeit linkbasierte Grafiken und Inhalte) und nutzen Sie die kollektive Aufmerksamkeit der Benutzerführung als die Überwachung.
- Graphenbasiertes Ranking und zeitliches Ankertext-Mining in Web-Archiven - wir lösen das Problem Entdeckung wichtiger Dokumente über die Zeitspanne der Webarchive hinweg die Linkgrafik. Im Einzelnen kombinieren wir die Probleme Relevanz, zeitliche Autorität und Diversität und Zeit in einem einheitlichen Rahmen. Das Modell berücksichtigt die unvollständige Verbindungsstruktur und die natürliche Zeit Verzögerung in Web-Archiven beim Abbau der zeitlichen Autorität.
- Methoden zur Verbesserung von Vorhersagemodellen im frühen Stadium in sozialen Medien und im klinischen Bereich - Wir untersuchen verschiedene Methoden, um den Stabilitäts- und Anreicherungskontext des Modells zu steuern Vorhersagemodelle im 'Kaltstart'. Wir zeigen ihre Wirksamkeit für das Gerücht Erkennung und Vorhersage von Blutzuckerwerten.

Insgesamt zeigen die Ergebnisse dieser Arbeit, wie wichtig es ist, diese zeitlichen Prozesse zu verfolgen Dynamik umgibt Ereignisse und Entitäten für IR-Anwendungen. Wir zeigen das bestimmend Solche Änderungen in zeitlichen Mustern und Trends in vorherrschenden zeitlichen Sammlungen können besser befriedigen Erwartungen der Benutzer und Maximierung des Rankings.

Schlagwörter: *zeitliche Dynamik, Ranking, Empfehlung, Ereignisse*

ACKNOWLEDGMENTS

I would like to pay my special regards to Prof. Dr. techn. Dipl.-Ing. Wolfgang Nejdl for being the best professor I have ever known during my academic career.

I wish to show my gratitude to Asst. Prof. Dr. Nattiya Kanhabua, Asst. Prof. Dr. Eelco Herder and Dr. Claudia Niederée for their invaluable leadership support and supervision. I would also like to thank Mrs. Angelika Van Agen, Mrs. Michaela Kleiner and all L3S secretaries for their constant help and assistance.

I wish to acknowledge the friendship support and collaboration of the small but mighty Vietnamese community in L3S. I wish you all the best and hope the community continue to grow and gain great achievements in the future.

Thank you to all colleagues for being part of the family-like work environment at L3S. I have learned a lot from many of you.

Last but not least, I am indebted to my grand family, who nurtured my wildest dreams.

FOREWORD

The methods and algorithms presented in this thesis have been published at various conferences, as follows:

Chapter 3 addresses the problem of temporal dynamics of topics and entities that are tracked from query logs and describes the contributions included in:

- Tu Nguyen, Nattiya Kanhabua, Wolfgang Nejdl: *Multiple models for recommending temporal aspects of entities*. In Proceedings of the 15th Extended Semantic Web Conference, ESWC'18, pages 462–480, Springer, Cham. [NKN18a]
- Tu Nguyen, Nattiya Kanhabua: *Leveraging dynamic query subtopics for time-aware search result diversification*. In Proceedings of the 36th European Conference on IR Research, ECIR'14, pages 222-234, 2014, Springer. [NK14]

Chapter 4 focuses on events and their temporal dynamics in Wikipedia and builds upon the work published in:

- Nattiya Kanhabua, Tu Nguyen, Claudia Niederée. *What triggers human remembering of events? A large-scale analysis of catalysts for collective memory in Wikipedia*. In Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL'14, pages 341-350, 2014. ACM. [KNN14a]
- Tu Nguyen, Tuan Tran, Wolfgang Nejdl. *A Trio Neural Model for Ranking Dynamic Entities Relatedness.* In Proceedings of the 22nd SIGNLL Conference on Computational Natural Language Learning, CoNLL'18. ACL. [NTN18].

Chapter 5 addresses the problem of time-aware ranking in Web Archives and includes the contributions published in:

- Tu Nguyen, Nattiya Kanhabua, Claudia Niederée, Xiaofei Zhu: *A Time-aware Random Walk Model for Finding Important Documents in Web Archives*. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'15, pages 915-918, 2015. ACM. [NKNZ15]

- Tu Nguyen, Nattiya Kanhabua, Wolfgang Nejdl, Claudia Niederée: *Mining Relevant Time for Query Subtopics in Web Archives.* In Proceedings of the 24th International Conference on World Wide Web, WWW '15 Companion, pages 1357-1362, 2015. ACM. [NKNN15]

Chapter 6 addresses the problem of temporal dynamics in social networks and clinical domain and includes the contributions published in:

- Tu Nguyen, Cheng Li, Claudia Niederée: *On Early-Stage Debunking Rumors on Twitter: Leveraging the Wisdom of Weak Learners.* In Proceedings of the 9th International Conference on Social Informatics, Socinfo 2017, pages 141-158, 2017, Springer. [NLN17]
- Tu Nguyen, Markus Rokicki: *On the Predictability of non-CGM Diabetes Data for Personalized Recommendation.* In Proceedings of the ACM CIKM 2018 Workshops [NR18]

During the course of the Ph.D. studies I have also published a number of papers touching different aspects of content analytics, retrieval and social search. Not all aspects are touched in this thesis due to space limitation. The complete list of publications is as follows:

- Tu Nguyen, Tuan Tran, Wolfgang Nejdl. *A Trio Neural Model for Ranking Dynamic Entities Relatedness.* In Proceedings of the 22nd SIGNLL Conference on Computational Natural Language Learning, CoNLL'18. ACL. [NTN18]
- Tu Nguyen, Nattiya Kanhabua, Wolfgang Nejdl. *Multiple models for recommending temporal aspects of entities.* In Proceedings of the 15th Extended Semantic Web Conference, ESWC'18, pages 462–480, Springer, Cham. [NKN18a]
- Tu Nguyen, Cheng Li, Claudia Niederée. *On Early-Stage Debunking Rumors on Twitter: Leveraging the Wisdom of Weak Learners.* In Proceedings of the 9th International Conference on Social Informatics, Socinfo 2017, pages 141-158, 2017, Springer. [NLN17].
- Tu Nguyen. *A Comprehensive Low and High-level Feature Analysis for Early Rumor Detection on Twitter.* In CIKM 2017 Workshop on Interpretable Data Mining (IDM) - Bridging the Gap between Shallow and Deep Models. arXiv. [Ngu17].
- Tu Nguyen, Markus Rokicki: *On the Predictability of non-CGM Diabetes Data for Personalized Recommendation.* In Proceedings of the ACM CIKM 2018 Workshops [NR18].

- Khoi Duy Vo, Tuan Tran, Tu Nguyen, Xiaofei Zhu, Wolfgang Nejdl. *Can we find documents in web archives without knowing their contents?*. In Proceedings of the 8th ACM Conference on Web Science, Websci'16, pages 173-182, 2016. ACM. [VTN$^+$16].

- Nattiya Kanhabua, Philipp Kemkes, Wolfgang Nejdl, Tu Nguyen, Felipe Reis, Nam Khanh Tran. *How to Search the Internet Archive Without Indexing It.* In Proceedings of the 20th International Conference on Theory and Practice of Digital Libraries, TPDL'16, pages 147-160, 2016, Springer. [KKN$^+$16]

- Nattiya Kanhabua, Tu Nguyen, Wolfgang Nejdl. *Learning to detect event-related queries for web search.* In Proceedings of the 24th International Conference on World Wide Web, WWW'15 Companion, pages 1339-1344, 2015, ACM. [KNNN15].

- Tu Nguyen, Nattiya Kanhabua, Claudia Niederée, Xiaofei Zhu: *A Time-aware Random Walk Model for Finding Important Documents in Web Archives*. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'15, pages 915-918, 2015. ACM. [NKNZ15]

- Tu Nguyen, Nattiya Kanhabua, Wolfgang Nejdl, Claudia Niederée: *Mining Relevant Time for Query Subtopics in Web Archives.* In Proceedings of the 24th International Conference on World Wide Web, WWW '15 Companion, pages 1357-1362, 2015. ACM. [NKNN15]

- Tuan Tran, Tu Nguyen. *Hedera: scalable indexing and exploring entities in wikipedia revision history.* In Proceedings of tthe 13th International Semantic Web Conference, Posters & Demonstrations Track, ISWC'14, pages 279-300, 2014, CEUR-WS. org. [TN14].

- Tu Nguyen, Nattiya Kanhabua: *Leveraging dynamic query subtopics for time-aware search result diversification*. In Proceedings of the 36th European Conference on IR Research, ECIR'14, pages 222-234, 2014, Springer. [NK14]

- Michele Fenzi, Jörn Ostermann, Tu Nguyen, Thomas Risse, Nico Mentzer, Guillermo Payá-Vayá, Holger Blume. *ASEV–Automatic situation assessment for event-driven video analysis.* In Proceedings of the 11th IEEE International Conference on Advanced Video and Signal Based Surveillance,AVSS'14, pages 37-43, 2014, IEEE. [FOM$^+$14].

- Giang Tran, Mohammad Alrifai, Tu Nguyen, Wolfgang Nejdl. *Wikitimess knowledge extraction and enrichment process.*, L3S Technical Report, 2015. [TANN].

- Tu Nguyen, Wolf Siberski. *SLUBM: An Extended LUBM Benchmark for Stream Reasoning.* In Proceedings of the 2nd International Conference on Ordering and Reasoning, co-allocated with ISWC'13, pages 43-54, 2013, CEUR-WS. org. [NS13].

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Ever since the dawn of the World Wide Web in 1989, technology has been rapidly and constantly developed and facilitating usage of the Internet towards billions of people in the World. As of June 2018, 55.1% of the world's population has Internet access [1] and those are also the (possible) content creators of the Web. Nowadays, Web users have the means to consume and create content over various sharing platforms like News, Social Media, public encyclopedia and other platforms. The digital content generated over this long-serving period has become the richest, largest source of data as well as brought in many challenges for analytics. National libraries and organizations like the Internet Archive (archive.org) or the Portuguese Web Archive (arquivo.pt) have been capturing Web contents for decades thenceforward. These archives host a wealth of information, providing a gold mine for sociological, political, business, and media analysts. For instance, one could track and analyze public statements made by representatives of the White House, characterizing the evolution of patterns in their attitude towards controversial topics (e.g., *abortion* or *climate change*). Another typical use case is tracking over a long time horizon, how long-running events (e.g., the Syrian Civil War) develop, such as what are the major time points and key people involved, to capture the dynamic event unfolding. Or, being interested in only the present time, how are the patterns of the past can help capture the user intent when she is searching for an incoming event (e.g., when looking for the participants of the US Open 2019). Analyses of this kind could also be carried out on large news archives (e.g., The New York Times Archives), but this can be seen as variant of Web archive analytics; on top of that, the Web is much more heterogeneous, that provides a much wider variety of perspectives, thus is a richer source for interesting temporal patterns and trends.

Analysts (e.g., librarians, journalist) would not only be interested in text or Web pages intrinsically, even if the underlying sources are in text or multimodal form. Instead, they also want to see, compare, and understand the behavior of (and trends about) entities like

[1] https://en.wikipedia.org/wiki/Global_Internet_usage

companies, products, politicians, music bands, songs, movies, etc., thus calling for entity-level analytics over Web archives [WNS+]. Although such information from the past might still be findable in the current web, they are typically aggregated, filtered and interpreted from a current perspective. For experts and professionals such as journalists, researchers from political science and sociology, historians, etc. a first-hand and unbiased reflection of the world opens up the investigation of own stories and completely new re-search questions. It enables them to better understand how and why issues such as, for example, controversial topics evolved over time. They can also see the context of such discussions and have a first hand account of change such as the evolution of language. However, so far, since the time this thesis was started, the focus for Web archives was mainly on capturing and - in the area of use - on targeted navigational access via URLs. Only until recently, there appears work on more advanced forms of Web Archive search, that support analytic functionalities, from the text level to the entity level: e.g., detecting named entities, resolving ambiguous names, tracking the same entity in its mentions over extended time periods. Of which, they remain a grand challenge, regarding both semantics and scalability, for large-scale longitudinal analytics [WNS+, SW12].

1.2 Thesis Scope

We address the issues of mining the temporal dynamics of basic elements in different vast temporal collections, i.e., the Web (and Web Archives), Wikipedia and social networks (e.g., Twitter). In advance of that, we propose methods for ranking and recommendation applications in the collection's contexts. Before delving into the details of the problems, we clarify some notions and terminologies that will use throughout this thesis. We will use *entities* and *events* with which we basically refer to the corresponding Wikipedia entity and event pages. The *topics* or *aspects* of an entity will be precisely defined in later chapters, but in the most essential respects, it can be implied as a piece of text that expresses a distinct 'facet' of the entity.

(I) The intents behind ambiguous and multi-faceted entity-centric queries that users make every now and then are often hard to determine. In some special cases, aspects associated to such queries can even be temporally ambiguous, for instance, the query US Open is more likely to be targeting the tennis open in September, and the golf tournament in June. For this problem, we lay on two different research questions:

> ***RQ1.1*** How do the relevant aspects of an entity-centric query change around the associated event time, specifically *just* before, after and during the event time.

The outcome from **RQ1.1** is of great importance in understanding user intents at different phases of the event time. Thus, it is very important for many recommendation tasks for search engines, e.g., query suggestion.

RQ1.2 Given an entity-centric query of semantical or topical ambiguity at an *event time*, how should the ranked list of relevant documents be formed so that the coverage at top-k is maximized?

Returning top-k relevant documents for an ambiguous query is often solved by diversifying the results based on the topic dimension in literature. However, during the event time, the timeliness of the topic / aspects should be taken into account.

(II) Going beyond its role as an encyclopedia, Wikipedia becomes a global memory place for high-impact events, such as, natural disasters and manmade incidents, thus influencing collective memory, i.e., the way we remember the past. Due to the importance of collective memory for framing the assessment of new situations, our actions and value systems, its open construction and negotiation in Wikipedia is an important new cultural and societal phenomenon. We represent two research questions that leverage the patterns in collective memory captured in Wikipedia as follows:

RQ2.1 How past events are remembered and what triggers human remembering of these events in Wikipedia?

What signals that ignite remembering a past event is an unclear question that often depends on many factors and varies along different event types. The analysis of this phenomenon is an important foundation for technology, which more effectively complements the processes of human forgetting and remembering and better enables us to learn from the past.

RQ2.2 How do we quantify the semantic relatedness between two entities / events?

Determining the relative relatedness among entities accurately has many important implications.

(III) Web search is good in delivering (more or less) up-to-date or fresh information for topics of all types. Due to its vivid and wide use and participative content creation, the web is in addition a good reflection of processes, practices, and topics in all areas of life includ- ing politics, society, science etc. When we regularly take snapshots of the web at different times as it is done in web Archiving (at least for part of the web), we can, thus, capture this world reflection at different times as well as its evolution via subsequent versions. Thus, web archives have the potential to provide a rich source of first-hand information from the past - and about how things evolved. It can, for example, be seen how topics such as integration, nuclear power or democracy where discussed in the early 90's - and how this discussion changed over time. Or looking at more mundane issues, in 30 years from now we can see what people did wear, eat, and talk about in 2014 from archived evidences. Although such content might seem trivial in the first place, it accumulates into an unpreceeded form grass-root historical records. However, searching in this unique longitudinal collection of huge redundancy (pages of near-identical content are crawled all over again) is completely different from searching over the web. Those tasks especially differ in the dominating user intents and in the core role of time in the structure of web archives. In this context, content relevance is not the main and only driver but also time relevance and impact are the other key factors. A search primitive should represent a good result coverage to support exploration and discovery.

> ***RQ3*** Given a query and the web archive, how do we come up with a top-k ranked list of documents where the coverage of the most important topic-wise and time-wise documents are maximized.

Supporting search, which goes beyond navigational search via URLs, is a very challenging task in these unique structures with huge, redundant and noisy temporal content of Web Archives. The search needs of expert users such as journalists, economists or historians for discovering a topic-in-time are not only about getting the most important documents but they (the search results) also should cover the most interesting time-periods for the topic.

(IV) We present a general research question regarding stablizing and enhancing model performance at early stage, in the context of social media (task: rumor detection) and clinical domain (task: blood glucose prediction).

> ***RQ4*** How do temporal models develop and how do we control and improve the stability of such models at early-stage?

Temporal models always need to make a reliable prediction / recommendation as early as possible. However, they often suffer from the instability as learning from the noisy and scarce data during this “cold-start” phase.

1.3 Contributions of the Thesis

In this thesis, we answer the research questions formalized in the previous section. The contribution of this thesis is on providing effective methods for mining, ranking and recommedation in the Web.

Recommendation and Ranking for Web Search: *Query recommendation* and *document raking* in the Web - we address the issues for (1) suggesting entity-centric queries and (2) ranking effectiveness surrounding the happening time period of an associated event. In particular, for (1), we study the task of temporal aspect recommendation for a given entity, which aims at recommending the most relevant aspects and takes into account time in order to improve search experience. For such cases, aspect suggestion based solely on salience features can give unsatisfactory results, for two reasons. First, salience is often accumulated over a long time period and does not account for recency. Second, many aspects related to an event entity are strongly time-dependent. To this end, we propose a novel event-centric ensemble ranking method that learns from multiple time and type-dependent models and dynamically trades off salience and recency characteristics. For (2), we propose a probabilistic approach for search result diversification of temporally ambiguous queries. The key idea is to re-rank search results based on the freshness and popularity of temporal aspects.

Entity relatedness in Wikipedia: We study the long-term dynamics of Wikipedia as a global memory place for high-impact events, specifically the reviving memories of past events. Additionally, we propose a neural network-based model to measure the temporal relatedness of entities and events. The model engages different latent representations of an entities (i.e., from time, link-based graph and content) and use the *collective attention* from user navigation as the supervision. Our proposed model thus is capable of incorporating multiple views of the entities, both from content provider and from user's perspectives. We also introduce an attention-based convolutional neural networks (CNN) to capture (i.e., learn the representation) the temporal signals of an entity.

Ranking in Web Archives: *Graph-based ranking* and *temporal anchor-text mining* in Web Archives - we tackle the problem of discovering relevant documents of importance along the time-span of the web archives in a ranking approach. The intuition is that the impact/authority of a document in the web archives with regards to a query is strongly time-influenced. Based on this idea, we propose a novel random walk-based ranking algorithm that integrates relevance, temporal authority, diversity and time in a unified framework. The model is based on the vertex-reinforced random walks that encourages diversity in a 'winner-takes-all' manner (such as absorbing authority from its neighborhood).

Methods for *enhacing predictive models*: at early-stage in social media and clinical domain - we investigate several methods to control the model instability and enriching context of predictive models at the "cold-start" period. We demonstrate their effectiveness for the *rumor detection* (1) and *blood glucose prediction* (2) cases. In (1), we propose an approach that leverages stacked $CNN + LSTM$ for learning the latent representations of individual rumor-related tweets to gain contexts on the credibility of each tweets. Our extensive experiments show that our model, with the credible 'wisdom' of aggregated tweets, clearly improves the classification performance within the critical very first hours of a rumor. In (2), we study the bootstrapping techniques for variance reduction in *Bagging* and applying them for filtering data of uncertainty at inference time during the 'cold-start' stage. Specifically, we leverage methods for estimating the sampling variance of bagged predictors in Random Forest; the variance is an indicator for the uncertainty of the learned ensemble model. To this end, our contributions are two-fold: first, we provide a quantitative study on the predictability of machine learned models on limited and sparse clinical data; second, we propose a prediction system that is robust on noisy data (based on several filtering methods).

1.4 Thesis Structure

The remainder of the thesis is organized as follows:

In chapter 2, we discuss selected general background techniques and algorithms that build a basis to achieve the goals of this thesis. In particular, we focus on selected techniques from the areas of Machine Learning, Information Retrieval, and Natural Language Processing.

Chapter 3 addresses the problem of temporal dynamics of topics and entities that are tracked from query logs. We provide two applications that directly benefit from such temporal dynamics modeling, i.e., (1) query suggestions and (2) search result diversification.

Chapter 4 focuses on events and their temporal dynamics in Wikipedia. We first mine the semantic relatedness of highly impact events in history and how they are related / get recalled throughout time. Then we provide a deep learning method for quantifying such relatedness via learning the entity representations from different embedding techniques.

Chapter 5 addresses the problem of time-aware ranking in Web archives. We focus on the temporal evolutions of subtopics (particularly event aspects) in the unique structure of these collections and leverage them to enhance the Markov chain reinforcement policy of the random walk-based ranking solution for the corresponding queries.

Chapter 6 addresses the problem of temporal dynamics in social networks and clinical domain. We further use different ensemble techniques in order to control the model's prediction instability at early stage.

2 General Background

In this chapter, we present core machine learning theoretical backgrounds that are the backbone of the thesis. We first focus on supervised-based learning with traditional algorithms and neural network-based approaches. We then introduce ranking models and conclude with convex optimization techniques.

2.1 Supervised Learning Models

Supervised learning can roughly be understood as the affair of teaching a model by feeding it input data as well as correct labels for the (not necessarily) complete input. For the case that only a small amount of input are labeled, and the model needs to make use of the large amount of unlabeled data to generalize the population structure, the technique is known as semi-supervised learning. We only discuss the general supervised approach in this Chapter.

2.1.1 Support Vector Machines

Support Vector Machines, (SVM) is a discriminant supervised learning model that can be generally understood as the task of constructing a hyperplane which segregates input instances linearly. An optimal hyperplane is constructed based on the so-called support vectors, which determines the maximal margin between support vectors of different classes. We demonstrate the basic mechanism of Linear-SVM in Figure 2.1.

In more details, the hyperplane for linear SVM is a set of points X such that, $w^T\phi(X)+b=0$, where $\phi(X)$ denotes the multi-dimensional vector representation of instance X, whereas w is the normal vector (that is perpendicular) to the hyperplane. To separate the two classes of dataset linearly, we can identify two hyperplanes such that $w^T\phi(X)+b=1$ and $w^T\phi(X)+b=-1$, so that the distance (which is $\frac{2}{\|w\|}$) between the two classes (with labels $\{-1,1\}$) is maximized. For non-linear model, the objective function is changed to the *hinge loss* that maximize $(0,1-y_i(w\cdot\phi(x_i)-b))$. To make SVM work for non-linear

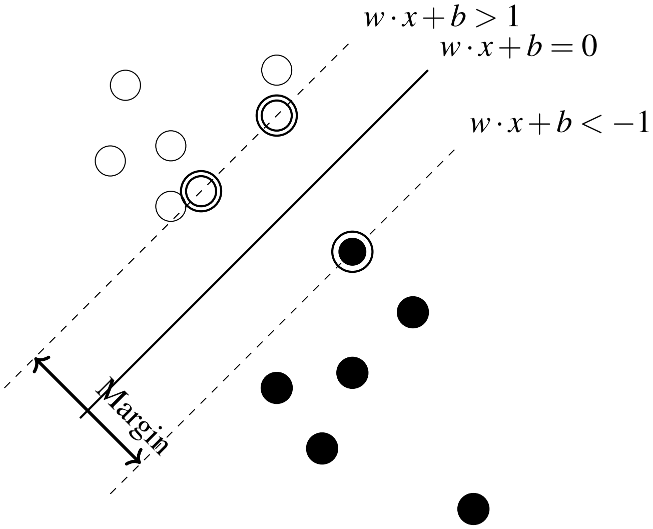


Figure 2.1: The graphical illustration of SVM.

binary classification, the data is often transformed into higher dimensional space. In order to avoid this computationally expensive effort, kernel methods are introduced. Specifically, with what is popularly known as the *kernel trick*, we can implicitly map the input into higher dimensional feature spaces by computing only the inner products between input data vectors. When the number of features is large (proportionally to the number of instances), a linear kernel is preferred (computation-wise), otherwise nonlinear kernels (e.g., Gaussian kernel) are chosen.

2.1.2 Ensemble Models

We present two popular ensemble approaches in this section: *bagging* and *boosting*. Bagging is a popular technique for stabilizing statistical learners. Bagging is often conceptualized as a variance reduction technique, and so it is important to understand how the sampling variance of a bagged learner compares to the variance of the original learner. A most popular model for *bagging* is the Random Forest. In Random Forests (RF), the input space from a set of training instances is split into K classification trees resulting into a forest. The generation of trees follows two main principles. First, the input feature space is split into K random vectors, or feature subsets chosen randomly, resulting in classification trees hK. In the second approach, in the case of low-dimensional input feature space, one can employ a linear combination of features and consequentially generate classification trees based on the CART algorithm. On the other hand, the idea of *boosting* is to train weak learners sequentially, each trying to correct its predecessor. There are two different ways of correction: (1) change the weights for every incorrect classified observation at every iteration (i.e., AdaBoost) and (2) try to fit the new predictor to the residual errors made by the previous predictor (i.e., Gradient Boosting).

2.1.3 Convolutional Neural Network

The CNN network is widely used in Computer Vision and NLP for modeling the representation of images and text. In this thesis, we describe the network in details with regards to its application to text modeling.

Let x_i in $\mathbb{R}^d$ be the $d-$dimensional word vector corresponding to the $i-th$ word in the sentence. A sentence of length n (padded where necessary) is represented as:

$$x_{1:n} = x_1 \oplus x_2 \oplus \cdot \oplus x_n \tag{2.1}$$

where $\oplus$ is the concatenation operator. In general, let $x_{i:i+j}$ refer to the concatenation of words $x_i, x_{i+1}, \cdots, x_{i+j}$. A convolution operation involves a filter $w \in \mathbb{R}_{hd}$, which is applied to a window of h words to produce a new feature. For example, a feature c_i is generated from a window of words $x_{i:i+h1}$ by:

$$c_i = f(w\dot{x}_{i:i+h-1} + b) \tag{2.2}$$

with b in $\mathbb{R}$ is a bias and f is a non-linear function such as the hyperbolic tangent. This filter is applied to each possible window of words in the sentence $x_{1:h}, x_{2:h+1}, \cdots, x_{nh+1:n}$ to produce a **feature map**.

$$c = [c_1, c_2, \cdots, c_{n-h+1}] \tag{2.3}$$

with c in $\mathbb{R}^{n-h+1}$. A max-over-time pooling operation is often applied over the feature map and take the maximum value as the feature corresponding to this particular filter. The idea is to capture the most important feature -one with the highest value- for each feature map. This pooling scheme naturally deals with variable sentence lengths.

2.1.4 Long short term memory

Recurrent neural networks (RNNs, see Figure 2.2) are able to process input sequences of arbitrary length via the recursive application of a transition function on a hidden state vector h_t . At each time step t, the hidden state h_t is a function of the input vector x_t that the network receives at time t and its previous hidden state h_{t1}. Generally, the RNN transition function is an affine transformation (linear mapping between affine spaces) followed by a point-wise nonlinearity function (e.g., more commonly, the hyperbolic tangent):

$$h_t = tanh(\mathbf{W}x_t + Uh_{t-1} + b) \tag{2.4}$$

However, transition functions of this form suffers from the problem that, during training components of the gradient vector can grow or decay exponentially over long sequences [BSF94]. The LSTM architecture (see Figure 2.3) addresses directly this problem by introducing a *memory cell* that is able to preserve state over long periods of time. We describe its architecture in details as follows:

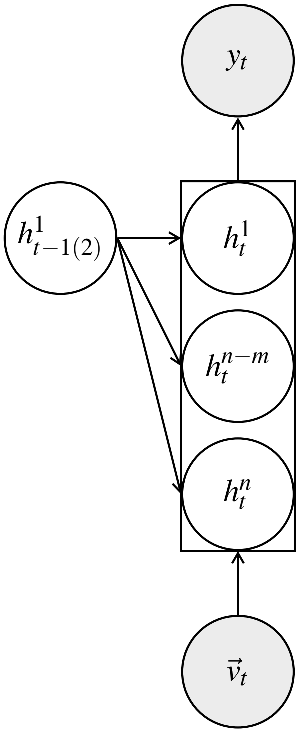


Figure 2.2: The graphical illustration of RNN network.

We define the LSTM unit at each time step t to be a collection of vectors in $\mathbf{R}^d$: an input gate i_t, a forget gate f_t, an output gate o_t, a memory cell c_t and a hidden state h_t. The entries of the gating vectors i_t, f_t and o_t are in $[0,1]$. We refer to d as the memory dimension of the LSTM. The LSTM transition equations are the following:

$$\begin{aligned}
i_t &= \sigma\big(\mathbf{W}^{(i)}x_t + U^{(i)}h_{t-1} + b_{(i)}\big),\\
f_t &= \sigma\big(\mathbf{W}^{(f)}x_t + U^{(f)}h_{t-1} + b_{(f)}\big),\\
o_t &= \sigma\big(\mathbf{W}^{(o)}x_t + U^{(o)}h_{t-1} + b_{(o)}\big),\\
u_t &= tanh\big(\mathbf{W}^{(u)}x_t + U^{(u)}h_{t-1} + b_{(u)}\big),\\
c_t &= i_t \odot u_t + f_t \odot c_{t-1},\\
h_t &= o_t \odot tanh(c_t)
\end{aligned}$$

where x_t is the input at the current time step, σ denotes the logistic sigmoid function and $\odot$ denotes elementwise multiplication.

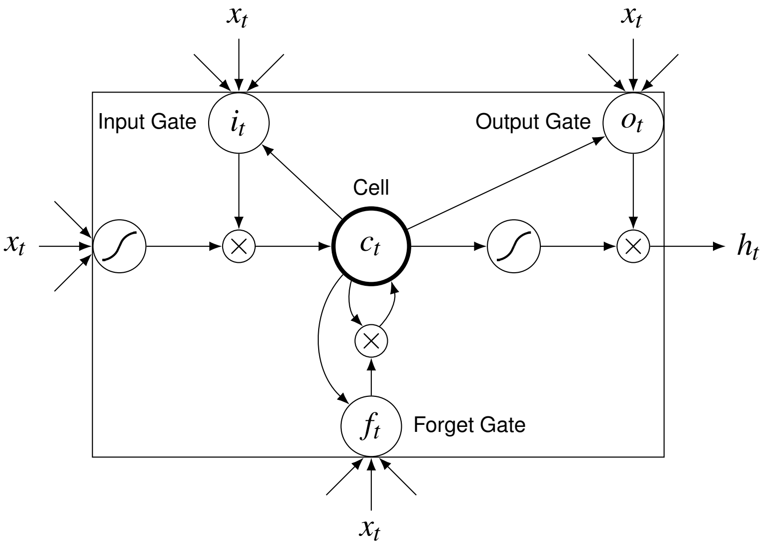


Figure 2.3: The graphical illustration of LSTM network.

2.2 Learning to Rank

Learning to rank for Information Retrieval is a problem formalized as described next. In learning (training), a collection of queries and their corresponding retrieved documents are given. Furthermore, the annotations (i.e., relevance judgements) of the document with respect to the queries are also provided. The relevance judgements, provided by human annotators, can represent ranks (e.g., categories in a total order). The objective of learning is to construct a ranking model, e.g., a ranking function, that achieves the best result on test data in the sense of optimization of a performance measure (e.g., error rate, classification accuracy, Mean Average Precision, etc.) In retrieval (test phase), given a query, the learned ranking function is applied, returning a ranked list of documents in descending order of their relevance scores.

L2R can be generally categoried into 3 different approaches, based on the input representation: (1) *pointwise*, (2) *pairwise* and (3) *listwise*. In pointwise approach, the goal is to approximate document-query scores using ordinal regression or classification algorithms. Whereas, in pairwise approach, the absolute approximation errors are not important, the goal is instead to minimize the mis-ranks, i.e. given two documents of the same query, documents of less relevance should not be scored higher. crowdsourced results. In listwise approach, the L2R system tries to optimize directly the list in which documents are scored and ordered, based on the lists presented in training queries. Because variables of the optimization functions are sets instead of individual documents, listwise L2R are more difficult to model, and different assumptions must be introduced to simplify the process, such as in Plackett-Luce model.

2.2.1 RankSVM

Ranking SVM is a variant of the support vector machine algorithm, which is used to solve certain ranking problems (via learning to rank).

The Ranking SVM algorithm is a learning retrieval function that employs pair-wise ranking methods to adaptively sort results based on how ‘relevant’ they are for a specific query. The Ranking SVM function uses a mapping function to describe the match between a search query and the features of each of the possible results. This mapping function projects each data pair (such as a search query and clicked web-page, for example) onto a feature space. These features are combined with the corresponding click-through data (which can act as a proxy for how relevant a page is for a specific query) and can then be used as the training data for the Ranking SVM algorithm.

Generally, Ranking SVM includes three steps in the training period:

(1) It maps the similarities between queries and the clicked pages onto a certain feature space. (2) It calculates the distances between any two of the vectors obtained in step 1. (3) It forms an optimization problem which is similar to a standard SVM classification and solves this problem with the regular SVM solver.

2.2.2 Neural Ranking

Neural ranking models for information retrieval (IR) use shallow or deep neural networks to rank search results in response to a query. Traditional learning to rank models employ machine learning techniques over hand-crafted IR features. By contrast, neural models learn representations of language from raw text that can bridge the gap between query and document vocabulary [MC17]. One way of learning is a joint representation of the query and the document is generated using manually designed features and the neural network is used only at the point of match to estimate relevance. Another way is depending on learning good low-dimensional vector representations-or embeddings-of query and document text, and using them within traditional IR models or in conjunction with simple similarity metrics (e.g., cosine similarity). The latent (dense) representation of such query and documents in this way if often obtained by CNN or LSTM (in case of short-text documents).

2.3 Optimization Methods in Machine Learning

Optimization is the most essential factor / process for the success of machine learning algorithms. Optimization can be defined by a loss function/cost function and the process minimizing it using one or the other optimization routine. A good choice of optimization algorithm can make a huge difference between getting a good accuracy in hours or days.

2.3.1 Unconstrained Optimization

Gradient-based methods. Batch methods (e.g., limited memory BFGS) which require the full training set for computing the next parameter update at each iteration, tend to converge very well to local optima. However, in practice, computing the cost and gradient for the entire training set can be very slow and sometimes intractable for limited memory resource. Another issue with batch methods is that they are not flexible for new data in an 'online' setting. The stochastic Gradient Descent (SGD) provides the solution to both of these issues by following the negative gradient of the objective after seeing only a single or a few training examples. The use of SGD in the neural network setting is motivated by the high cost of running back propagation over the full training set. SGD can overcome this cost and still lead to fast convergence.

Second-order methods. Second-order methods use the second order derivative (also known as the Hessian) to minimize or maximize the Loss function. The second order derivative provide us a fully detailed information whether the first derivative is increasing or decreasing which hints at the function's curvature. It provides us with a quadratic surface which touches the curvature of the error surface. Although the second order derivative is very costly in practice, its advantage is that it does not neglect or ignore the curvature of Surface like first-order approaches (see Figure 2.4). Secondly, in terms of Step-wise Performance they are better.

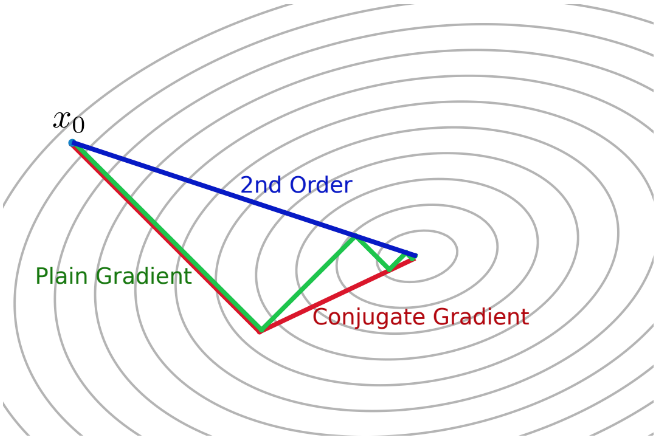


Figure 2.4: First and second-order optimization [Int].

2.3.2 Constrained Optimization

Different from uncontrained optimization, constrained optimization has no convexity assumption and is often augmented with equality constraints. The general approach for dealing with such problem is transforming the constraint problem to either: (1) a series of unconstraint problems, (2) a single but larger unconstraint problem or (3) another constraint problem, hopefully simpler (dual, convex). We describe (3) in more details as it is adopted in this thesis, specifically in the optimization of SVM.

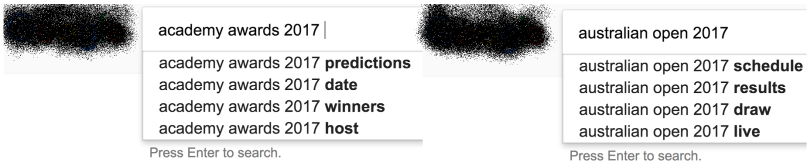
3 Temporal Dynamics for Web Search

3.1 Temporal Dynamics of Entities Aspects - The recommendation task

Beyond the traditional "ten blue links", to enhance user experience with entity-aware intents, search engines have started including more semantic information, (1) suggesting related entities [BCMT13, FBMB15, YMHH14, ZRZ16], or (2) supporting entity-oriented query completion or complex search with additional information or *aspects* [BDDI, RMdR15, SMDR+]. These aspects cover a wide range of issues and include (but are not limited to) types, attributes/properties, relationships or other entities in general. They can change over time, as *public attention* shifts from some aspects to others. In order to better recommend such entity aspects, this temporal dimension has to be taken into account.

Exploiting collaborative knowledge bases such as Wikipedia and Freebase is common practice in semantic search, by exploiting anchor texts and inter-entity links, category structure, internal link structure or entity types [BCMT13]. More recently, researchers have also started to integrate knowledge bases with query logs for *temporal* entity knowledge mining [CLK+16, YMHH14]. In this section, we address *the temporal dynamics of recommending entity aspects* and also utilize query logs, for two reasons. First, query logs are strongly entity related: more than 70% of Web search queries contain entity information [LPG+, PMZ]. Queries often also contain a short and very specific piece of text that represents users' intents, making it an ideal source for mining entity aspects. Second, different from knowledge-bases, query logs naturally capture temporal dynamics around entities. The intent of entity-centric queries is often triggered by a current event [KTSD, KLL+], or is related to "what is happening right now".

Previous work do not address the problem of temporal aspect recommendation for entities, often event-driven. The task requires taking into account the impact of temporal aspect dynamics and explicitly considering the relevance of an aspect with respect to the time period of a related event. To demonstrate the characteristics of these entity aspects, we showcase a real search scenario, where entity aspects are suggested in the form of query



[redacted]

academy awards 2017

academy awards 2017 **predictions**
academy awards 2017 **date**
academy awards 2017 **winners**
academy awards 2017 **host**

Press Enter to search.

[redacted]

australian open 2017

australian open 2017 **schedule**
australian open 2017 **results**
australian open 2017 **draw**
australian open 2017 **live**

Press Enter to search.

Figure 3.1: [Screenshot] Recommendation generated by a commercial search engine for academy awards 2017 and australia open 2017, submitted on March 31th, 2017, on a clean history browser.

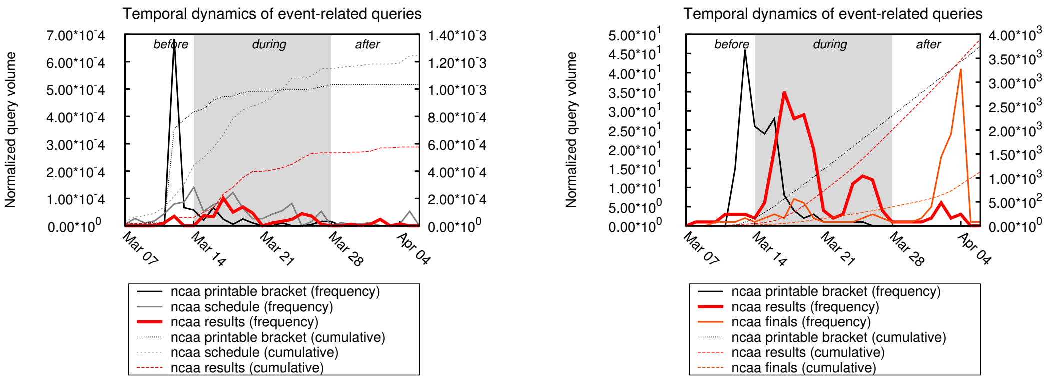
suggestion / auto-completion, given the entity name as a prior. Figure 3.1 shows the lists of aspect suggestions generated by a well-known commercial search engine for academy awards 2017 and australia open 2017. These suggestions indicate that the top-ranked aspects are mostly time-sensitive, and as the two events had just ended, the recommended aspects are timeliness-wise irrelevant (e.g., *live*, *predictions*).

Although the exact techniques behind the search engine's recommendation are unknown, the mediocre performance might be caused by the effect of aspect salience (query popularity in this case) and the *rich get richer* phenomenon: the salience of an aspect is accumulated over a long time period. Figure 4.12 illustrates changes in popularity of relevant searches captured in the AOL (left) and Google (right) query logs (e.g., ncaa printable bracket, ncaa schedule, and ncaa finals) for the NCAA[1] tournament. The basketball event began on March 14, 2006, and concluded on April 3, 2006. In order to better understand this issue, we present two types of popularity changes, namely, (1) frequency or query volume (aggregated daily), and cumulative frequency. Frequencies of pre-event activities like printable bracket and schedule gain increased volume over time, especially in the *before* event period. On the other hand, up-to-date information about the event, such as, ncaa results rises in importance when the event has started (on March 14), with very low query volume before the event. While the popularity of results or finals aspect exceeds that of ncaa printable bracket significantly in the periods *during* and *after* event, the cumulative frequency of the pre-event aspect stays high. We witness similar phenomenon with the same event in 2017 in the Google query logs. We therefore postulate that (1) long-term salience should provide good ranking results for the periods *before* and *during*, whereas (2) short-term or recent interest should be favored on triggers or when the temporal characteristics of an event entity change, e.g., from *before/during* to *after* phase. Different event types (breaking or anticipated events) may vary significantly in term of the impact of events, which entails different treatments with respect to a ranking model.

Our contributions can be summarized as follows.

- We present the first study of temporal entity aspect recommendation that explicitly models triggered event time and type.
- We propose a learning method to identify time period and event type using a set of

[1] A major sports competition in the US held annually by the National Collegiate Athletic Association (NCAA)- https://en.wikipedia.org/wiki/Ncaa.



(a) AOL

(b) Google

Figure 3.2: Dynamic aspect behaviors for entity ncaa in AOL and Google.

features that capture temporal dynamics related to event diffusion.

- We propose a novel event-centric ensemble ranking method that relies on multiple time and type-specific models for different event entities.

To this end, we evaluated our proposed approach through experiments using real-world web search logs – in conjunction with Wikipedia as background-knowledge repository.

3.1.1 Related Work

Entity aspect identification has been studied in [SMDR$^+$, RMdR15]. [SMDR$^+$] focuses on salient ranking features in microblogs. Reinanda et al. [RMdR15] start from the task of mining entity aspects in the query logs, then propose salience-favor methods for ranking and recommending these aspects. When regarding an aspect as an entity, related work connected to temporal IR is [ZRZ16], where they study the task of time-aware entity recommendation using a probabilistic approach. The method also *implicitly* considers event times as triggering sources of temporal dynamics, yet relies on coarse-grained (monthly) granularity and does not recognize different phases of the event. It is therefore not really suitable for recommending fine-grained, temporal aspects. 'Static' entity recommendation was first introduced by the Spark [BCMT13] system developed at Yahoo!. They extract several features from a variety of data sources and use a machine learning model to recommend entities to a Web search query. Following Spark, Sundog [FBMB15] aims to improve entity recommendation, in particular with respect to freshness, by exploiting Web search log data. The system uses a stream processing based implementation. In addition, Yu et al. [YMHH14] leverage user click logs and entity pane logs for global and personalized entity recommendation. These methods are tailored to ranking entities, and face the same problems as [ZRZ16] when trying to generalize to 'aspects'.

It is also possible to relate these entity aspects to RDF properties / relations in knowledge bases such as FreeBase or Yago. [VTS16, DA16] propose solutions for ranking these properties based on salience. Hasibi et al. [HBB] introduce dynamic fact based ranking (property-object pairs towards a sourced entity), also based on *importance* and *relevance*. These properties from traditional Knowledge Bases are often too specific (fact-centric) and temporally static.

3.1.2 Background and Problem Statement

Preliminaries

In this work, we leverage clues from entity-bearing queries. Hence, we first revisit the well-established notions of query logs and query-flow graphs. Then, we introduce necessary terminologies and concepts for entities and aspects. We will employ user log data in the form of queries and clicks.

Our datasets consist of a set of queries Q, a set of URLs U and click-through information S. Each query $q \in Q$ contains query terms $term(q)$, timestamps of queries $time(q)$ (so-called *hitting time*), and an anonymized ID of the user submitted the query. A clicked URL $u \in U_q$ refers to a Web document returned as an answer for a given query q. Click-through information is a transactional record per query for each URL clicked, i.e., an associated query q, a clicked URL u, the position on result page, and its timestamps. A co-clicked query-URL graph is a bipartite graph $G = (V, E)$ with two types of nodes: query nodes V_Q and URL nodes V_U, such that $V = V_Q \cup V_U$ and $E \subseteq V_Q \times V_U$.

Problem Definitions

We will approach the task of recommending temporal entity aspect as a ranking task. We first define the notions of an *entity query*, a *temporal entity aspect*, developed from the definition of entity aspect in [RMdR15], and an *event entity* . We then formulate the task of recommending temporal entity aspects.

Definition 1. *An entity query q_e is a query that is represented by one Wikipedia entity e.* We consider q_e as the representation of e.

Definition 2. *Given a "search task" defined as an atomic information need, a temporal "entity aspect" is an entity-oriented search task with time-aware intent.*. An entity-oriented search task is a set of queries that represent a common task in the context of an entity, grouped together if they have the same intent [RMdR15]. We will use the notion of query q to indicate an entity aspect a interchangeably hereafter.

Definition 3. *An entity that is related to a near event at time t_i is called an event-related entity, or event entity for short.* Relatedness is indicated by the observation that *public attention* of temporal entity aspects is triggered by the event. We can generalize the term *event entity* to represent any entity that is related to or influenced by the event. An event entity e that is associated to the event whose type $\mathscr{C}$ can be either *breaking* or *anticipated*. An event entity is also represented as a query with hitting time t. The association between t and the event time –defines e's time period $\mathscr{T}$– that can be either of the *before*, *during* or *after* phases of the event. When the entity is no longer event-related, it is considered a "static" entity.

Problem (Temporal Entity-Aspect Recommendation): *Given an event entity e and hitting time t as input, find the ranked list of entity aspects that most relevant with regards to e and t.*

Different from time-aware entity recommendation [ZRZ16, TTN], for an entity query with exploratory intent, users are not just interested in related entities, but also entity aspects (which can be a topic, a concept or even an entity); these provide more complete and useful information. These aspects are very time-sensitive especially when the original entity is about an event. In this work, we use the notion of *event entity*, which is generalized to indicated related entities of any trending events. For example, Moonlight and Emma Stone are related entities for the 89th Academy Awards event. We will handle the aspects for such entities in a temporally aware manner.

3.1.3 Multiple entity-aspect ranking models

As event entity identification has been well-explored in related work [KNNN15, KRM16, KSLP$^+$], we do not suggest a specific method, and just assume the use of an appropriate method. Given an event entity, we then apply our aspect recommendation method, which is composed of three main steps. We summarize the general idea of our approach in Figure 3.3. First, we extract suggestion candidates using a bipartite graph of co-clicked query-URLs generated at hitting time. After the aspect extraction, we propose a *two-step* unified framework for our entity aspect ranking problem. The first step is to identify event type and time in a joint learning approach. Based on that, in the second step, we divide the training task to different sub-tasks that correspond to specific event type and time. Our intuition here is that the timeliness (or short-term interest) feature-group might work better for specific subsets such as breaking and after events and vice versa. Dividing the training will avoid timeliness and salience competing with each other and maximize their effectiveness. However, identifying time and type of an event on-the-fly is not a trivial task, and breaking the training data into smaller parts limits the learning power of the individual models. We therefore opt for an ensemble approach that can utilize the whole training data to (1) supplement the uncertainties of the time-and-type classification in the first step and (2) leverage the learning power of the sub-models in step 2. In the rest of this section, we explain our proposed approach in more detail.

3.1.4 Aspect Extraction

The main idea of our approach for extracting aspects is to find related entity-bearing queries; then group them into different clusters, based on *lexical* and *semantic* similarity, such that each cluster represents a distinct aspect. The click-through information can help identifying related queries [Sil10] by exploiting the assumption that any two queries which share many clicked URLs are likely to be related to each other.

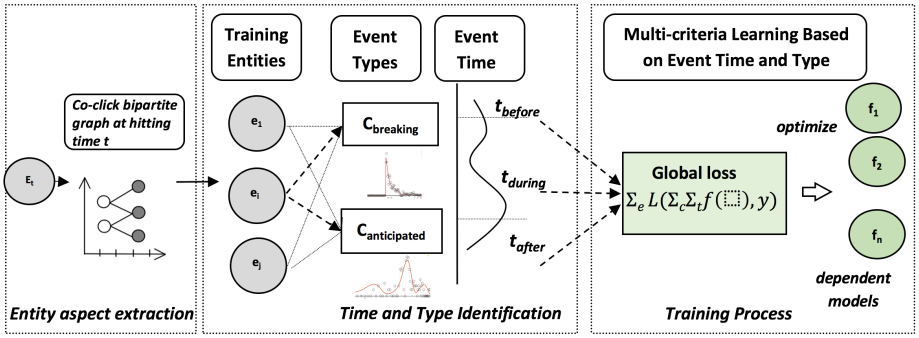


Figure 3.3: Learning time and type-specific ranking models.

For a given entity query e, we perform the following steps to find aspect candidates. We retrieve a set of URLs U_e that were clicked for e from the beginning of query logs until the hitting time t_e. For each $u_j \in U_e$, we find a set of distinct queries for which u_j has been clicked. We give a weight w to each query-URL by normalizing *click frequency* and *inverse query frequency* (CF-IQF) [DKL], which calculate the importance of a click, based on click frequency and inverse query frequency. $CF - IQF = cf \cdot log(N/(qf+1))$, where N is the number of distinct queries. A high weight $CF - IQF$ indicates a high click frequency for the query-URL pair and a low query frequency associated with the URL in the whole query log. To extract aspect candidates from the click bipartite graph, we employ a personalized random walk to consider only one side of the query vertices of the graph (we denote this approach as **RWR**). This results in a set of related queries (aspects) to the source entity e, ranked by click-flow relatedness score. To this end, we refine these extracted aspects by clustering them using Affinity Propagation (AP) on the similarity matrix of *lexical* and *semantic* similarities. For semantic measure, we use a *word2vec* skip-gram model trained with the English Wikipedia corpus from the same time as the query logs. We pick one aspect with highest frequency to represent each cluster, then select top-k aspects by ranking them using RWR relatedness scores [2].

Time and Type Identification

Our goal is to identify the probability that an event-related entity is of a specific event type, and in what time period of the event. We define these two targets as a joint-learning time-series classification task, that is based on event diffusion. In the following, we first present the feature set for the joint-learning task, then explain the learning model. Last we propose a light-weight clustering approach that leverages the learning features, to integrate with the

[2] About complexity analysis, the click bipartite graph construction costs $O(m+n)$ and RWR in practice, can be bounded by $O(m+n)$ for top-k proximity nodes. Note that m,n are the number of edges and nodes respectively. AP is quadratic $O(kn^2)$ time, (with k is the number of iterations), of our choice as we aim for a simple and effective algorithm and our aspect candidate sets are not large. A more efficient algorithm such as the Hierarchical AP can be used when candidate sets are large. The cost of constructing the similarity matrix is $O(n^2)$.

ranking model in Section 3.1.4.

Features. We propose a set of time series features for our multi-class classification task. *seasonality* and *periodicity* are good features to capture the *anticipated* -recurrent events. In addition, we use additional features to model the temporal dynamics of the entity at studied/hitting time t_e. We leverage query logs and Wikipedia revision edits as the data sources for *short* and *long* span time series construction, denoted as $\psi_Q^{(e)}$ and $\psi_{WE}^{(e)}$ (for seasonal, periodical event signals) respectively [3]. The description of our features follows:

- **Seasonality** is a temporal pattern that indicates how periodic is an observed behavior over time. We leverage this time series decomposition technique for detecting not only seasonal events (e.g., Christmas Eve, US Open) [Sho] but also more fine-grained periodic ones that recurring on a weekly basis, such as a TV show program.

- **Autocorrelation**, is the cross correlation of a signal with itself or the correlation between its own past and future values at different times. We employ autocorrelation for detecting the trending characteristics of an event, which can be categorized by its predictability. When an event contains strong inter-day dependencies, the autocorrelation value will be high. Given observed time series values $\psi_1, ..., \psi_N$ and its mean $\bar{\psi}$, autocorrelation is the similarity between observations as a function of the time lag l between them. In this work, we consider autocorrelation at the one time unit lag only (l = 1), which shifts the second time series by one day.

- **Correlation coefficient**, measures the dynamics of two consecutive aspect ranked lists at time t_e and $t_e - 1$, return by **RWR**. We use Goodman and Kruskal's gamma to account for possible new or old aspects appear or disappear in the newer list.

- **Level of surprise**, measured by the error margin in prediction of the learned model on the time series. This is a good indicator for detecting the starting time of *breaking* events. We use Holt-Winters as the predictive model.

- **Rising and falling signals.** The intuition behind time identification is to measure whether $\psi_Q^{(e)}$ is going up (*before*) or down (*after*) or stays trending (*during*) at hitting time. Given $\psi_Q^{(e)}$, we adopt an effective parsimonious model called SpikeM [MSP+12], which is derived from epidemiology fundamentals to predict the rise and fall of event diffusion. We use the *Levenberg-Marquardt* algorithm to learn the parameter set and use the parameters as features for our classification task.

Learning model. We assume that there is a semantic relation between the event types and times (e.g., the before phase of *breaking* events are different from *anticipated*). To leverage the dependency between the ground labels of the two classification tasks, we apply a joint learning approach that models the two tasks in a cascaded manner, as a simple

[3] Wikipedia page views is an alternative, however it is not publicly available for the time of our query logs, 2006

version of [HGSK09]. Given the same input instance $\mathscr{I}$, the 1^{st} stage of the cascaded model predicts the event type $\mathscr{C}$ with all proposed features. The trained model $\mathscr{M}^1$ is used in the 2^{nd} stage to predict the event time $\mathscr{T}$. We use the logistic regression model $\mathscr{M}^2_{LR}$ for the 2^{nd} stage, which allows us to add additional features from $\mathscr{M}^1$. The feature vector of $\mathscr{M}^2_{LR}$ consists of the same features as $\mathscr{M}^1$, together with the probability distribution of $P(\mathscr{C}_k|e,t)$ (output of $\mathscr{M}^1$) of as additional features.

Ranking-sensitive time and type distribution. The output of an effective classifier can be directly used for determining a time and type probability distribution of entities; and thus dividing the training entities into subsets for our *divide-and-conquer* ranking approach. However, having a pre-learned model with separate and large training data is expensive and could be detrimental to ranking performance if the training data is biased. We therefore opt for effective on-the-fly *ranking-sensitive* time and type identification, following [BLL$^+$] that utilizes the 'locality property' of feature spaces. We adjust and refine the approach as follows. Each entity is represented as a feature vector, and consists of all proposed features with importance weights learned from a sample of training entities (for ranking). We then employ a Gaussian mixture model to obtain the centroids of training entities. In our case, the number of components for clustering are fixed before hand, as the number of event types multiplied by the number of event times. Hence the probability distribution of entity e at time t belonging to time and type $\mathscr{T}_l, \mathscr{C}_k$, $P(\mathscr{T}_l, \mathscr{C}_k|e,t)$ is calculated as $1 - \frac{\mathbf{x}^e - \mathbf{x}^2_{c_{\mathscr{T}_l,\mathscr{C}_k}}}{\max_{\forall T,C} \mathbf{x}^e - \mathbf{x}^2_{c_{\mathscr{T}_l,\mathscr{C}_k}}}$, or the distance between feature vector $\mathbf{x}^e$ and the corresponding centroid $c_{\mathscr{T}_l,\mathscr{C}_k}$.

Time and Type-Dependent Ranking Models

Learning a single model for ranking event entity aspects is not effective due to the dynamic nature of a real-world event driven by a great variety of multiple factors. We address two major factors that are assumed to have the most influence on the dynamics of events at aspect-level, i.e., time and event type. Thus, we propose an adaptive approach based on the ensemble of multiple ranking models learned from training data, which is partitioned by entities' temporal and type aspects. In more detail, we learn multiple models, which are co-trained using data *soft* partitioning / clustering method in Section 3.1.4, and finally combine the ranking results of different models in an ensemble manner. This approach allows sub-models to learn for different types and times (where feature sets can perform differently), without hurting each other. The adaptive global loss then co-optimizes all sub-models in a unified framework. We describe in details as follows.

Ranking Problem. For aspect ranking context, a typical ranking problem is to find a function f with a set of parameters ω that takes aspect suggestion feature vector $\mathscr{X}$ as input and produce a ranking score $\hat{y}$: $\hat{y} = \mathsf{f}(\mathscr{X}, \omega)$. In a learning to rank paradigm, it is aimed at finding the best candidate ranking model f^* by minimizing a given loss function $\mathscr{L}$ calculated as: $\mathsf{f}^* = \arg\min_f \sum_{\forall a} \mathscr{L}(\hat{y}_a, y_a)$.

Multiple Ranking Models. We learn multiple ranking models trained using data constructed from different time periods and types, simultaneously, thus producing a set of ranking models $\mathbf{M} = \{M_{\mathscr{T}_1,\mathscr{C}_1}, \ldots, M_{\mathscr{T}_m,\mathscr{C}_n}\}$, where $\mathscr{T}_i$ is an event time period, $\in \mathscr{T}$, and $\mathscr{C} = \{\mathscr{C}_1, \mathscr{C}_2, \ldots, \mathscr{C}_n\}$ are the types of an event entity. We use an ensemble method that combines results from different ranking models, each corresponding to an identified ranking-sensitive query time $\mathscr{T}$ and entity type $\mathscr{C}$. The probabilities that an event entity e belongs to time period $\mathscr{T}_l$ and type $\mathscr{C}_k$ given the hitting time t is $P(\mathscr{T}_l, \mathscr{C}_k|e,t)$, and can be computed using the time and type identification method presented in Section 3.1.4.

$$\mathsf{f}^* = \arg\min_f \sum_{\forall a} \mathscr{L}(\sum_{k=1}^{n} P(\mathscr{C}_k|a,t) \sum_{l=1}^{m} P(\mathscr{T}_l|a,t,\mathscr{C}_k)\hat{y_a}, y_a) \tag{3.1}$$

Multi-Criteria Learning. Our task is to minimize the global relevance loss function, which evaluates the overall training error, instead of assuming the independent loss function, that does not consider the correlation and overlap between models. We adapted the L2R RankSVM [Joa06]. The goal of RankSVM is learning a linear model that minimizes the number of discordant pairs in the training data. We modified the objective function of RankSVM following our global loss function, which takes into account the temporal feature specificities of event entities. The temporal and type-dependent ranking model is learned by minimizing the following objective function:

$$\begin{aligned} \min_{\omega,\xi,e,i,j} \frac{1}{2}||\omega||^2 + C\sum_{e,i,j} \xi_{e,i,j} \\ \text{subject to, } \sum_{k=1}^{n} P(\mathscr{C}_k|e,t) \sum_{l=1}^{m} P(\mathscr{T}_l|e,t,\mathscr{C}_k)\omega_{kl}^T X_i^e \\ \geq \sum_{k=1}^{n} P(\mathscr{C}_k|e,t) \sum_{l=1}^{m} P(\mathscr{T}_l|e,t,\mathscr{C}_k)\omega_{kl}^T X_j^e + 1 - \xi_{e,i,j}, \\ \forall X_i^e \succ X_j^e, \xi_{e,i,j} \geq 0. \end{aligned} \tag{3.2}$$

where $P(\mathscr{C}_k|e,t)$ is the probability the event entity e, at time t, is of type $\mathscr{C}_k$, and $P(\mathscr{T}_l|e,t,\mathscr{C}_k)$ is probability e is in this event time $\mathscr{T}_l$ given the hitting-time t and $\mathscr{C}_k$. The other notions are inherited from the traditional model ($X_i^q \succ X_j^e$ implies that an entity aspect i is ranked ahead of an aspect j with respect to event entity e. C is a trade-off coefficient between the model complexity $||\omega||$ and the training error $\xi_{a,i,j}$.

Ensemble Ranking. After learning all time and type-dependent sub models, we employ an unsupervised ensemble method to produce the final ranking score. Supposed $\bar{a}$ is a testing entity aspect of entity e. We run each of the ranking models in $\mathbf{M}$ against the instance of $\bar{a}$, multiplied by the time and type probabilities of the associated entity e at hitting time t. Finally, we sum all scores produced by all ranking models to obtain the ensemble ranking, $score(\bar{a}) = \sum_{m \in M} P(\mathscr{C}_k|e,t)P(\mathscr{T}_l|e,t,\mathscr{C}_k)\mathsf{f^*}_m(\bar{a})$.

Ranking Features

We propose two sets of features, namely, (1) *salience* features (taking into account the general importance of candidate aspects) that mainly mined from Wikipedia and (2) *short-term interest* features (capturing a trend or timely change) that mined from the query logs. In addition, we also leverage click-flow relatedness features computed using RWR. The features from the two categories are explained in details as follows.

***Salience* features** - or in principle, long-term prominent features.

- **TF.IDF** of an aspect a is the average $TF.IDF(w)$ of all terms $w \in a$; $TF.IDF(w)$ is calculated as $tf(w,D)\dot{l}og\frac{N}{df(w)}$, whereas D is a section in the related Wikipedia articles C of entity e. To construct C, we take all in-link articles of the corresponding Wikipedia article of e; $tf(w,D)$ is the term frequency, $df(w)$ denotes the number of sections which w appears.

- **MLE-based**, where we reward the more (cumulated) frequently occurring aspects from the query logs. The maximum likelihood s_{MLE} is $\frac{sum_{w\in a}n(w,e)}{\sum_{a'}\sum_{t\in a'}f(w,e)}$, where $f(w,e)$ denotes the frequency a segment (word or phrase) $w \in a$ co-occurs with entity e.

- **Entropy-based**, where we reward the more "stable" aspects over time from the query logs. The entropy is calculated as: $\mathsf{s}_E = \sum_{t\in T} P(a|t,e)logP(a|t,e)$, where $P(a|t,e)$ is the probability of observing aspect a in the context of entity e at time t.

- **Language Model-based**, how likely aspects are generated by as stastical LM based on the textual representation of the entity $\mathsf{d}(e)$. We model $\mathsf{d}(e)$ as the corresponding Wikipedia article text. We use the unigram model with default Dirichlet smoothing.

***Short-term* interest features**, are described as follows.

- **Temporal click entropy.** Click entropy [DSW] is known as the measurement of how much diversity of clicks to a particular query over time. In detail, the click entropy is measured as the query click variation over a set of URLs for a given query q. In this work, a temporal click entropy accounts for only the number of clicks on the time unit that the entity query is issued. The temporal click entropy TCE_t can be computed as $\sum\limits_{u\in U_q} -P(u|q)\log P(u|q)$ where U_q is a set of clicked URLs for a given query q at time t. The probability of u being clicked among all the clicks of q, $P(u|q)$ is calculated as $\frac{|click(u,q)|}{\sum_{u_i\in U_q}|click(u_i,q)|}$.

- **Trending momentum** measures the trend of an aspect based on the query volume. The trending momentum at time t, Tm_t is calculated using the moving average (Ma) technique, i.e., $Tm_t = Ma(t,i_s) - Ma(t,i_l)$. Whereas, i_s,i_l denotes the short and long time window from the hitting time.

- **Cross correlation** or temporal similarity, is how correlated the aspect *wrt.* the main entity. The more cross-correlated the temporal aspect to the entity, the more influence it brings to the global trend. Given two time series ψ_t^e and ψ_t^a of the entity and aspect at time t, we employ the cross correlation technique to measure such correlation. Cross correlation $CCF(\psi_t^e, \psi_t^a)$ gives the correlation score at lagging times. Lagging time determines the time delay between two time-series. In our case, as we only interest in the hitting time, we take the maximum CCF in a lag interval of $[-1, 1]$.

- **Temporal Language Model-based**, similar to the *salient* feature, only the textual representation $d(e)$ is the aggregated content of top-k most clicked URLs at time t.

3.1.5 Evaluation

In this section, we explain our evaluation for assessing the performance of our proposed approach. We address three main research questions as follows:

RQ1: How good is the classification method in identifying the most relevant event type and period with regards to the hitting time?

RQ2: How do long-term salience and short-term interest features perform at different time periods of different event types?

RQ3: How does the ensemble ranking model perform compared to the single model approaches?

In the following, we first explain our experimental setting including the description of our query logs, relevance assessment, methods and parameters used for the experiments. We then discuss experimental results for each of the main research questions.

Experimental Setting

Datasets. We use a real-world query log dataset from AOL, which consists of more than 30 million queries covering the period from March 1, to May 31, 2006. Inspired by the taxonomy of event-related queries presented in [KMT+13], we manually classified the identified events into two distinct subtypes (i.e., *Breaking* and *Anticipated*). We use Tagme [4] to link queries to the corresponding Wikipedia pages. We use the English Wikipedia dump of June, 2006 with over 2 million articles to temporally align with the query logs. The Wikipedia page edits source is from 2002 up to the studied time, as will be explained later. To count the number of edits, we measure the difference between consecutive revision pairs extracted from the Special:Export [5].

Identifying event entities. We reuse the event-related queryset from [KNNN15], that contains 837 entity-bearing queries. We removed queries that refer to past and future events and only chose the ones which occured in the period of the AOL dataset, which results in

[4] https://tagme.d4science.org/tagme/

[5] https://en.wikipedia.org/wiki/Special:Export

Table 3.1: Dynamic relevant assessment examples.

Entity	Suggestion	Dynamic Label		
		Before	During	After
kentucky derby	+ odds	VR	VR	R
kentucky derby	+ contenders	VR	R	R
kentucky derby	+ winner	NR	R	VR
kentucky derby	+ results	NR	VR	VR

300 distinct entity queries. Additionally, we construct a more recent dataset which consists of the volume of searches for 500 trending entity queries on Google Trend. The dataset covers the period from March to May, 2017. To extract these event-related queries, we relied on the Wikipedia Portal:Current events[6] as the external indicator, as we only access Google query logs via public APIs. Since the click logs are missing, the Google Trend queryset is used only as a supplementary dataset for *RQ1*.

Dynamic Relevance Assessment. There is no standard ground-truth for this novel task, so we relied on manual annotation to label entity aspects dynamically; with respect to the studied times according to each event period. We put a range of 5 days before the event time as *before* period and analogously for *after*. We randomly picked a day in the 3 time periods for the studied times. In our annotation process, we chose 70 popular and trending event entities focusing on two types of events, i.e., *Breaking* (30 queries) and *Anticipated* (40 queries). For each entity query, we make used of the top-k ranked list of candidate suggestions generated by RWR, cf. Section 3.1.4. Four human experts were asked to evaluate a pair of a given entity and its aspect suggestion (as relevant or non-relevant) with respect to the event period. We defined 4 levels of relevance: 3 (very relevant), 2 (relevant), 1 (irrelevant) and 0 (don't know). Finally, 4 assessors evaluated 1,250 entity/suggestion pairs (approximately 3,750 of triples), with approximately 17 suggestions per trending event on average. The average Cohen's Kappa for the evaluators' pairwise inter-agreement is k = 0.78. Examples of event entities and suggestions with dynamic labels are shown in Table 3.1. The relevance assessments will be made publicly available.

Methods for Comparison. Our baseline method for aspect ranking is RWR, as described in Section 3.1.4. Since we conduct the experiments in a query log context, time-aware query suggestions and auto-completions (QACs) are obvious competitors. We adapted features from state-of-the-art work on time-aware QACs as follows. For the QACs' setting, entity name is given as prior. Instead of making a direct comparison to the linear models in [RMdR15] – that are tailored to a different variant of our target – we opt for the supervised-based approach, $SVM_{salient}$, which we consider a fairer and more relevant salient-favored competitor for our research questions.

Most popular completion (**MLE**) [BYK] is a standard approach in QAC. The model can be regarded as an approximate Maximum Likelihood Estimator (MLE), that ranks the suggestions based on past popularity. Let $P(q)$ be the probability that the next query is q. Given a prefix x, the query candidates that share the prefix $\mathcal{Q}_c$, the most likely suggestion

[6] https://en.wikipedia.org/wiki/Portal:Current_events

Table 3.2: Example entities in May 2006.

anticipated	may day, da vinci code, cinco de mayo, american idol,
	anna nicole smith, mother's day, danica patrick, emmy rossum,
	triple crown, preakness stakes, belmont stakes kentucky derby, acm awards
breaking	david blaine, drudge report, halo 3, typhoon chanchu,
	patrick kennedy, indonesia, heather locklear

$q \in \mathcal{Q}_c$ is calculated as: $MLE(x) = argmax_{q \in \mathcal{Q}_c} P(q)$. To give a fair comparison, we apply this on top of our aspect extraction cf. Section 3.1.4, denoted as $RWR+MLE$; analogously with recent MLE.

Recent MLE (**MLE-W**) [WJ, SR] does not take into account the whole past query log information like the original MLE, but uses only recent days. The popularity of query q in the last n days is aggregated to compute $P(q)$.

Last N query distribution (**LNQ**) [WJ, SR] differs from MLE and W-MLE and considers the last N queries given the prefix x and time x_t. The approach addresses the weakness of W-MLE in a time-aware context, having to determine the size of the sliding window for prefixes with different popularities. In this approach, only the last N queries are used for ranking, of which N is the trade-off parameter between *robust* (non time-aware bias) and *recency*.

Predicted next N query distribution (**PNQ**) employs the past query popularity as a prior for predicting the query popularity at hitting time, to use this prediction for QAC [WJ, SR]. We adopt the prediction method proposed in [SR].

Parameters and settings. The jumping probability for RWR is set to 0.15 (default). For the classification task, we use models implemented in Scikit-learn [7] with default parameters. For learning to rank entity aspects, we modify RankSVM. For each query, the hitting time is the same as used for relevance assessment. Parameters for RankSVM are tuned via grid search using 5-fold cross validation (CV) on training data, trade-off $c = 20$. For W-MLE, we empirically found the sliding window $W = 10$ days. The time series prediction method used for the PNQ baseline and the prediction error is Holt-Winter, available in R. In LNQ and PNQ, the trade-off parameter N is tuned to 200. The short-time window i_s for the trending momentum feature is 1-day and long i_l is 5-days. Top-k in the temporal LM is set to 3. The time granularity for all settings including hitting time and the time series binning is 1 day.

For RQ1, we report the performance on the *rolling* 4-fold CV on the whole dataset. To seperate this with the L2R settings, we explain the evaluating methodology in more details in Section 3.1.5. For the ranking on partitioned data (RQ2), we split *breaking* and *anticipated* dataset into 6 sequential folds, and use the last 4 folds for testing in a rolling manner. To evaluate the ensemble method (RQ3), we use the first two months of AOL for training (50 queries, 150 studied points) and the last month (20 queries as shown in Table 3.2, 60 studied points) for testing.

Metrics. For assessing the performance of classification methods, we measured accu-

[7] http://scikit-learn.org/

Table 3.3: Event type and time classification performance.

	Dataset	Model	Accuracy	Weighted F1
Event-type	AOL	**majority votes**	0.64	0.58
		SVM	**0.79**	**0.89**
	GoogleTrends	**majority votes**	0.61	0.68
		SVM	**0.83**	**0.85**
Event-time	AOL	**Logistic Regression**	0.68	0.72
		Cascaded	**0.73**	**0.83**
	GoogleTrends	**Logistic Regression**	0.71	0.78
		Cascaded	**0.75**	**0.82**

racy and F1. For the retrieval effectiveness of query ranking models, we used two metrics, i.e., Normalized Discounted Cumulative Gain (NDCG) and $recall@k$ ($r@k$). We measure the retrieval effectiveness of each metric at 3 and 10 ($m@3$ and $m@10$, where $m \in \{NDCG, R\}$). $NDCG$ measures the ranking performance, while $recall@k$ measures the proportion of relevant aspects that are retrieved in the top-k results.

Cascaded Classification Evaluation

Evaluating methodology. For **RQ1**, given an event entity e, at time t, we need to classify them into either *Breaking* or *Anticipated* class. We select a studied time for each event period randomly in the range of 5 days before and after the event time. In total, our training dataset for AOL consists of 1,740 instances of *breaking* class and 3,050 instances of *anticipated*, with over 300 event entities. For *GoogleTrends*, there are 2,700 and 4,200 instances respectively. We then bin the entities in the two datasets chronologically into 10 different parts. We set up 4 trials with each of the last 4 bins (using the history bins for training in a *rolling* basic) for testing; and report the results as average of the trials.

Results. The baseline and the best results of our 1^{st} stage event-type classification is shown in Table 3.3-**top**. The accuracy for basic majority vote is high for imbalanced classes, yet it is lower at weighted F1. Our learned model achieves marginally better result at F1 metric.

We further investigate the identification of event time, that is learned on top of the event-type classification. For the gold labels, we gather from the studied times with regards to the event times that is previously mentioned. We compare the result of the cascaded model with non-cascaded logistic regression. The results are shown in Table 3.3-**bottom**, showing that our cascaded model, with features inherited from the performance of SVM in previous task, substantially improves the single model. However, the overall modest results show the difficulty of this multi-class classification task.

Ranking Aspect Suggestions

For this part, we first focus on evaluating the performance of single L2R models that are learned from the pre-selected time (before, during and after) and types (*Breaking* and *Anticipate*) set of entity-bearing queries. This allows us to evaluate the feature performance i.e., *salience* and *timeliness*, with time and type specification (RQ2). We then evaluate

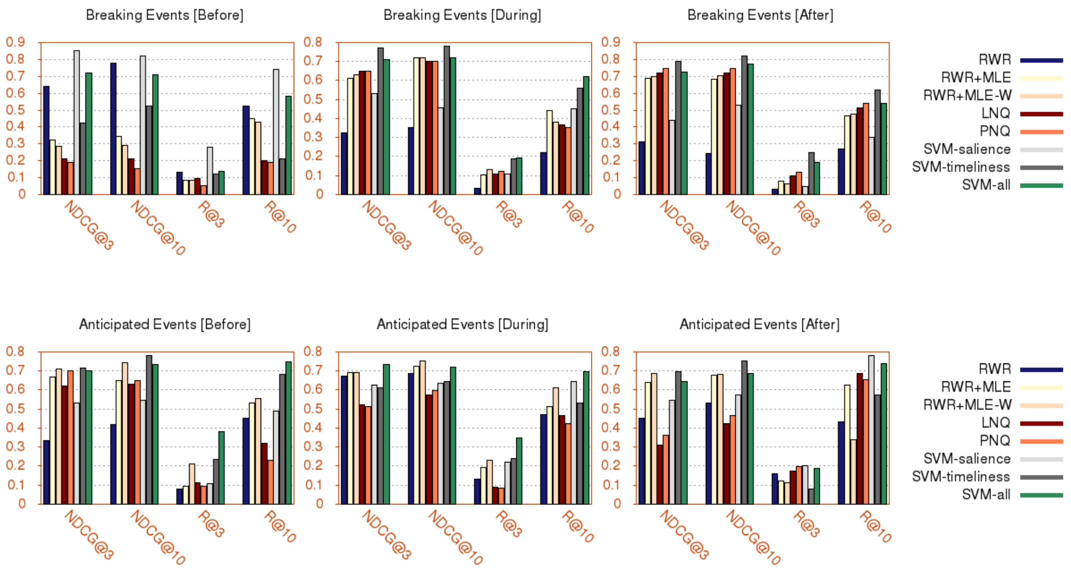


Figure 3.4: Performance of different models for event entities of different types.

our ensemble ranking model (results from the cascaded evaluation) and show it robustly improves the baselines for all studied cases (RQ3). Notice that, we do not use the learned classifier in Section 3.1.5 for our ensemble model, since they both use the same time period for training, but opt for the *on-the-fly* ranking-sensitive clustering technique, described in Section 3.1.4.

RQ2. Figure 4.13 shows the performance of the aspect ranking models for our event entities at specific times and types. The most right three models in each metric are the models proposed in this work. The overall results show that, the performances of these models, even better than the baselines (for at least one of the three), vary greatly among the cases. In general, $SVM_{salience}$ performs well at the **before** stage of breaking events, and badly at the **after** stage of the same event type. Whereas $SVM_{timeliness}$ gives a contradictory performance for the cases. For anticipated events, $SVM_{timeliness}$ performs well at the **before** and **after** stages, but gives a rather low performance at the **during** stage. For this event type, $SVM_{salience}$ generally performs worse than $SVM_{timeliness}$. Overall, The SVM_{all} with all features combined gives a good and stable performance, but for most cases, are not better than the well-performed single set of features L2R model. In general, these results prove our assumption that *salience* and *timeliness* should be traded-off for different event types, at different event times. For feature importances, we observe regularly, stable performances of *same-group* features across these cases. *Salience* features from knowledge bases tend to perform better than from query logs for *short-duration* or less popular events. We leave the more in-depth analysis of this part for future work.

RQ3. We demonstrate the results of single models and our ensemble model in Table 3.4. As also witnessed in RQ2, SVM_{all}, will all features, gives a rather stable performance for both NDCG and Recall, improved the baseline, yet not significantly. Our *Ensemble* model,

Table 3.4: Performance of the baselines (RWR relatedness scores, RWR+MLE, RWR+MLE-W, LNQ, and PNQ) compared with our ranking models; $*$,$\dagger$, $\mp$ indicates statistical improvement over the baseline using t-test with significant at $p < 0.1$, $p < 0.05$, $p < 0.01$ respectively.

Methods	NDCG@3	NDCG@10	R@3	R@10
RWR	*0.3208*	*0.4137*	*0.1208*	*0.3749*
RWR+MLE	+29.94%	+9.73%	-21.09%	+5.15%$*$
RWR+MLE-W	+11.56%	+11.46%	-18.93%$*$	+3.28%
LNQ	+15.39%	-3.75%	-19.74%	-30.31%
PNQ	+13.19%	-9.95%	-23.46%	-33.53%
$SVM_{salience}$	+41.75%$*$	+9.18%	+23.32%$*$	+9.93%
$SVM_{timeliness}$	+15.19%	+17.53%	+14.77%	+11.3%
SVM_{all}	+52.65%$*$	+40.87%$*$	+9.73%$\dagger$	+24.3%
Ensemble	**+85.12%**$\mp$	**+45.34%**$\dagger$	**+42.78%**$*$	**+17.45%**$*$

that is learned to trade-off between *salience* and *timeliness* achieves the best results for all metrics, outperforms the baseline significantly. As the testing entity queries in this experiment are at all event times and with all event types, these improvements illustrate the robustness of our model. Overall, we witness the low performance of adapted QAC methods. One reason is as mentioned, QACs, even time-aware generally favor already *salient* queries as follows the *rich-get-richer* phenomenon, and are not ideal for entity queries that are event-related (where aspect relevance can change abruptly). Time-aware QACs for partially long prefixes like entities often encounter sparse traffic of query volumes, that also contributes to the low results.

3.1.6 Conclusion

We studied the temporal aspect suggestion problem for entities in knowledge bases with the aid of real-world query logs. For each entity, we ranked its temporal aspects using our proposed novel time and type-specific ranking method that learns multiple ranking models for different time periods and event types. Through extensive evaluation, we also illustrated that our aspect suggestion approach significantly improves the ranking effectiveness compared to competitive baselines. In this work, we focused on a "global" recommendation based on public attention. The problem is also interesting taking other factors (e.g., *search context*) into account, which will be interesting to investigate in future work.

3.2 Diversification of Entity-aspects for Web Ranking

A significant fraction of web search queries are ambiguous, or contain multiple aspects or subtopics [CCS09]. For example, the query apple can refer to a kind of fruit or a company selling computer products. Moreover, the underlying aspects of the query apple inc can be a new Apple product, software updates or it latest press releases. While it is difficult to

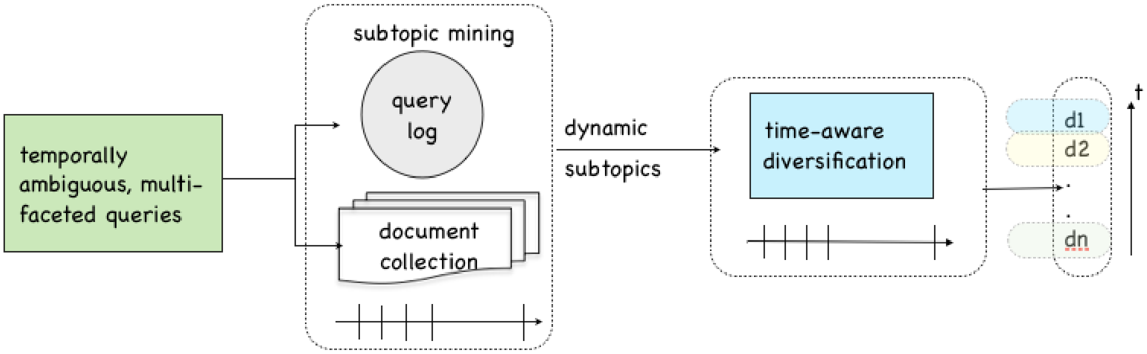

temporally ambiguous, multi-faceted queries
subtopic mining
query log
document collection
dynamic subtopics
time-aware diversification
d1
d2
dn
t

Figure 3.5: Pipeline for dynamic subtopic mining and time-aware diversification

identify user's search intent for multi-faceted queries, it is common to present results with a high coverage of relevant aspects. This problem has been well studied in aforementioned work on search result diversification [AGHI09a, CG98, CC09, DHC$^+$11, RBS10, SMO10]. However, previous work only consider a set of static subtopics without taking into account the temporal dynamics of query subtopics.

In this section, we study the search result diversification of *temporally ambiguous, multi-faceted queries*, where the relevance of query subtopics is highly time-dependent. For example, when issuing the query kentucky derby in April, relevant aspects are likely to be about "festival" or "food" referring to the Kentucky Derby Festival, which occurs two weeks before the stakes race. However, at the end of May, other facets like "result" and "winner" should be more relevant to than pre-event aspect. Identifying dynamic subtopics for temporally ambiguous, multi-faceted queries is essential for time-aware search result diversification. In order that, we explicitly extract dynamic subtopics and leverage them into diversifying retrieved results. To the best of our knowledge, none of the aforementioned works considers the temporal changes in query subtopics before.

Our contributions in this section are as follows. We study the temporal dynamics of subtopics for queries, which are *temporally ambiguous* or *multi-faceted*. We analyze the temporal variability of query subtopics by applying subtopic mining techniques at different time periods. In addition, our analysis results reveal that the popularity of query aspects changes over time, which is possibly the influence of a real-world event. The analysis study is based on two data sources, namely, query logs and a temporal document collection, where time information is available. To this end, we propose different time-aware search result diversification methods, which leverage dynamic subtopics and show the performance improvement over the existing non time-aware methods.

3.2.1 Dynamic Subtopic Mining

In this section, we present our methodology in modeling and mining temporal subtopics from two different datasets. The mined subtopics are input for our time-aware diversification approach. Figure 6.2 depicts our proposed system pipeline.

Mining Subtopics from Query Logs

In our work, we followed a state-of-the-art finding related queries technique proposed in [CS07]. We applied Markov random walk with restart (RWR) on the weighted bipartite graph composed of two sets of nodes, namely, queries and URLs. The bipartite graph is constructed using the history information with regards to different time points. Our model for dynamic subtopic mining assigns each subtopic a *temporal weight* that reflects the probability of the relevance of a subtopic at the particular time.

Clustering subtopic candidates Random walk with restart on the click-through graph provides us a set of related queries. However, these related queries can be duplicated or near-duplicated in their semantics. To achieve finer-grained query subtopics at hitting time q_t, we cluster the acquired queries in a similar approach proposed in [SZG$^+$11]. The steps are as follows: (1) Construct a query similarity matrix (using lexical, click and semantic similarity), (2) Cluster related queries (using Affinity Propagation technique), and (3) Extract dynamic query subtopics. Due to the limited space in this thesis, readers can refer to [SZG$^+$11] for detailed description of the steps.

Temporal subtopic weight We calculate the subtopic weight from query log $w_{query\_log}(c)$ of a subtopic c to a query q solely based on the relatedness score from performing the RWR. For each query cluster $\mathbb{C}_i$ that represents a subtopic c_i, the weight of c, $w(c)$ is the proportion between the total RWR score of all queries in $\mathbb{C}_i$ and of all related queries.

Mining Subtopics from a Temporal Document Collection

In this section, we make use of Latent Dirichlet Allocation (LDA) [BNJ03], an unsupervised method to mine and model latent query subtopics from a relevant set of documents D. Relevant sets of documents are captured at fixed time periods in order to measure the variance of the mined latent subtopics over time. Here, a subtopic $c \in C$ is modeled as multinomial distribution of words, a document $d \in D$ composes of a mixture of topics.

Estimating number of subtopics Deciding the optimum number of subtopics is an important task for assessing the overall query subtopic dynamics. The number of subtopics is expected to change when mining it at different time points. In this work, we follow the approach that proposed by Arun et al. [ASVMNM10] to identify the number of latent subtopics that are naturally present in each partition. The non-optimum number of subtopics produces the high divergence between the salient distributions derived from two matrix factors (compose of topics-words and documents-topics). In our case, we set the number of topics in a pre-defined range from γ to δ, the chosen number of topic is the one with the minimum KL-divergence value.

Temporal subtopic weight We estimate the weight of a mined subtopic at every hitting time. The weight $w_{docs}(c)$ of a subtopic c reflects the probability that a given query q implies the subtopic c. The temporal distribution that specifies the probability that a given query belongs to a subtopic c, $Pr(c|q)$ derives from the popularity of the subtopic in the studied time slice of the document collection. It is calculated as the proportion between the total probabilities of all documents belongs to a subtopic $Pr(c|d)$ and the number of documents in the time slice. $Pr(c|d)$ is calculated from the Dirichlet prior topic distribution of LDA.

3.2.2 Time-aware diversification

Most of the existing diversification approaches in related work deploy a greedy approximation approach. We examine three state-of-the-art diversification models (i.e., IA-Select [AGHI09a], xQuaD [SMO10] and topic-richness [DHC+11]). We aim to maximize the utility of the models by fostering recent documents in the ranking, with the assumption that the recency level of a subtopic is linearly proportional to its temporal popularity.

temp-IA-Select The objective function of IA-Select can be expressed using a probabilistic model as:

$$f_s(d) = \sum_c Pr(q|d)Pr(d|c)Pr(c|q) \prod_{d\prime \in S} (1 - Pr(q|d\prime)Pr(d\prime|c)) \tag{3.3}$$

where S is the selected set of diversified documents from the original result set. Our assumption is our temporal mined subtopics are fresh subtopics and the subtopics tend to favor recent documents. We propose an exponential distribution on the probability of documents $Pr(d)$ with regards to a subtopic c. The document-subtopic probability $Pr(d|c)$ at time t_d, defined as $Pr_{t_d}(d|c)$ is calculated in Equation 3.4.

$$Pr_{t_d}(d|c) = Pr(c|d)Pr(d|t_d) = Pr(c|d) \cdot \lambda \cdot e^{-\lambda \cdot t_d} \tag{3.4}$$

We apply $Pr_{t_d}(d|c)$ into the probabilistic objective function of IA-Select to achieve our time-aware objective function (temp-IA-Select), described in Equation 3.5. With this approach, a document d which is published closer to the hitting time t_q, in essence, has a shorter age t_d will be weighted higher than the one with the same $Pr(c|d)$. Note that for this setting, we do not account for time to calculate the document-query probability, $Pr(d|q)$, that remains unchanged over time. Our intuition is to leverage only exponential distribution of a document d towards certain subtopic d in favoring recent documents in the task of diversifying search results (according to the mined subtopics C).

$$f_s(d) = \sum_c Pr(c|q)Pr(q|d)Pr(c|d) \cdot \lambda \cdot e^{-\lambda \cdot t_d} \prod_{d\prime \in S} (1 - Pr(q|d\prime)Pr(c|d\prime) \cdot \lambda \cdot e^{-\lambda \cdot t_{d\prime}}) \tag{3.5}$$

temp-xQuaD Analogously, we modified the probabilistic model of xQuaD. Different from IA-Select, xQuaD introduces the parameter α, to control the trade-off between relevance and diversity. The objective function of temp-xQuaD is given in Equation 3.6.

$$\begin{aligned} f_s(d) = (1-\alpha)Pr(d|q) + \alpha \sum_c Pr(c|q)Pr(c|d) \cdot \lambda \cdot e^{-\lambda \cdot t_d} \\ \prod_{d\prime \in S} (1 - Pr(c|d\prime) \cdot \lambda \cdot e^{-\lambda \cdot t_{d\prime}}) \end{aligned} \tag{3.6}$$

temp-topic-richness Differently, the subtopics in topic-richness are modeled as a set of different data sources. The objective function of topic richness model is the generalization of IA-Select and xQuaD framework. Hence, we inject the temporal factor into the model analogously to what we did with temp-IA-Select and temp-xQuaD.

3.2.3 Experiments

In this section, we first investigate the quality of the subtopic mining from multiple sources. We then evaluate the performance of our time-aware diversification models on top of the mined subtopics on different metrics.

Ranking Models

Due to the time gap between the AOL query log (March to May 2006) and Blogs08 collections (crawled from January 2008 to February 2009), we exclude the subtopics mined from AOL query log. The latent LDA subtopics mined from Blogs08 are the sole source of subtopics in this experiment. We take the top-10 words from the word probabilities of a LDA topic as an explicit representation of the subtopic. We evaluate the effectiveness of diversification models at diversifying the search results produced by Okapi BM25 retrieval model. Only English documents with content-length of more than 300 characters are accepted in the final top-100 results. For each query, we choose a studying time-point (as a simulated hitting time) based on the burst period of its query volumes derived from Google Trend (e.g., for US Open, the time-point is *June 2008* and *September 2008*).

Relevance assessments In this work, there is no existing gold standard dataset. Instead, we build our own gold standard on the Blogs08. From the top-100 documents for each of the (30) queries, we assess the subtopic-document relevance using human assessment. The relevance criteria is based on how relevant is the document to the subtopic at the simulated hitting time. Each document is given a binary relevance judgment (by two experts), as follows the same setting from TREC Diversity Track 2009 and 2011. Given this orientation, a document is assessed based on the two dimensions, *relevance* and *time*. E.g., a document written about some happening that is content relevant to the subtopic but outdated is considered irrelevant. Notice that we asked the judges to assess with regards to different hitting times (simulated by monthly granularity)[8].

Evaluation metrics To evaluate the performance of our time-aware models, we use three different metrics (i.e., α-nDCG, Precision-IA and ERR-IA) that account for both the diversity and relevance of the results. In our evaluation, all metrics are computed following the standard practice in the TREC 2009 and TREC 2011 Web track [CCS09, CCSV11]. In particular, α-nDCG is computed at $\alpha = 0.5$, in order to give equal weights to both relevance and diversity. We made a slight difference that in TREC 2009 Web track where they consider all query aspects equally important. We set the subtopics weight based on

[8] The judgment is available at: www.l3s.de/~tunguyen/ecir2014_dataset.zip

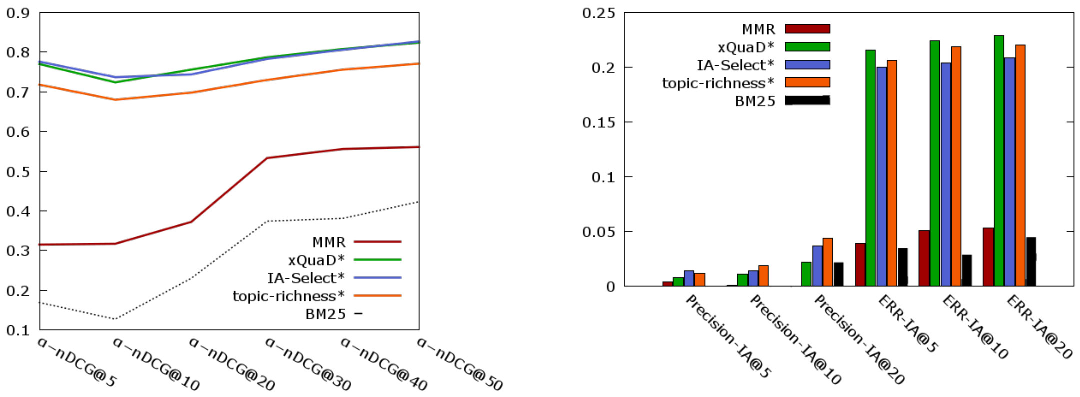


Figure 3.6: Ranking results of baseline models, * models are with dynamic subtopic mining

our dynamic subtopic measurement.

State-of-the-art model performance We measure the performance of the four state-of-the-art models: MMR, xQuaD, IA-Select and the topic richness model. The results are shown in Figures 3.6. For xQuaD, IA-Select and topic-richness, we use the mined temporal subtopics and their temporal weights as input (we skip their static methods (e.g., via Open Directory Project) since it is irrelevant in our case). We denote this change to the models with (*) symbol. We observe that measuring with α-nDCG@k, xQuaD*, IA-Select* and topic-richness model outperform MMR, while MMR shows certain increase over the baseline where there is no diversity re-ranking. We observe the same fashion when measuring with Precision-IA@k and ERR-IA@k. The results are expected since MMR does not account for subtopics when diversifying top-k result, it just tries to maximize the content gap between the top-k documents.

Time-aware parameter optimization The recency rate parameter λ is tuned to optimize the diversification models. We test λ in a wide range from 0.01 to 0.40. The parameter value with highest performance in terms of α-nDCG@k, ERR-IA@k and Precision-IA@k is chosen as best parameter value for the latter experiments. We choose k to be 10 in this set of experiments, as 10 is the common cutoff level in relevant diversity tasks [CCS09, CCSV11]. We obtained λ equals to 0.04 as the optimal value of the experiments.

Diversification performance In these experiments, we aim to evaluate our time-aware models to answer our stated research question whether taking time into account that favors recency can improve the performance of the state-of-the-art diversification models. Tables 3.5 and 3.6 represent the results of the state-of-the-art and our time-aware models for α-nDCG and the two metrics Precision-IA and ERR-IA at different cutoffs respectively. The results for α-nDCG show that temp-XQuaD significantly ($p < 0.05$) outperforms the state-of-the-art xQuaD all cut-offs (with $p < 0.01$ at k = 30). temp-xQuaD also achieves better results for Precision-IA and ERR-IA, however the results are not significant. One intuitive reason is that, different from α-nDCG that is influenced by the diversity of the top-k document result, Precision-IA and ERR-IA is more sensitive on document ranking, while

Table 3.5: α-nDCG results with $^{\triangle}$ ($p < 0.05$), $^{\triangle\triangle}$ ($p < 0.01$) indicate a significant improvement

	$\alpha-nDCG@5$	$\alpha-nDCG@10$	$\alpha-nDCG@20$	$\alpha-nDCG@30$	$\alpha-nDCG@40$	$\alpha-nDCG@50$
temp-xQuaD	**0.783**$^{\triangle}$	**0.737**$^{\triangle}$	**0.758**$^{\triangle}$	**0.805**$^{\triangle\triangle}$	**0.820**$^{\triangle}$	**0.847**$^{\triangle}$
xQuaD*	0.699	0.687	0.706	0.751	0.772	0.789
temp-IA-Select	**0.781**	**0.739**$^{\triangle\triangle}$	**0.755**$^{\triangle\triangle}$	**0.798**$^{\triangle\triangle}$	**0.822**$^{\triangle\triangle}$	**0.836**$^{\triangle}$
IA-Select*	0.738	0.698	0.718	0.760	0.790	0.807
temp-topic-richness	**0.697**	**0.662**	**0.686**$^{\triangle}$	**0.731**$^{\triangle}$	**0.753**$^{\triangle}$	**0.769**$^{\triangle}$
topic-richness*	0.654	0.638	0.660	0.702	0.727	0.741

Table 3.6: Precision-IA and ERR-IA results with $^{\triangle}$ ($p < 0.05$) indicate a significant improvement

	P-IA@5	**P-IA@10**	**P-IA@20**	**ERR-IA@5**	**ERR-IA@10**	**ERR-IA@20**
temp-xQuaD	**0.010**	0.011	**0.029**	**0.214**	**0.218**	**0.232**$^{\triangle}$
xQuaD*	0.008	0.011	0.021	0.206	0.214	0.219
temp-IA-Select	0.010	0.010	0.027	**0.207**	**0.216**	**0.235**
IA-Select*	**0.013**	**0.013**	**0.034**	0.014	0.194	0.198
temp-topic-richness	0.010	0.011	0.030	**0.191**	**0.196**	**0.201**
topic-richness*	**0.011**	**0.017**	**0.040**	0.181	0.188	0.193

we only test on the top-100 documents. The margin value can become significant when testing with top-1000 documents for the two metrics. Similar to temp-xQuaD, temp-IA-Select surpass IA-Select in overall, significantly outperforms the state-of-the-art IA-Select when measuring by α-nDCG at the cutoff k = 10, 20, 30, 40 and 50. temp-IA-Select also gives better yet not significant performance when measured by ERR-IA. However, temp-IA-Select does not surpass the original IA-Select for Precision-IA. The results of Precision-IA at cutoff k = 5, 10 and 20 show a slight decrease in performance of temp-IA-Select. We also report the results for temp-topic-richness and topic-richness in a similar fashion. Overall, our time-aware models exceed their originated state-of-the-art diversification models in most of the experimental settings. temp-xQuaD is the most consistent algorithm that outperforms xQuaD and gives better results among the six tested algorithms. On the other hand, even though surpassing the based model, temp-topic-richness gives a lower performance compared to the other two time-aware diversification models. However, the model is meant for taking subtopics from multiple sources, its performance could be enhanced if we account for other sources of subtopics (i.e., query log).

3.2.4 Related Work

Studying the temporal dynamics of subtopics has been addressed in some recent works [WZJL13, ZWJL13]. Whiting *et al.* [WZJL13] considered event-driven topics as a prominent source of high temporal variable subtopics (search intent). They proposed an approach (in the absence query log) to present query intents by sections in the Wikipedia article. They further linked the temporal variance of intents (reflected by query volumes) with the change activity of the article sections. The proposed approach has certain limitations where the temporal dynamics and complexity in content structure of a Wikipedia article (where the subtopics are mined) is left un-tapped. Zou *et al.* [ZWJL13], in another aspect, studied the effects of such subtopic temporal dynamics for the task of diversity evaluation. They conducted a small study on the Wikipedia disambiguation pages to analyze the changes in a subtopic popularity (the number of page views) over time. They concluded that such temporal dynamics impact the traditional diversity metrics for ambiguous queries, where the subtopic popularity is considered static over time. On the other hand, Berberich *et al.* [BB13] aimed to diversify search results over time, for those queries that are temporally ambiguous (i.e., the relevant time is un-known). Their proposed model, therefore, ignores the underlying intents of such queries and solely focuses on diversifying the relevant time periods of such queries. Styskin *et al.* [SRVS11] proposed a machine learning approach to identify recency-sensitive queries. Their large-scale experiments on real (recency-sensitive) queries show that promoting recent results (to the extent proportional to the query's recency level) to the result sets increases users' satisfaction.

3.2.5 Conclusion

In this section, we studied the problem of diversifying search results for temporally ambiguous, multi-faceted queries. For such queries, the popularity and relevance of their corresponding subtopics are highly time-dependent, that is, the temporal dynamics of query subtopics can be observed. We determined dynamic subtopics by analyzing two data sources (i.e., query log and a document collection), which provides interesting insights for the identified temporal subtopics. Moreover, we proposed three time-aware diversifying methods that take into account the recency aspect of subtopics for re-ranking. The experimental results show that leveraging temporal subtopics as well as recency can improve the diversification performance (diversity and relevance) and outperform the baselines significantly, for temporally ambiguous, multi-faceted queries.

4 Events and Collective Memory in Wikipedia

4.1 Remembering Events in Wikipedia

The way humans forget and remember is a fascinating area of research for both individual and collective remembering. Aspects such as the constructiveness of memories are challenging our intuitive understanding. While forgetting enables us to stay focused and cope with the multitude of our daily experiences, the way past memories are triggered by new experiences is sometimes surprising.

The basis for developing an effective technology that can complement processes in human memory is a deep understanding on how humans remember and forget. Due to its importance for societal processes, it is also important to consider remembering as a crowd phenomenon and investigate what is remembered by communities and societies, e.g., about past events. This is related to the concept of collective memory introduced by Halbwachs [Halon]. Collective memory is a socially constructed, common image of the past of a community, which frames its understanding and actions. At the same time, collective memory is not static; it is determined by the concerns of the present [Halon]. With the social Web, the construction and dynamics of collective memory have become an observable phenomenon, which promises new insights. We are especially interested in systematically investigating what triggers (or revives) the memory of past events. Knowledge about such triggering behavior can be used both for recommending related events that are probably remembered by the user, e.g., for enriching a news report about an event and for surprising the reader by reminding the reader of related events she/he (most probably) has forgotten, thus introducing some serendipity.

Web 2.0 offers new rich data sources for a large scale analysis of pattern in human and especially collective remembering and forgetting, which complements qualitative studies from cognitive psychology. One important source for better understanding pattern of collective memory and its construction processes is Wikipedia [FM12, Pen09]. The social negotiation and construction processes for example, is reflected by early editing activities on pages referring to events [FM12, KGC13] as well as by discussions on the talk pages

[Pen09].

In our analysis, we investigate the triggering or reviving of memories of past events using revisiting pattern in English Wikipedia as indicators for what is collectively (actively) remembered and what is rather on the path of forgetting. The content and usage of Wikipedia articles is an important source of information about real-world events [GKK$^+$13]. In this study, we focus on exploiting view logs of Wikipedia event pages as the signals of collective memory. From a cognitive point-of-view, access or view logs may not directly reflect how people forget information, e.g., people may remember about an event, but they do not access assets associated to the event. However, we argue that significant patterns found in view activities are a good estimate of public remembering. Such a visit is typically triggered by thinking of this past event and will also refresh the memories on an event by revisiting the information on the respective page. Additionally, analyzing Wikipedia article updates faces scalability issues and this is left for future work.

Generally, individual memories are subject to a forgetting process, which is driven by some form of the forgetting curve first proposed by Ebbinghaus [Ebb85]. Especially episodic memory [Tul02], which is responsible for memorizing details of events, is subject to fast forgetting due to interference with memories of new events. Both the effect of proactive interference [Und57] and retroactive interference [MM31] make it difficult to remember event details after a while. Various factors can, however, boost human memory of a event or person from one's past, such as, similar events, anniversaries or even a scent. In general, there is a strong relationship between the capability to remember something and the frequency and recency of activating this memory [AS91]. Such triggering of memories can also be observed for more global events on a cumulative level of communities as the sum of individual remembering re-enforced by information sharing and media coverage. The *2011 nuclear catastrophe in Fukushima* did, for example, trigger the memory of the *Chernobyl event* happened 25 years before raising the Wikipedia event page views from about 9,500 views per day in the first two months of 2011 to up to more than half a million views per day at the time of the Fukushima disaster (around March 15, 2011).

In more detail, we are interested in the catalysts for such reviving of event memory. We investigate the role of time passed, the type of event, and other factors play in reviving memory. Our work extends the work of [AYJ11], who examine collective memory based on its reflection in a newspaper collection, in two directions. Firstly, we analyze the long-term dynamics of collective remembering by looking how forgetting is interrupted by memory revival. This also supplements work on the early memory construction phase in creating Wikipedia articles [FM12] by looking into long-term temporal development. Secondly, we add an extra perspective by analysing what people actually look at (in Wikipedia), complementing the news coverage perspective of [AYJ11].

Our contributions. We analyse over **5500** high-impact events from 11 different event categories, e.g., earthquakes, Atlantic hurricane, aviation accidents and incidents, and terrorist incidents. For investigating catalysts for event memory revival, we leverage the view logs of Wikipedia event pages.

Due to the unique characteristics of every single event especially of the unplanned events, it is very challenging to identify systematic pattern in the revisiting of past events. Therefore, this work just presents a first study in identifying catalysts for event memory triggering.

Using time series analysis, we consider (1) temporal correlations in peaking page visits between events, (2) a surprise score or the residual sum of squares on prediction error, and (3) the skewness of view shapes, as indicators for the capability to act as a catalyst for the memories about the past event. Furthermore, we investigate if there are also other indicators of relationships between the events (e.g., the same types or magnitude of events, same city or country, etc.), by using different features, namely, time, location and impact. To this end, we conduct extensive experiments for identifying promising features.

4.1.1 Forgetting & Memory Catalysts

Remembering and forgetting in the context of high-impact events, so called *flashbulb memories*, have been analysed in various studies [CBB10, CSHK09, HPB$^+$09] in cognitive psychology. According to a more recent definition [CBB10], flashbulb memory is "memory about an emotionally impacting event of personal and national importance, which is consequential, socially shared and rehearsed by media". It comprises an autobiographical part, which refers to remembering the personal context, in which one learned about the event and the memory about the event itself. Aspects that have been studied in [HPB$^+$09] are the details that people still remember over different periods of time (e.g., 1 week, 11 and 35 months) after the event, the confidence and consistency of their memories over time and the impact of media coverage. However, due to their qualitative nature, those studies are typically limited to a small number of events and a restricted number of users.

Social media analysis has been successfully used in different works for analysing collective attention and awareness [LGRC]. Due to their dynamics, events typically play an important role in such analysis. The transition to analysing remembering of events as a crowd phenomenon relates individual remembering to collective remembering. In social science, the concept of collective memory [AC95, Halon] is used in this context. It refers to the collectively constructed image (memory) of the past, which is shared by a community and, roughly speaking, used by the community for framing their current understanding and activities.

The Web and especially the social Web have a high impact on collective remembering [Pen09]. Due to its popularity as an information reference and the easy and long-term accessibility of information about an event, Wikipedia is a promising subject for analysing collective remembering. In addition to the access numbers, the importance that is assigned to Wikipedia as an information reference for event information is confirmed by the high level of community involvement reflected in the number of editors (19 million registered users and about 30 thousand active editors[1] in English Wikipedia), the fast reflection of new events in Wikipedia [KGC13], and the conflicts and *edit wars* that can be observed on controversial topics. Although religious and political topics are most dominant in edit

[1] Editors with more than 5 edits per month

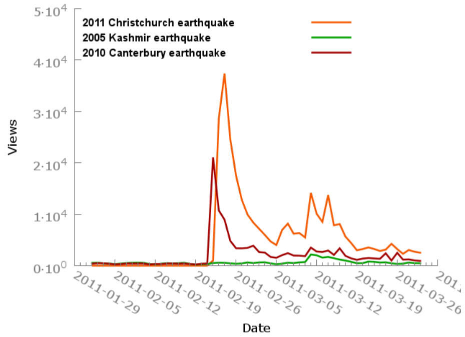


Figure 4.1: Wikipedia page views triggered by the Christchurch earthquake in February 2011.

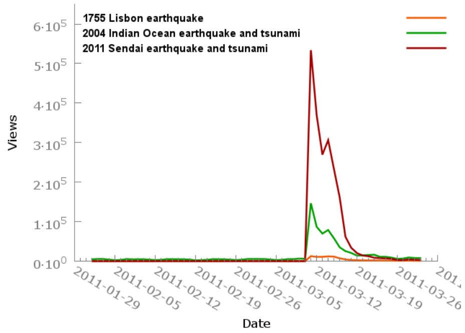


Figure 4.2: Wikipedia page views triggered by the tsunami in Japan in March 2011.

wars, there is also a considerable number of events in the top-10 lists of controversial topics extracted from Wikipedia in different languages in [YAGJ14].

Figure 4.1 shows the Wikipedia views of the event page for the earthquake in Christchurch, New Zealand in February 2011 (as a triggering event) and compares it with the view number of two other earthquakes, namely, the earthquake in Canterbury in September 2010 and the large earthquake in Kashmir in October 2005. The strong peak in the views of the Canterbury earthquake around February 2011 suggests a strong influence of the Christchurch earthquake as a catalyst for remembering the Canterbury earthquake. This strong influence can be explained by the facts that a) both earthquakes happened in the same region and b) there is a time gap of just five months between the two events. In contrast, memory for the Kashmir earthquake, which is more distant in time and location, seems to be revived to a much lesser degree by the Christchurch earthquake.

Figure 4.2 shows page views for the event page of the Japan tsunami in 2011 as the triggering event and views for the page of the Indian Ocean earthquake and tsunami in 2004.

The increasing view numbers suggest that the event in Japan acts as a catalyst for remembering the 2004 Tsunami and the earthquake in Canterbury in September 2010 does also for the event pages of both earthquakes in New Zealand when taking a closer look to Figure 4.1. Interestingly, there is an increase even for an earthquake, which lays far more in the past, like the Lisbon earthquake in 1755 shown in Figure 4.2. Of course, an increased number of Wikipedia views is only an indirect signal of memory revival for the considered event. However, we believe that a person, who visits an event page from a past event at least thinks of the event, which brings it back to active memory. Furthermore, visiting a Wikipedia event page on purpose will typically result also in reading some information about the event, such as, refreshing or extending the information memorized about the event.

4.1.2 Collective Memory

For the work in this chapter, we leverage the page view statistics of English Wikipedia in analysing the collective memory of a past event. An event e can be represented by a Wikipedia event page, with starting time e_{st} and ending time e_{et}, and it consists of other information, such as, location, impact (e.g., magnitude, fatality and the cost of damage). The time series X of e is created using the aggregated number of daily views of its corresponding Wikipedia page. In the rest of this section, we present our methodological approach for detecting the reviving of memory of past events, which helps in identifying the catalysts for such remembering.

Remembering Score

For identifying the reviving of memory of past events, we exploit remembering signals based on the event time series and three time series analysis techniques, i.e., cross-correlation coefficient, surprise detection, and skewness.

1) **Cross-correlation coefficient** (CCF) is a statistical method to estimate how variables are related at different time lags. That is, the CCF value at time t between two time series X and Y indicates the correlation of the first series with respect to the second series shifted by a time amount t, e.g., in days or weeks. A common measure for the correlation is the Pearson product-moment correlation coefficient. The CCF between two time series describes the normalized cross covariance and can be computed as:

$$CCF(X,Y) = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

where x_i and y_i are values at time t_i of X and Y, $\bar{x}$ and $\bar{y}$ are the means values, and σ_X and σ_Y are the standard deviations. In our case, the time series X and Y are corresponding to the time series of two events respectively. The CCF function has values between -1 and +1, where the value ranges from 1 for perfectly correlated results, becomes 0 when there is no correlation, decreases to -1 when the results are perfectly correlated negatively. This measurement can be interpreted as the similarity between two time series in volume, with consideration of time shifts. Hence, the CCF value reflects the remembering of a past event with respect to a studied event.

2) **Sum of squared error** (SSE) is a measure of the accuracy of short-term forecasts for time series data. It has been mentioned in [RSD+] that an unplanned event happened when there is a significant error in the residuals of its predictive model. Our intuition of employing this feature is as follows. Any two co-peaking events are not necessarily correlated with each other, e.g., a past event occurs to have an anniversary and simply peaks around the same time. We consider SSE or prediction error is a good feature for surprise detection, rather than just looking at co-peaking. In this work, we use the HoltWinters prediction model computed as the residual sum of squares on prediction errors. The score implies how *unplanned* the value of time series is at the time period of interest. Given a time series $Y = \{y_1,...,y_t\}$, a predictive model and its fitted values $V = \{v_1,...,v_t\}$, the SSE or surprise score at time t is calculated by:

$$SSE(Y,V) = \sum_{i=1}^{t}(y_i - v_i)^2$$

3) **Kurtosis** is a basic statistic method to measure the *skewness* of a distribution. Intuitively, we seek for a signal of relatedness other than co-peaking. Inspired by the work [JD07], we use this feature to capture the reviving of past events by considering how much of the probability distribution is contained in the peaks, and how much in the low-probability regions. Kurtosis is calculated as the average deviations of the time series elements to the forth power divided by the standard deviation to the forth power (refer to [JD07] for more detailed computation).

From the list of candidates for related events, we apply certain filtering criteria to leave out those events with insignificant behavior. We categorize these criteria into two distinct classes: **burstiness** (relied on burst detection) and **basic time series statistics**, i.e., *min*, *max* and *average frequency*. In burstiness filtering, we define *overlapRatio*, which is the ratio between the number of bursts occur overlapped with the event period and the total number of bursts within 14 days before/after the event. The time window can be varied. Note that, this burst entropy is an indicator of how *random* is the activeness of the event in the studied period. The other techniques in this class are derived from *burst strength* and *burst duration*. Finally, the **filtering** score is determined as: $Filter = \sum_{f \in \mathscr{F}} \phi \cdot f$, where each feature f comprises the set of filtering techniques $\mathscr{F}$, and ϕ is a mixture parameter.

We compute *remembering* scores based on three main signals, i.e., CCF, SSE and Kurtosis, for quantifying the remembering of past events. The higher the values of CCF and SSE, the better the past events are remembered. On the contrary, Kurtosis values must be low such that a smaller peak around the mean has high remembering scores. Our remembering functions are defined as: (1) a combination of the signals (denoted *a basic remembering score*), and (2) a filtering method applied to a basic remembering score (denoted *a filtered remembering score*). The basic remembering score is calculated as:

$$Remembering = \alpha \cdot CCF + \beta \cdot SSE + \gamma \cdot Kurtosis$$

where the value of each individual feature will be normalized using two methods: (1) zscore - normalize each feature by its mean/standard deviation, and (2) linear - normalize each feature by its min/max values. We report the best results obtained from both normalization techniques.

To this end, a filtered remembering score can computed in two ways: (1) mixture of *remembering* and *filtering* scores, and (2) the multiplication of the two scores.

Features for Triggered Remembering

Temporal Similarity We compute the temporal similarity between two events by taking into account a time distance. We adopt the *TSU* metric, proposed in [KN10], that measures the similarity between a temporal query and a document based on their temporal metadata, i.e., temporal expression(s) in the given query and document timestamps. *TSU* copes with the time uncertainty related to the two entities by relying on a decay function. In our case, we consider the time period of an event; thus $TSU(e_i, e_j)$ between two events e_i and e_j can be computed as follows:

$$TSU(e_i, e_j) = \frac{1}{2} \times \left(DecayRate^{\lambda \cdot \frac{|st_i - st_j|}{\mu}} + DecayRate^{\lambda \cdot \frac{|et_i - et_j|}{\mu}} \right)$$

where $DecayRate$ and λ are constants, $0 < DecayRate < 1$ and $\lambda > 0$. *st* and *et* are the starting time and ending time of the event *e*. μ is the unit of time distance between the two events, e.g., one week or 3 months. In addition to *TSU*, we also use the *time difference* of two events, i.e., an absolute distance in days, months, or years, as our temporal features.

Location Similarity We extracted locations where events occurred, as described in its corresponding Wikipedia article, from our annotated dataset. Similar to [SGJ11], we create a geographic hierarchy of event locations as follows: *city* → *state* → *country* → *neighbor countries* → *continent*. Because our event dataset consists of mostly *high impact* ones, we consider *city* as our finest granularity, and focus more on the *country* level. For instance, if a flood event happened in Thailand (e.g., floods in Thailand 2011), events that took place in the nearby region (floods in Vietnam 2008) will be accounted for our location similarity metric. To obtain higher quality of location expressions (from our annotated dataset, which contains Wikipedia text snippets), we further use Stanford NER[2] for geographical entity extraction. Such location expressions are often short and missing information, we *fully* disambiguate and enrich them (to cover all upper levels) by looking up into data dumps provided by GeoNames[3] (e.g., Chicago ▶ [Chicago (city), Illinois (state), United State (country), Canada, Mexico, Cuba (neighbor countries), North America (continent)]. We define a location similarity metric (based on the Jaccard similarity coefficient) by assigning geographical *weights* to elements in the set. According to our geographic hierarchy, we give higher weights to the lower levels and lower weights to the upper levels. In detail, we give weights in the scale from 1 to 4, where 4 is assigned to *city*, 3 to *state*, 2 to *country*

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

[3] http://www.geonames.org/export/

and 1 to *neighbor countries* and *continent*. The location similarity score is given by the weighted Jaccard similarity between two enriched-location sets.

Impact of Events The impact score of an event is measured based on the impact it causes in different aspects. The aspects include *damaged area*, *damaged properties*, *the cost of damage*, *magnitude* (for earthquake events), *highest winds*, *lowest pressure* (for Atlantic hurricanes) and *fatalities*. The information derived from each aspects are quantified, normalized and consequently accumulated in the final results. We categorize the aspects into two different types: *fatalities* (number of deaths) and *other* (the rest of the aspect), as our empirical studies shown that fatalities often induce people's remembering.

4.1.3 Experiment

Experimental Settings

We analysed over **5,500** high-impact events from 11 different event categories depicted in Table 4.1. For computing the temporal similarity, we experimented with 6 different time granularities, i.e., 1 day, 7 days, 1 month, 6 months, 1 year and 10 years, where $DecayRate = 0.5$, $\lambda = 0.5$ and μ is varied according to different granularities defined above. We employed the open source implementation of burst detection by CISHELL[4]) using its default parameters. For the aggregation methods, we set the parameters for *remembering* scores as α is 0.5, β is 0.4 and γ is 0.1, where these values were empirically determined. When applying our filtering technique, we weighed each filtering feature equally, i.e., giving 0.2 to the mixture parameter ϕ for all features. The time series parameters: a time window in days w, lags in days l, and a smoothing sm, are $w \in \{7, 14\}$, $l \in \{3, 7\}$, and $sm = 1$ respectively and these parameters will be studied for their performance in the experiments.

Metrics. In our experiment, we measured the association between two measured quantities *remembering* scores and the proposed catalyst features, i.e., temporal similarity and location-based similarity using different correlation coefficients: Pearson's, Spearman's, and Kendall's. The first coefficient is a measure of linear correlation, whereas the latter two metrics measure rank correlation statistics. Correlation coefficient measures the statistical correlation between two variables, which ranges from 1 for perfectly correlated results, through 0 when there is no correlation, to -1 when the results are perfectly correlated negatively. As observed empirically, Spearman's rank correlation coefficient provides better results than the other two metrics. Hence, we will only report the results for this correlation metric.

Experimental Results

The expectation in analysing the triggering of related events was that there is a clear correlation with the type of events, time and location. Roughly speaking, when event *e* happened,

[4] http://wiki.cns.iu.edu/display/CISHELL/Burst+Detection

Table 4.1: Statistics of event categories with time spanning from earliest dates to *October 2013*.

Category	#Events	#Triggers	Earliest Date
Atlantic hurricane	654	134	1900-08-27
Aviation accidents	787	146	1912-05-13
Civil wars	*78*	*7*	*793*
Earthquakes	468	119	426 BC
Floods	*114*	*78*	*1897-04-01*
Mass murder	1136	344	1897-05-27
Pacific typhoon	*253*	*68*	*1944-12-17*
Terrorist incidents	727	295	1950-04-01
Tsunamis	*49*	*5*	*1700-01-26*
Volcanic events	*11*	*7*	*1815-01-01*
Wildfires	74	44	1970-09-26
Total	**4351**	**1247**	

people would remember events that are of the same type as *e*, happened nearby and/or in the recent past of *e*. Our analysis has, however, shown that there are no such clear pattern. This is partly due to a variety of factors including the unique characteristic of each event, the uneven distribution of unplanned events in space and time, and the dominating influence of very large events. Therefore, our experiments just give first insights into how event memory is triggered as part of collective memory.

One of the factors to be considered is the number of events available in individual categories. Table 4.1 shows the data set statistics of each event category used in our study. We performed our experiments for the listed 11 event categories. For five of the categories - shown in italics in the table - the number of triggering events as well as the number of events that could be triggered was too low for making any reliable statements. Therefore, we only present results for the remaining six categories here.

For better understanding the impact of the temporal and spatial distribution of the events in the individual categories on event memory triggering, we first take a look on this distribution. For some of the considered event categories, Figure 4.3 shows the number of events in each year (for the last 100 years) and the distribution of the most frequent locations. The temporal distribution shows in all cases a strong focus on more recent events. This is at least partially due to the development of Wikipedia, which leads to a better coverage of events with the increasing popularity of Wikipedia. For older events, we can observe the typical pattern in collective memory that events with higher impact are better remembered in the collective memory and thus also more readily a-posteriori documented in Wikipedia.

For **Aviation accidents**, for example, there is a high variation in the numbers for individual years, highlighting the random character of those events. The numbers are more evenly distributed for the event categories **Earthquakes** and **Atlantic hurricane**. For the spatial distribution, we see a more or less random distribution for **Aviation accidents** with majority of events in the location group *Others*. In contrast, other events categories, such as, **Earthquakes** and **Terrorist incidents** show the focus on the typical critical regions. For **Wildfires** (not shown) this effect is even stronger. We can see a clear focus on US and Australia (together 79% of all the events). Similarly, **Atlantic hurricane** show a focus on

Table 4.2: Filtering by the maximum number of daily view effects on the ranking of top remembered events for the triggering event Hurricane Sandy.

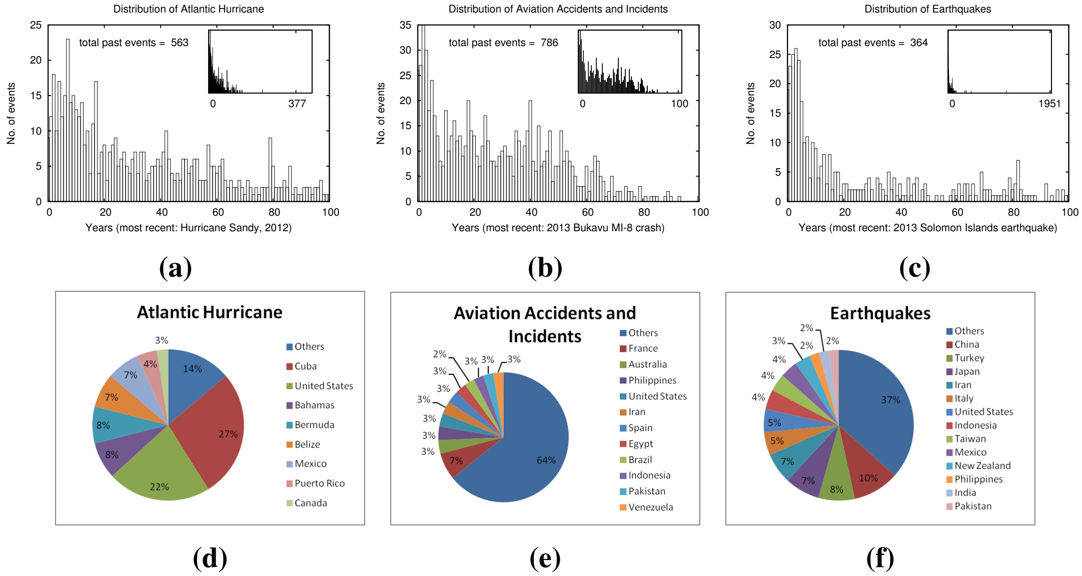
event	max > 100		event	max > 500		event	max > 1000	
	#view	remember		#view	remember		#view	remember
Hurricane Katrina	106551	0.84	**Hurricane Katrina**	106551	0.84	**Hurricane Katrina**	106551	0.81
1991 Perfect Storm	71092	0.54	**1991 Perfect Storm**	71092	0.52	**1991 Perfect Storm**	71092	0.51
Great Hurricane of 1780	11492	0.47	**Great Hurricane of 1780**	11492	0.47	**Great Hurricane of 1780**	11492	0.45
Hurricane Inez	220	0.45	**Hurricane Donna**	8565	0.43	**Hurricane Donna**	8565	0.42
1856 Last Island hurricane	143	0.45	**Hurricane Mitch**	10026	0.43	**Hurricane Mitch**	10026	0.40
Hurricane Donna	8565	0.44	**Hurricane Frederic**	804	0.43	**Hurricane Juan**	1443	0.39
Hurricane Mitch	10026	0.44	**Hurricane Georges**	1571	0.43	**Hurricane Georges**	1571	0.39
Hurricane Isaac (2000)	439	0.44	**Hurricane Charley** (1986)	620	0.42	**Hurricane Andrew**	28511	0.39
Hurricane Nicole (1998)	150	0.44	**Hurricane Gustav**	1576	0.41	**Hurricane Gustav**	1576	0.38
Hurricane Frederic	804	0.43	**Hurricane Alicia**	1030	0.41	**Hurricane Alicia**	1030	0.37
Hurricane Claudette (2003)	174	0.43	**Hurricane Juan**	1443	0.41	**Hurricane Gilbert**	4351	0.37
Hurricane Georges	1571	0.43	**Hurricane Lili**	513	0.41	**Hurricane Isaac (2012)**	11351	0.37
Hurricane Hilda	146	0.43	**Hurricane Andrew**	28511	0.40	**Hurricane Wilma**	17496	0.36
Hurricane Omar (2008)	169	0.43	**Hurricane Faith**	748	0.40	**Hurricane Frances**	1708	0.36
Hurricane Bret (1999)	145	0.43	**Hurricane Alex (2010)**	689	0.40	**Hurricane Hugo**	9655	0.36

the typical hurricane regions.

As described in Section 4.1.2, we conducted several intuitive filtering methods to exclude trivial events from the remembering analysis. To have a clearer view of the possible effects of these filtering on the remembering score, we give an example of one of the chosen methods. Table 4.2 shows the top remembered events which are triggered by *Hurricane Sandy* event, filtered by the maximum number of views per day. The results are three different ranked lists based on three manually defined thresholds: at least 100, 500 and 1000 as maximum number of views, respectively. *Hurricane Inez* and *1856 Last Island hurricane* which appear high in the first list, are left out when the threshold is increased (from left to right). These events have high remembering score, yet seem to be not publicly attentive (the number of daily views never gets over 200). More interesting events (e.g., *Hurricane Andrew*) are boosted higher in the latter lists.

To address how impact features (i.e., location, time and size) exert influence on collective memory, we present a qualitative analysis on several spotlights from 11 studied categories. The following case studies describe a close examination of some of the experimental results on triggering events throughout different categories. Figure 4.4 depicts the distribution of Atlantic hurricane events triggered by *Hurricane Sandy (2012)*, and *Hurricane Hanna (2008)*, from top to bottom respectively, with regard to three dimensions: location, time and remembering score. In general, location and time contribute low effect on remembering scores for events in this category. However, the events with significantly peaked scores have clear location similarities with the triggering event, and their happening time is close to the time of the trigger. Figure 4.5 holds up this claim by delineating top-10 events triggered by the two events. *Hurricane Gustav* is the freshest hurricane toward *Hurricane Hanna* and struck at around the area of Puerto Rico and East Coast of the US. By contrast, *Hurricane Sandy* commemorates old hurricanes decades ago, but location is still a strong indication of the remembering. One interesting finding is that both *Hurricane Sandy* and its triggered *1991 Perfect Storm* were initially formed around Canada areas. Note that, the mentioned events are high-impact (most destructive and costly).

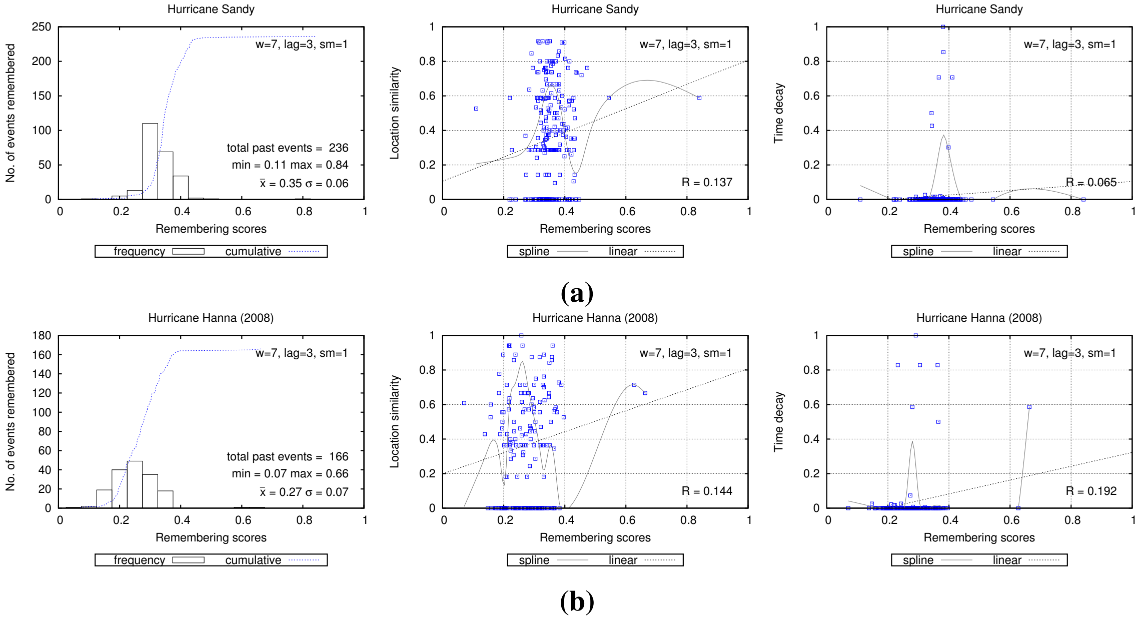
The second category of events that we want to take closer look to are **Aviation acci-**



(a) (b) (c)

(d) (e) (f)

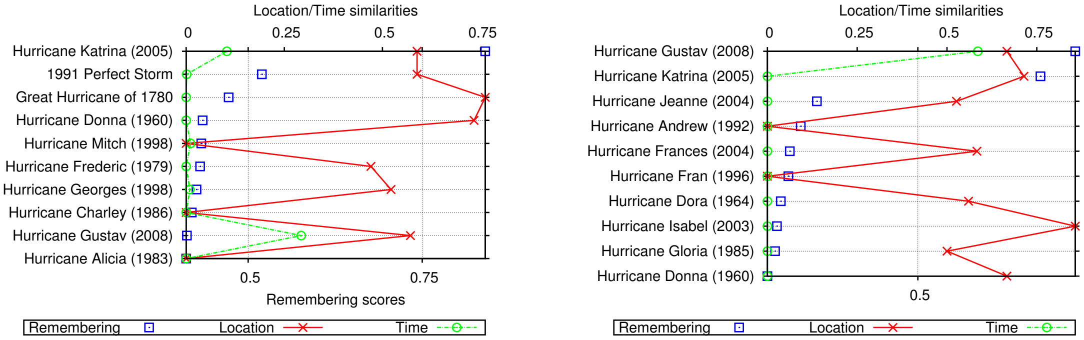
Figure 4.3: Distributions of highlighted category events over two dimensions: time and location (top to bottom).



(a)

(b)

Figure 4.4: Results for Hurricane Sandy, 2012 (top) and Hurricane Hanna, 2008 (bottom): (left to right) Distribution of remembering scores, the correlation of remembering scores vs. location and time. Spline lines approximate the interpolation of data points weighted by 3.0.



(a) **(b)**

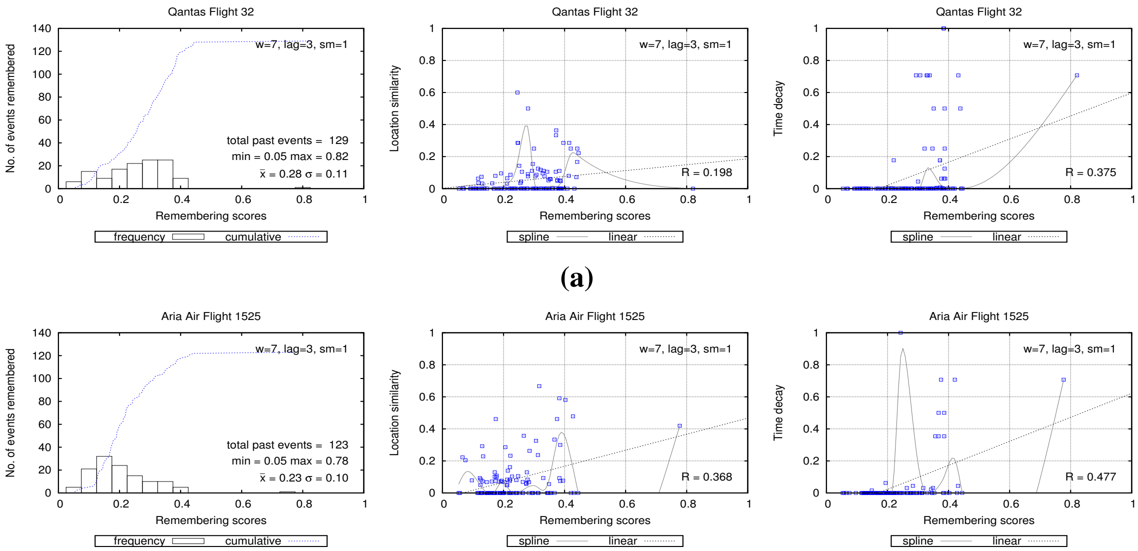
Figure 4.5: Lists of top-10 past events triggered by remembering of Hurricane Sandy (left) and Hurricane Hanna (right).

dents. The *Qantas Flight 32* accident in 2010, for which the top-10 list of remembered events is shown on the top of Figure 4.7, is a good example on the mix of impact factors that trigger the past remembering. On the first glance there seems to be no clear pattern besides that nearby events are better remembered[5]. A closer look on the individual candidates shows that some of the accidents are probably remembered, since they happened recently before the observed accident, such as, *Aero Caribbean Flight 883 (2010)*, *Qantas Flight 30 (2008)*, which is also the same airline, and *Air France Flight 447 (2009)*. Others are probably remembered because they happened spatially close-by, such as, *Japan Airline Flight 123 (1985)*. *Air France Flight 4590 (2000)* is neither temporally nor spatially very close. However, this was the Concorde accident, which had in general very high visibility and is, thus, also very strongly remembered. It also appears in the top-10 list of the second flight analysed in Figure 4.7 (appeared bottom). A similar situation occurs for the *Tenerife Airport disaster (1977)*, which is classified as “deadliest accident in aviation history” in Wikipedia, because two aircraft collided. A similar pattern can be observed for the second analysed flight *Aria Air Flight 1525*, an Iranian flight accident in 2009. Although in this case, the impact of location seems to be stronger.

For the event category **Earthquakes**, we want to discuss here an interesting series of events in more detail. These are the *2010 Canterbury earthquake*, the *2011 Christchurch earthquake* and the *June 2011 Christchurch earthquake*. All three of the earthquakes took place close to each other partially affecting the same city, i.e., Christchurch. Furthermore, they also happened to take place in the same time frame, with just some months between the individual events. If we consider the events independent from each other we might expect a very similar set of triggered event memories for all three of them. Figure 4.8, however, shows a different picture. Of course, the later events have the predecessor event(s) in their top ranked list (close in time and place).

For the first event in the series, the *2010 Canterbury earthquake*, the top ranked events

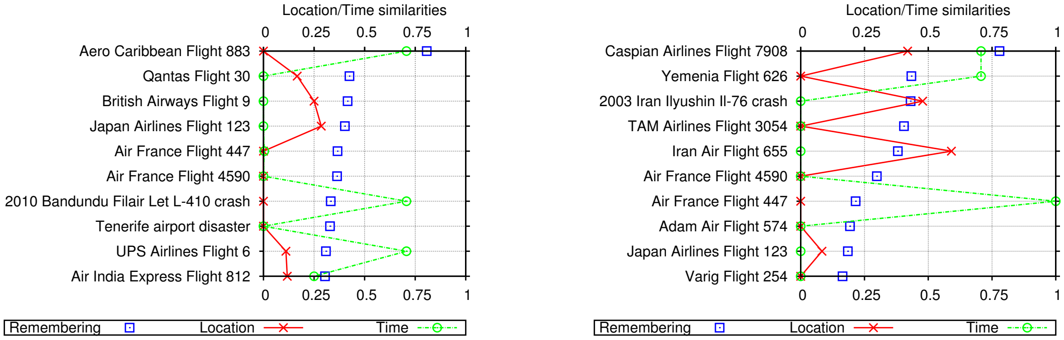
[5] This impact is even stronger than it seems, since *Qantas Flight 30* and *British Airways Flight 9* both were on the way to Australia, when the accident happened.



(a)

(b)

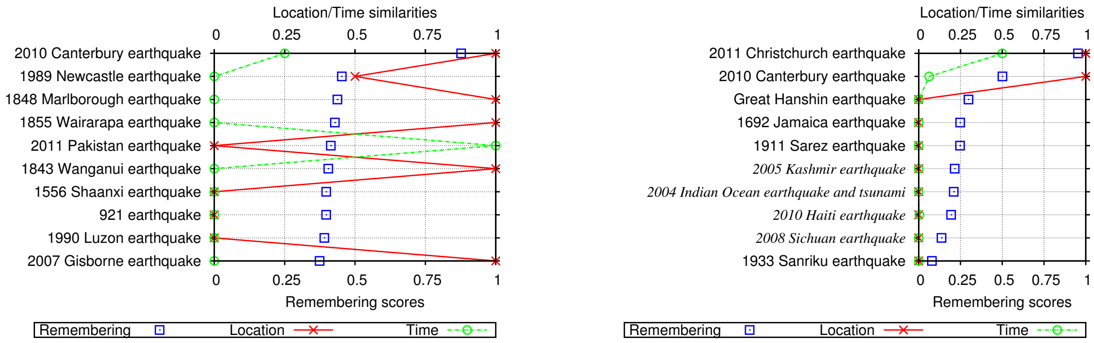
Figure 4.6: Results for Qantas Flight 32 (top) and Aria Air Flight 1525 (bottom): (left to right) Distribution of remembering scores, the correlation of remembering scores vs. location and time. Spline lines approximate the interpolation of data points weighted by 3.0.



(a)

(b)

Figure 4.7: Lists of top-10 past events triggered by remembering of Qantas Flight 32 (top) and Aria Air Flight 1525 (bottom).



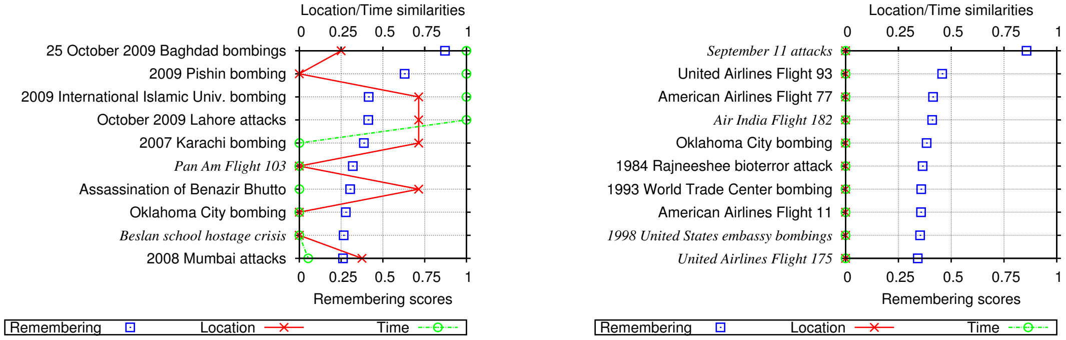
(a) **(b)**

Figure 4.8: Lists of top-10 past events triggered by remembering of 2011 Christchurch earthquake (top) and June 2011 Christchurch earthquake (bottom).

are a recent high-impact event (*2010 Haiti earthquake*) and two close-by events. The rest of the list are high impact and historical earthquakes. In contrast, the Christchurch earthquake in February shows a much stronger locality focus. For this second event in the series, people seem to be interested much more in the previous events in the same region. Recent events and high-impact events outside the region are in the minority in the top-10 list. For the third event, the remembered events are dominated by the two predecessor events. The remembering score drops very quickly after those two events. Within the other remembered events there are mainly historical events and previous high-impact events, of which only one, the *2010 Haiti earthquake* is recent.

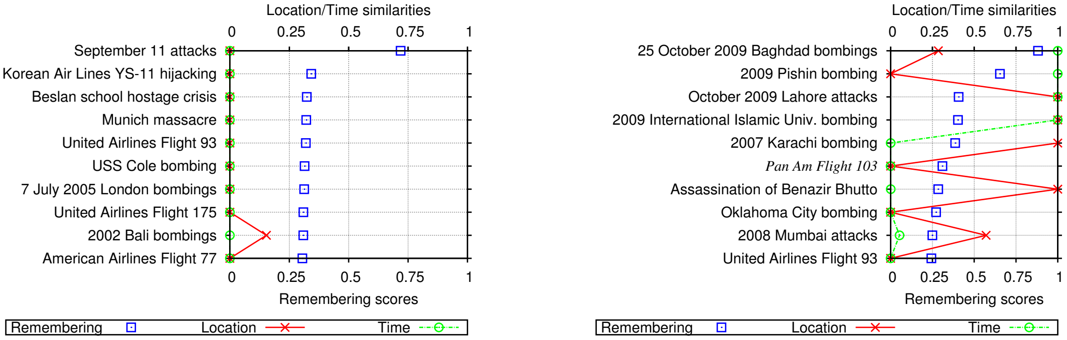
In summary, the results suggest that recent events in the same region are good candidates to be remembered. In addition to location and high-impact of earthquakes, recent events that are not close-by do not play a very important role in event memory triggering. Rather there is an interest in high impact events from the past including historical earthquakes. This pattern can also be seen for other earthquakes, which we analysed as triggering events. However, the results also suggest that, for a full analysis, it might also be necessary to look beyond single events, especially, if there are several events in temporal and local proximity.

The last category of events considered here in more detail are **Terrorist incidents**. We have selected four spotlight events from this category for discussion: the *2009 UN guest house attack in Kabul*, the *19 September 2010 Baghdad bombings*, the *9 September 2012 Iraq attacks* and the *28 October Peshawar bombing*. The first and the last event in the list show a quite similar characteristic with the airplane accidents, where the top-10 list is a mix of events very close in time to the triggering event, in near-by places or a generally remembered high-impact event, such as, the *Pan Am Flight 103* event (Lockerbie bombing). The top-10 lists for the second and the third event are however rather surprising, because the remembered events in the lists are neither temporally nor spatially related to the triggering events. For the *2010 Baghdad bombing*, nearly all triggered events are in the US, the top ranked events being related to the *September 11 attacks*. This might be the



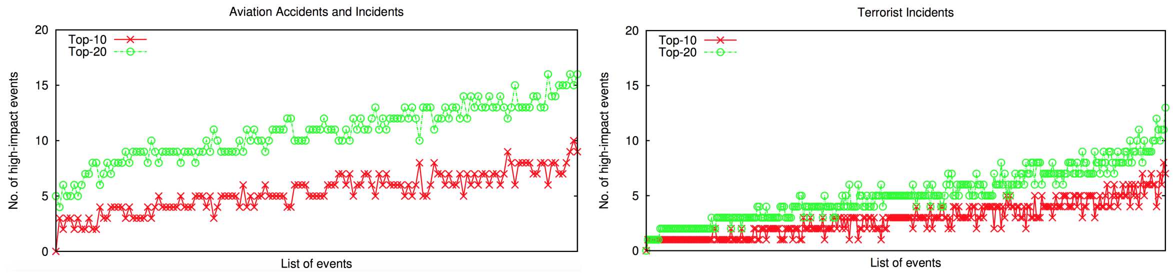
(a) **(b)**

Figure 4.9: Lists of top-10 past events triggered by remembering of 2009 UN guest house attack in Kabul (top) and 19 September 2010 Baghdad bombings (bottom).



(a) **(b)**

Figure 4.10: Lists of top-10 past events triggered by remembering of 9 September 2012 Iraq attacks (top) and 28 October 2009 Peshawar bombing (bottom).



(a) (b)

Figure 4.11: Distributions of high-impact events in the top-10 and top-20 triggered events ranked by their remembering scores for two categories.

effect of the cultural bias of English Wikipedia that we are using as the basis: Terrorism events happening in Iraq are linked to terrorist events in US rather than to other bombings in Iraq, since the US events are possibly stronger in the collective memory of the typical use of the English Wikipedia. The third event, the *9 September Iraq attacks* exhibits a similar pattern again with a strong impact on remembering of the *September 11 attacks*.

Another interesting observation is that **semantic similarity** between events beyond a shared category play a quite important role for event triggering in the terrorist attack. For example, the event *June 2012 Kaduna church bombings* (not depicted) triggers the remembering of other religion related terror attacks, such as, the *Sarin gas attack* on the Tokyo subway (2nd ranked), the *Grand Mosque Seizure* (5th ranked), and the *16th Street Baptist Church bombing* (24th ranked). As another example, the *2008 Mumbai attacks* trigger the memory of other "terror attacks in business, entertainment or hotel areas", e.g., the *Islamabad Marriott Hotel bombing* (2nd ranked), the *2002 Bali bombings* (7th ranked), and the *Moscow theater hostage crisis* (15th ranked). This finding suggests that besides location and time, semantic similarity between events also influences, which events are remembered. The more fine granular classification of events might provide additional factors, although it is crucial to find similar subclasses to those that the human brain uses to associate events with each other.

Influence of High-impact Events. As we have already seen in the previous discussions, high-impact events tend to be remembered although they are not strongly correlated in time or place with the considered event. Therefore, we investigated how the event impact or size influence the remembering of past events. Our assumption is that events with a high impact, for example in magnitudes or perceived damages in terms of casualties or costs are better remembered. For this purpose, we analyse the number of high impact events, which appear in top-10 and top-20 ranked lists of selected events across two highlighted categories, namely, **Aviation accidents** (rather high density), and **Terrorist incidents** (rather low density) ; see Figure 4.11. For assessing the impact of the events we use the reported number of fatalities and a cut-off threshold. Based on the available data, we chose the thresholds to be 30 and 100 for the two categories. It can be seen that the number of high-impact events within top-10 lists ranks mostly between 25% and 50% of the list elements.

Within the top-20 lists the percentage of high-impact events is higher, ranging mostly between 25% and 75%. However, there is a difference between the individual event categories with a very high number of high-impact events in the top-20 events of aviation accidents, which might be due to the generally low number of aviation accidents.

These results support the findings of the discussions below, that the impact of events besides closeness in time and space, is an important factor for remembering events.

4.1.4 Related Work

So far little work has been done on analysing Wikipedia as a *global memory place* [Pen09]. Most of this work on Wikipedia and collective memory [FM12, KGC12, KGC13] focuses on the early phase of capturing the events in Wikipedia, which is characterized by negotiation and sense making processes. In [KGC12, KGC13], for example, the collaborative creation of breaking news articles is analysed from a behavioral science perspective, and in [FM12] the use of language pattern and inferred psychological processes in documenting disasters. We use Wikipedia to analyse memory reviving pattern of collective memory. Ciglan and Nørvåg [CN10] proposed to detect events by analysing trends in page view statistics. In their recent work, Georgescu et al. [GKK+13] extracted event-related information from Wikipedia updates for a given entity based on burst detection, temporal information, and textual content.

With respect to collective memory, the work by Au Yeung and Jatowt presented in [AYJ11] is most related to our approach. There, the authors analysed references to the past (as an indicator to what is remembered) in a large news collection from different countries for identifying, which years are most frequently referenced. Furthermore, they exploit topic modeling and conditional probability computations over the topic and year reference distributions to identify referenced past topics and reference triggering topics. Our approach differs in two main aspects. Firstly, we rely on actual information access instead of news references for identifying, what is remembered. Secondly, we perform a more systematic analysis of catalysts for triggering memory, which goes beyond the analysis of what is actually remembered in [AYJ11].

Our approach is also related to work on analysing peaks of collective attention in other Social Media such as Twitter [AHSW11, LGRC] in the goals and applied methods. In [AHSW11], they analysed trending topics looking into factors for attracting collective attention and its decay finding a strong influence of exogenous factors. Another study on Twitter [LGRC] event-related peaks in Twitter are analysed relating their content to temporal activity profiles, especially clustering the fraction of tweets before, during and after the peak.

4.1.5 Conclusions and Future Work

In this section, we studied catalysts for revisiting memories of past events based on Wikipedia view logs. The purpose of this analysis was an improved understanding of collective memory as it is collaboratively constructed in Wikipedia. Identifying pattern of past memory triggering has proven to be more complicated than expected due to the noise and multitude of signals in view logs, due to the multitude of event types in Wikipedia, due to the unique characteristics of every single event and due to the multitude of possible reasons for revisiting a page of a past event.

In spite of this, we managed to identify some first pattern for event memory triggering for diverse event types including natural and manmade disasters as well as accidents and terrorism. For doing this we have combined correlation detection, analysis of the *surprise* aspect (unexpected change) in the distribution of the past event surrounding the peak time of the triggering event and analysis of the skewness of the distribution of the past event at the peak time of the triggering event. Our analysis confirmed the influence of closeness in time and location, but also has shown that these aspects cannot be considered in isolation and that high-impact events and semantic similarity of events also influences, which event memories are triggered by an event.

Since not so much work has been done in this area so far there is clearly still a need for further research. In our future work, we plan to deepen our systematic analysis of factors for revisiting past events and of the combination of those factors. We also plan to consider more features for the identification of memory catalysts and to verify the predictive qualities of such features in larger experiments. Furthermore, we also plan to investigate external factors for observed memory *revivals* such as media coverage linking new events to past events or reflection of such relationships in other types of social media and how to combine them with our Wikipedia-based analysis.

4.2 Dynamic Entity Relatedness Ranking

Measuring semantic relatedness between entities is an inherent component in many text mining applications. In search and recommendation, the ability to suggest most related entities to the entity-bearing query has become a standard feature of popular Web search engines [BCMT13]. In natural language processing, entity relatedness is an important factor for various tasks, such as entity linking [HSN^{+}12] or word sense disambiguation [MRN14].

However, prior work on semantic relatedness often neglects the time dimension and consider entities and their relationships as static. In practice, many entities are highly ephemeral [JLG^{+}16], and users seeking information related to those entities would like to see fresh information. For example, users looking up the entity Taylor Lautner during 2008–2012 might want to be recommended with entities such as The Twilight Saga, due to Lautner's well-known performance in the film series; however the same query in August 2016 should be served with entities related to his appearances in more recent films such as "Scream Queens", "Run the Tide". In addition, much of previous work resorts to deriving semantic relatedness from co-occurrence -based computations or heuristic functions without direct optimization to the final goal. We believe that desirable framework should see entity semantic relatedness as not separate but an integral part of the process, for instance in a supervised manner.

In this section, we address the problem of ***entity relatedness ranking***, that is, designing the semantic relatedness models that are optimized for ranking systems such as top-k entity retrieval or recommendation. In this setting, the goal is not to quantify the semantic relatedness between two entities based on their occurrences in the data, but to optimize the partial order of the related entities in the top positions. This problem differs from traditional entity ranking [KYZ+15] in that the entity rankings are driven by user queries and are optimized to their (ad-hoc) information needs, while entity relatedness ranking also aims to uncover the meanings of the the relatedness from the data. In other words, while conventional entity semantic relatedness learns from data (editors or content providers' perspectives), and entity ranking learns from the user's perspective, the entity relatedness ranking takes the trade-off between these views. Such a hybrid approach can benefit applications such as exploratory entity search [MBL15], where users have a specific goal in mind, but at the same time are opened to other related entities.

We also tackle the issue of *dynamic ranking* and design the supervised-learning model that takes into account the temporal contexts of entities, and proposes to leverage *collective attention* from public sources. As an illustration, when one looks into the Wikipedia page of Taylor Lautner, each navigation to other Wikipedia pages indicates the user interest in the corresponding target entity given her initial interest in Lautner. Collectively, the navigation traffic observed over time is a good proxy to the shift of public attention to the entity (Figure 4.12).

In addition, while previous work mainly focuses on one aspect of the entities such as textual profiles or linking graphs , we propose a *trio neural* model that learns the low level representations of entities from three different aspects: Content, structures and time aspects. For the time aspect, we propose a convolutional model to *embed* and *attend* to local patterns of the past temporal signals in the Euclidean space. Experiments show that our trio model outperforms traditional approaches in ranking correlation and recommendation tasks. Our contributions are summarized as follows.

- We present the first study of dynamic entity relatedness ranking using collective attention.
- We introduce an attention-based convolutional neural networks (CNN) to capture the temporal signals of an entity.
- We propose a joint framework to incorporate multiple views of the entities, both from content provider and from user's perspectives, for entity relatedness ranking.

4.2.1 Related Work

Entity Relatedness and Recommendation

Most of existing semantic relatedness measures (e.g. derived from Wikipedia) can be divided into the following two major types: (1) *text*-based, (2) *graph*-based. For the first,

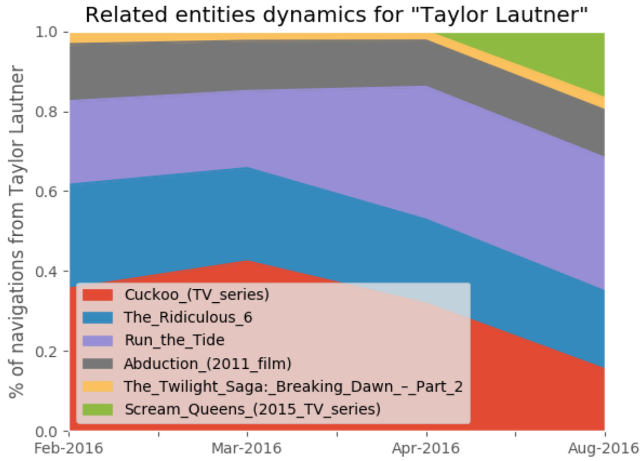


Figure 4.12: The dynamics of collective attention for related entities of Taylor Lautner in 2016.

traditional methods mainly focus on a high-dimensional semantic space based on occurrences of words ([GM07, GM09]) or concepts ([AB14]). In recent years, embedding methods that learn low-dimensional word representations have been proposed. [HHD+15] leverages entity embedding on knowledge graphs to better learn the distributional semantics. [NXC+16] use an adapted version of Word2Vec, where each entity in a Wikipedia page is considered as a term. For the *graph*-based approaches, these measures usually take advantage of the hyperlink structure of entity graph [WM08, GB14]. Recent graph embedding techniques (e.g., DeepWalk [PARS14]) have not been directly used for entity relatedness in Wikipedia, yet its performance is studied and shown very competitive in recent related work [ZLS15, PFC17].

Entity relatedness is also studied in connection with the entity recommendation task. The Spark [BCMT13] system firstly introduced the task for Web search, [YMHH14, ZYL+16] exploit user click logs and entity pane logs for global and personalized entity recommendation. However, these approaches are optimized to user information needs, and also does not target the *global* and *temporal* dimension. Recently, [ZRZ16, TTN] proposed time-aware *probabilistic* approaches that combine 'static' entity relatedness with temporal factors from different sources. [NKN18b] studied the task of time-aware ranking for entity aspects and propose an ensemble model to address the sub-features competing problem.

Neural Network Models

Neural Ranking. Deep neural ranking among IR and NLP can be generally divided into two groups: representation-focused and interaction-focused models. The *representation-focused* approach [HHG+13] independently learns a representation for each ranking element (e.g., query and document) and then employ a similarity function. On the other hand, the *interaction-focused* models are designed based on the early interactions between the ranking pairs as the input of network. For instance, [LL13, GFAC16] build interactions (i.e., local matching signals) between two pieces of text and trains a feed-forward network for computing the matching score. This enables the model to capture various interactions between ranking elements, while with former, the model has only the chance of isolated observation of input elements.

Attention networks. In recent years, attention-based NN architectures, which learn to focus their "attention" to specific parts of the input, have shown promising results on various NLP tasks. For most cases, attentions are applied on *sequential models* to capture *global* context [LPM15]. An attention mechanism often relies on a *context* vector that facilitates outputting a "summary" over all (deterministic soft) or a sample (stochastic hard) of input states. Recent work proposed a CNN with attention-based framework to model *local* context representations of textual pairs [YSXZ16], or to combine with LSTM to model *time-series* data [OR16, LGA17] for classification and trend prediction tasks.

4.2.2 Problem

Preliminaries

We denote as named entities any real-world objects registered in a database. Each entity has a textual document (e.g. content of a home page), and a sequence of references to other entities (e.g., obtained from semantic annotations), called the entity *link profile*. All link profiles constitute an entity linking graph. In addition, two types of information are included to form the entity collective attention.

Temporal signals. Each entity can be associated with a number of properties such as view counts, content edits, etc. Given an entity e and a time point n, given D properties, the temporal signals set, in the form of a (univariate or multivariate) *time series* $X \in \mathbf{R}^{D \times T}$ consists of T real-valued vector $x_{n-T}, \cdots, x_{n-1}$, where $x_t \in \mathbf{R}^D$ captures the past signals of e at time point t.

Entity Navigation. In many systems, the user navigation between two entities is captured, e.g., search engines can log the total click-through of documents of the target entity presented in search results of a query involving the source entity. Following learning to rank approaches [KYZ$^+$15], we use this information as the ground truth in our supervised models. Given two entities e_1, e_2, the navigation signal from e_1 to e_2 at time point t is denoted by $y^t_{\{e_1,e_2\}}$.

Problem Definition

In our setting, it is not required to have a pre-defined, static function quantifying the semantic relatedness between two entities. Instead, it can capture a family of functions F where the prior distribution relies on time parameter. We formalize the concepts below.

Dynamic Entity Relatedness between two entities e_s, e_t, where e_s is the source entity and e_t is the target entity, in a given time t, is a function (denoted by $\mathsf{f}_\mathsf{t}(e_s, e_t)$) with the following properties.

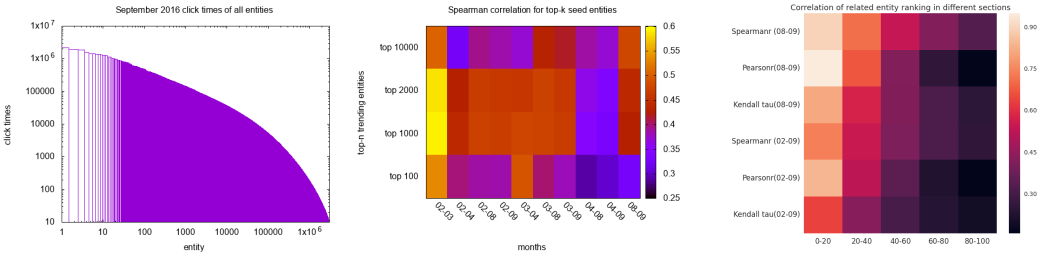
- asymmetric: $f_t(e_i, e_j) \neq f_t(e_j, e_i)$
- non-negativity: $f(e_i, e_j) \geq 0$
- indiscernibility of identicals: $e_i = e_j \rightarrow f(e_i, e_j) = 1$

Dynamic Entity Relatedness Ranking. Given a source entity e_s and time point t, rank the candidate entities e_t's by their semantic relatedness.

4.2.3 Approach Overview

Datasets and Their Dynamics

In this work we use Wikipedia data as the case study for our entity relatedness ranking problem due to its rich knowledge and dynamic nature. It is worth noting that despite experimenting on Wikipedia, our framework is universal can be applied to other sources of entity with available temporal signals and entity navigation. We use Wikipedia pages to represent entities and page views as the temporal signals (details in section 4.2.5). **Clickstream.** For entity navigation, we use the clickstream dataset generated from the Wikipedia webserver logs from February until September, 2016. These datasets contain an accumulation of transitions between two Wikipedia articles with their respective counts on a monthly basis. We study only actual pages (e.g. excluding disambiguation or redirects). In the following, we provide the first analysis of the clickstream data to gain insights into the temporal dynamics of the entity collective attention in Wikipedia.



(a) Click times distribution **(b)** Correlation of top-k entities **(c)** Correlation by # of navigations

Figure 4.13: Click (navigation) times distribution and ranking correlation of entities in September 2016.

Figure 4.13a illustrates the distribution of entities by click frequencies, and the correlation of top popular entities (measured by total navigations) across different months is shown in Figure 4.13b. In general, we observe that the user navigation activities in the

	% new e_s	% with new e_t	% w. new e_t in top-30	# new e_t (avg.)
08-2016	24.31	71.18	15.54	18.25
04-2016	30.61	66.72	53.44	42.20

Table 4.3: Statistics on the dynamic of clickstream, e_s denote source entities, e_t related entities.

top popular entities are very dynamic, and changes substantially with regard to time. Figure 4.13c visualizes the dynamics of related entities toward different ranking sections (e.g., from rank 0 to rank 20) of different months, in terms of their correlation scores. It can be interpreted that the entities that stay in top-20 most related ones tend to be more correlated than entities in bottom-20 when considering top-100 related entities.

As we show in Table 4.3, there are 24.31% of entities in top-10,000 most active entities of September 2006 do not appear in the same list the previous month. And 30.61% are new compared with 5 months before. In addition, there are 71% of entities in top-10,000 having navigations to new entities compared to the previous month, with approx. 18 new entities are navigated to, on average. Thus, the datasets are naturally very dynamic and sensitive to change. The substantial amount of missing *past* click logs on the **newly-formed relationships** also raises the necessity of an dynamic measuring approach.

Figure 4.14 shows the overall architecture of our framework, which consists of three major components: *time*-, *graph*- and *content*-based networks. Each component can be considered as a separate sub-ranking network. Each network accepts a tuple of three elements/representations as an input in a *pair-wise* fashion, i.e., the source entity e_s, the target entity e_t with higher rank (denoted as $e_{(+)}$) and the one with lower rank (denoted as $e_{(-)}$). For the *content* network, each element is a sequence of terms, coming from entity textual representation. For the *graph* network, we learn the embeddings from the entity linking graph. For the *time* network, we propose a new convolutional model learning from the entity temporal signals. More detailed are described as follows.

Neural Ranking Model Overview

The entity relatedness ranking can be handled by a point-wise ranking model that learns to predict relatedness score directly. However, as the navigational frequency distribution is often *skewed* at top, supervisions guided by long-tail navigations would be prone to errors. Hence instead of learning explicitly a calibrated scoring function, we opt for a *pair-wise* ranking approach. When applying to ranking top-k entities, this approach has the advantage of correctly predicting partial orders of different relatedness functions f_t at any time points regardless of their non-transitivity [CHWW12].

This work builds upon the idea of *interaction*-based deep neural models, i.e. learning soft semantic matches from the source-target entity pairs. Note that, we do not aim for a Siamese architecture [CHL05] (i.e., in *representation*-based models), where the weight parameters are shared across networks. The reason is that, the conventional kind of network

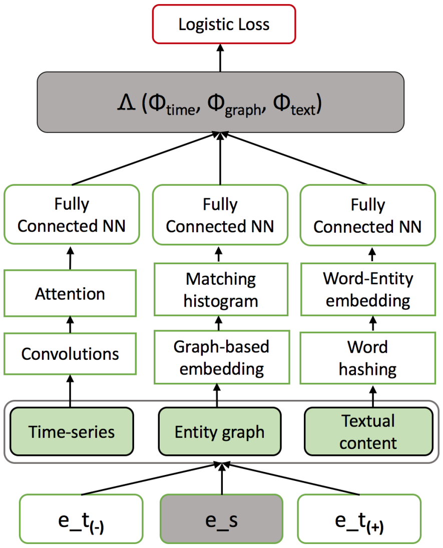


Figure 4.14: The trio neural model for entity ranking.

produces a *symmetric* relation, violating the **asymmetric** property of the relatedness function f_t (section 4.2.2). Concretely, each deep network ψ consists of an input layer z_0, $n-1$ hidden layers and an output layer z_n. Each hidden layer z_i is a fully-connected network that computes the transformation: $z_i = \sigma(\mathbf{w}_i \cdot z_{i-1} + b_i)$, where $\mathbf{w}_i$ and b_i are the weight matrix and bias at hidden layer i, σ is a non-linear function such as the rectified linear unit(ReLU). The final score under the trio setup is summed from multiple networks.

$$\phi(< e_s, e_{(+)}, e_{(-)} >) = \phi_{time} + \phi_{graph} + \phi_{content} \tag{4.1}$$

In the next section we describe the input representations z_0 for each network.

4.2.4 Entity Relatedness Ranking

Content-based representation learning

To learn the entity representation from its content, we rely on entity textual document (*word*-based) as well as its link profile (*entity*-based) (section 4.2.2). Since the vocabulary size of entities and words is often very large, conventional one-hot vector representation becomes expensive. Hence, we adopt the *word hashing* technique from [HHG+13], that breaks a term into character *trigraphs* and thus can dramatically reduce the size of the vector dimensionality. We then rely on embeddings to learn the distributed representations and build up the soft semantic interactions via input concatenation. Let $\mathsf{E} : \mathscr{V} \rightarrow \mathbb{R}^m$ be the embedding function, $\mathscr{V}$ is the vocabulary and m is the embedding size. $\mathsf{w} : \mathscr{V} \rightarrow \mathbb{R}$, is the weighting function that learns the global term importance and a weighted element-wise sum of word embedding vectors -*compositionality* function $\oplus$, the word-based representation for entity e is hence $\oplus_{i=1}^{|e_w|}(\mathsf{E}(w_i), \mathsf{w}(w_i))$. For entity-based representation, we break down the *surface form* of a linked entity into bag-of-words and apply analogously. The concatenation of the two representations for the tuple $< e_s, e_{(+)}, e_{(-)} >$ is then input to the deep feed-forward network.

Graph-based representation

To obtain the graph embedding for each entity, we adopt the idea of DeepWalk [PARS14], which learns the embedding by predicting the vertex sequence generated by random walk. Concretely, given an entity e, we learn to predict the sequence of entity references $\mathbb{S}_e$ – which can be considered as the *graph-wise* context in the Skip-gram model. We then adopt the matching histogram mapping in [GFAC16] for the *soft interaction* of the ranking model. Specifically, denote the bag of entities representation of e_s as $\mathbb{C}_{e_s}$, and that of e_t as $\mathbb{C}_{e_t}$; we discretize the soft matching (calculated by cosine similarity of the embedding vectors) of each entity pair in $(\mathbb{C}_{e_s}, \mathbb{C}_{e_t})$ into different bins. The logarithmic numbers of the count values of each bin then constitute the interaction vector. This soft-interaction in a way is similar in the idea with the traditional link-based model [WM08], where the relatedness measure is based on the overlapping of in-coming links.

Attention-based CNN for temporal representation

For learning representation from entity temporal signals, the intuition is to model the low-level *temporal correlation* between two *multivariate* time series. Specifically, we learn to embed these time series of equal size T into an Euclidean space, such that similar pairs are close to each other. Our embedding function takes the form of a convolutional neural network (CNN), shown in Figure 4.15. The architecture rests on four basic layers: a 1-D convolutional (that restricts the slide only along the time window dimension, following [ZLC$^+$14]), a batch-norm, an attention-based and a fully connected layer.

Convolution layer: A 1-D convolution operation involves applying a filter $\mathbf{w}_f \in \mathbf{R}^{1 \times w \times D}$ (i.e., a matrix of weight parameters) to each subsequence X_e^i of window size m to produce a new abstraction.

$$q_i = \mathbf{w}_f \mathsf{L^i}_{t:t+m-1,D} + b; \; s_i = BN(q_i); \; h_i = ReLU(s_i) \tag{4.2}$$

where $\mathsf{L^i}_{t:t+w-1,D}$ denotes the concatenation of w vectors in the lookup layer representing the subsequence X_e^i, b is a bias term. The convolutional layer is followed by a batch normalization (BN) layer [IS15], to speed up the convergence and help improve generalization.

Attention Mechanism: We apply an attention layer on the convolutional outputs. Conceptually, attention mechanisms allow NN models to focus selectively on only the important features, based on the attention weights that often derived from the interaction with the target or within the input itself (self-attention) [VSP$^+$17]. We adopt the former approach, with the intuition that the time-spatial patterns should not be treated equally, but the ones near the studied time should gain more focus. To ensure that each feature in $\mathbb{F}_i^c$

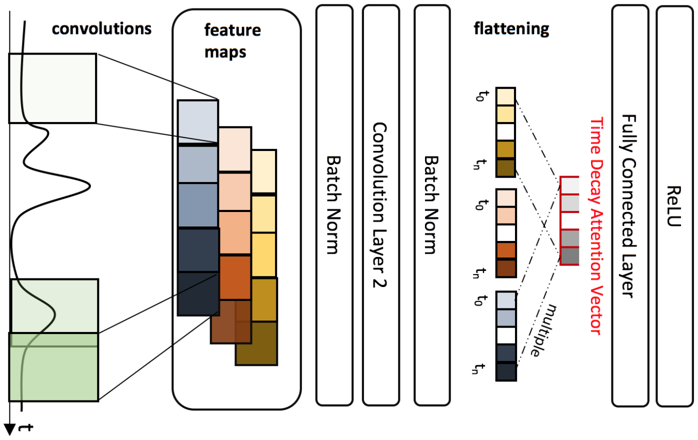


Figure 4.15: The attentional CNN for time series representation.

that associates with different timestamps are rewarded differently, the attention weights are guided by a time-decay weight function, in a *recency*-favor fashion. More formally, let $A \in \mathbf{R}^{T-w+1\times 1}$ be the time *context* vector and $\mathbb{F}_i^c \in \mathbf{R}^{1\times(T-w+1)}$ the output of convolution for X. Then the kth column of the re-weighted feature map $\mathbb{F}_i^h$ is derived by:

$$\mathbb{F}_i^h[:,k] = A[k] \cdot \mathbb{F}_i^c[:,k], k = 1 \cdots T - w + 1 \tag{4.3}$$

The time context vector a is generated by a decay weight function, since each column k in the vector is associated with a time t_k which is $T - k + w$ time units away from studied time t.

Decay weight function: we leverage the Polynomial Curve for the function. $PD(t_i, t) = \frac{1}{(t-t_i)^{\alpha}+1}$, whereas α defines the decay rate. It is worth noting that when α is increased, the attention layer acts just like a *pooling* one [6]. Stacking up multiple convolutional layers is possible, in this case $|A|$ is the size of the previous layer. The attention layer is only applied to the **last** convolution layer in our architecture. The output of the attention layer is then passed to a fully-connected layer with non-linear activation to obtain the **temporal** representation.

Learning and Optimization

Finally, we describe the optimization and training procedure of our network. We use a Logarithmic loss that can lead to better probability estimation at the cost of accuracy [7].

[6] Note that, for clear visualization, we put *flattening* before attention layer in Figure 4.15

[7] Other ranking-based loss such as Hinge loss favours over sparsity and accuracy (in the sense of direct punishing misclassification via margins) at the cost of probability estimation. The logistic loss distinguishes

Our network minimizes the cross-entropy loss function as follows:

$$L = -\frac{1}{N}\sum_{i=1}^{N}[P_{\{e_s,e_1,e_2\}_i}\log\bar{y}_i + (1-P_{\{e_s,e_1,e_2\}_i})\log(1-\bar{y}_i)] + \lambda|\theta|_2^2 \quad (4.4)$$

where N is the training size, $\bar{y}$ is the output of the sigmoid layer on the predicted label. θ contains all the parameters of the network and $\lambda|\theta|_2^2$ is the L2 regularization. $P_{\{e_s,e_{(+)},e_{(-)}\}_i}$ is the probability that $e_{(+)}$ is ranked higher than $e_{(-)}$ derived from entity navigation, $P_{\{e_s,e_{(+)},e_{(-)}\}_i} = y^{t(i)}_{\{e_s,e_{(+)}\}}/(y^{t(i)}_{\{e_s,e_{(+)}\}} + y^{t(i)}_{\{e_s,e_{(-)}\}})$, where $t(i)$ is the observed time point of the training instance i. The network parameters are updated using Adam optimizer [KB14].

4.2.5 Experiments

Dataset

To recap from Section 4.2.3, we use the click stream datasets in 2016. We also use the corresponding Wikipedia article dumps, with over 4 million entities represented by actual pages. Since the length of the content of an Wikipedia article is often long, in this work, we make use of only its **abstract** section. To obtain temporal signals of the entity, we use page view statistics of Wikipedia articles and aggregate the counts by month. We fetch the data from June, 2014 up until the studied time, which results in the length of 27 months.

Seed entities and related candidates. To extract popular and trending entities, we extract from the clickstream data the top 10,000 entities based on the number of navigations from major search engines (*Google* and *Bing*), at the studied time. Getting the subset of related entity candidates –for efficiency purposes– has been well-addressed in related work [GB14, PFC17]. In this work, we do not leverage a method and just assume the use of an appropriate one. In the experiment, we resort to choose only candidates which are visited from the seed entities at studied time. We filtered out entity-candidate pairs with too few navigations (less than 10) and considered the top-100 candidates.

Models for Comparison

We compare our models against the following baselines.

Wikipedia Link-based (WLM): [WM08] proposed a low-cost measure of semantic relatedness based on Wikipedia entity graph, inspired by Normalized Google Distance.

DeepWalk (DW): DeepWalk [PARS14] learned representations of vertices in a graph with a random walk generator and language modeling. We chose not to compare with the matrix factorization approach in [ZLS15], as even though it allows the incorporation of

better between examples whose supervision scores are close.

	Counts
Total seed entities	$10,000$
Total entities	$1,420,819$
Candidate per entities (avg.)	142
Training seed entities	$8,000$
Dev. seed entities	$1,000$
Test seed entities	$1,000$
Training pairs	$100,650K$
Dev. pairs	$12,420K$
Test pairs	$12,590K$

Table 4.4: Statistics of the dataset.

different relation types (i.e., among entity, category and word), the iterative computation cost over large graphs is very expensive. When consider only entity-entity relation, the performance is reported rather similar to DW.

Entity2Vec Model (E2V): or entity embedding learning using Skip-Gram [MSC+13] model. E2V utilizes textual information to capture latent word relationships. Similar to [ZLS15, NXC+16], we use Wikipedia articles as training corpus to learn word vectors and reserved hyperlinks between entities.

ParaVecs (PV): [LM14, DOL15] learned document/entity vectors via the distributed memory (**ParaVecs-DM**) and distributed bag of words (**ParaVecs-DBOW**) models, using hierarchical softmax. We use Wikipedia articles as training corpus to learn entity vectors.

RankSVM: [CLO+13] learned entity relatedness from a set of 28 **handcrafted features**, using the traditional learning-to-rank method, RankSVM. We put together additional well-known temporal features [KNN14b, ZRZ16] (i.e., time series cross correlation, trending level and predicted popularity based on *page views*) and report the results of the extended feature set.

For our approach, we tested different combinations of *content* (denoted as **Content$_{\textbf{Emb}}$**), *graph*, (**Graph$_{\textbf{Emb}}$**) and *time* (**TS-CNN-Att**) networks. We also test the *content* and *graph* networks with **pretrained** entity representations (i.e., ParaVecs-DM and DeepWalk).

Experimental Setup

Evaluation procedures. The time granularity is set to months. The studied time t_n of our experiments is September 2016. From the seed queries, we use 80% for training, 10% for development and 10% for testing, as shown in Table 4.4. Note that, for the **time-aware** setting and to avoid leakage and bias as much as possible, the data for training and development (including supervision) are up until time $t_n - 1$. In specific, for content and graph data, only $t_n - 1$ is used.

Metrics. We use 2 correlation coefficient methods, Pearson and Spearman, which have been used often throughout literature, cf. [DNLH16, PFC17]. The Pearson index focuses on the difference between predicted-vs-correct relatedness scores, while Spearman focuses on the ranking order among entity pairs. Our work studies on the strength of the dynamic relatedness between entities, hence we focus more on Pearson index. However, traditional correlation metrics do not consider the positions in the ranked list (correlations at the top or bottom are treated equally). For this reason, we adjust the metric to consider the rankings at specific top-k positions, which consequently can be used to measure the correlation for only top items in the ranking (based to the ground truth). In addition, we use Normalized Discounted Cumulative Gain (NDCG) measure to evaluate the recommendation tasks.

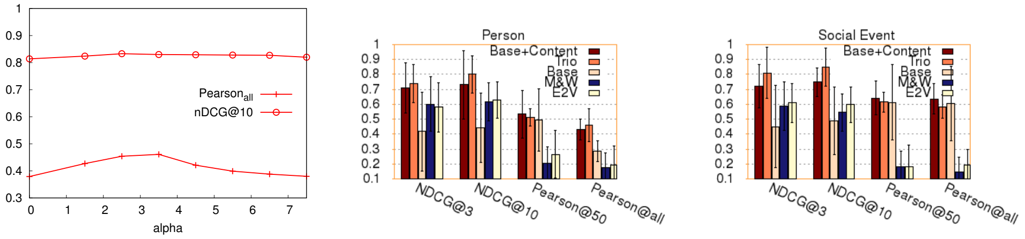
Implementation details. All neural models are implemented in TensorFlow. Initial learning rate is tuned amongst {1.e-2, 1.e-3, 1.e-4, 1.e-5}. The batch size is tuned amongst {50, 100, 200}. The weight matrices are initialized with samples from the uniform distribution [GB10]. Models are trained for maximum 25 epochs. The hidden layers for each network are among {2, 3, 4}, while for hidden nodes are {128, 256, 512}. Dropout rate is set from {0.2, 0.3, 0.5}. The pretrained DW is empirically set to 128 dimensions, and 200 for PV. For CNN, the filter number are in {10, 20, 30}, window size in {4, 5, 6}, convolutional layers in {1, 2, 3} and decay rate α in {1.0, 1.5,⋯,7.5}. 2 conv- layers with window size 5 and 4, number of filters of 20 and 25 respectively are used for decay hyperparameter analysis.

	Model	**Pearson** ×100				$\rho \times 100$	**nDCG** (proxy)			**nDCG** (human)		
		@10	@30	@50	all	all	@3	@10	@20	@3	@10	@20
Baselines	WLM	27.6	28.3	24.0	19.4	12.1	0.63	0.59	0.62	0.50	0.46	0.52
	RankSVM	28.5	34.7	31.4	20.7	27.5	0.65	0.61	0.64	0.52	0.61	0.65
	Entity2Vec	18.6	22.0	21.8	20.5	18.7	0.62	0.60	0.61	0.54	0.53	0.54
	DeepWalk	31.3	30.9	21.4	17.6	10.1	0.41	0.43	0.47	0.34	0.38	0.45
	ParaVecs-DBOW	18.6	22.0	21.8	20.5	16.0	0.62	0.60	0.61	0.50	0.50	0.55
	ParaVecs-DM	19.0	23.0	23.2	22.3	18.3	0.66	0.63	0.63	0.49	0.52	0.58
Model Ablation	TS-CNN	51.9	51.0	43.0	35.8	26.5	0.41	0.43	0.47	0.40	0.43	0.48
	TS-CNN-Att (**Base**)	57.9	49.7	44.7	37.1	24.9	0.43	0.44	0.49	0.38	0.45	0.50
	Base+PV	60.6	44.2	41.4	36.4	11.2	0.41	0.43	0.47	0.49	0.51	0.55
	Base+DW	43.5	36.5	35.7	32.7	31.0	0.44	0.48	0.53	0.47	0.51	0.52
	Base+PV+DW	56.9	46.1	43.4	32.9	28,4	0.41	0.44	0.48	0.49	0.54	0.57
	$Content_{Emb}$+$Graph_{Emb}$	48.9	40.1	49.9	37.5	27.9	0.67	0.62	0.70	0.61	0.69	0.65
	Base+$Content_{Emb}$	**67.1**	54.2	**53.4**	43.7	26.5	0.67	0.69	0.71	0.61	0.72	0.74
	Base+$Graph_{Emb}$	55.2	50.2	41.3	31.5	35.5	0.71	0.75	0.78	0.65$^{\mp}$	0.78$^{\mp}$	0.81$^{\mp}$
	Trio	58.6	**54.3**	50.2	**45.4**	**43.5**	**0.75**	**0.78**	**0.83**	**0.74**$^{\mp}$	**0.82**$^{\mp}$	**0.85**$^{\mp}$

Table 4.5: Performance of different models on task (1) Pearson, Spearman's ρ ranking correlation, and task (2) recommendation (measured by nDCG). Bold and underlined numbers indicate best and second-to-best results. $\mp$ shows statistical significant over WLM ($p < 0.05$).

Experimental Tasks

We evaluate our proposed method in two different scenarios: (1) Relatedness ranking and (2) Entity recommendation. The first task evaluates how well we can **mimic** the ranking via the entity navigation. Here we use the raw number of navigations in Wikipedia clickstream. The second task is formulated as: *given an entity, suggest the top-k most related entities to it right now*. Since there is no standard ground-truth for this temporal task, we constructed



(a) Decay parameter for time-series embedding.

(b) Model performances for *person*-type entities.

(c) Model performances for *social event*-type entities.

Figure 4.16: Performance results for variation of decay parameter and different entity types.

two relevance ground-truths. The **first** one is the *proxy* ground-truth, with relevance grade is *automatically* assigned from the (top-100) most navigated target entities. The graded relevance score is then given as the *reversed* rank order. For this, all entities in the test set are used. The **second** one is based on the human judgments with 5-level graded relevance scale, i.e., from 4 - highly relevant to 0 - not (temporally) relevant. Two human experts evaluate on the subset of 20 entities (randomly sampled from the test set), with 600 entity pairs (approx. 30 per seed, using *pooling* method). The ground-truth size is comparable the widely used ground-truth for *static* relatedness assessment, KORE [HSN^{+}12]. The Cohen's Kappa agreement is 0.72. Performance of the best-performed models on this dataset is then tested with paired *t*-test against the WLM baseline.

Results on Relatedness Ranking

We report the performance of the relatedness ranking on the left side of Table 4.5, with the Pearson and Spearman metrics. Among existing baselines, we observe that link-based approaches i.e., WLM and DeepWalk perform better than others for top-k correlation. Whereas, temporal models yield substantial improvement overall. Specifically, the TS-CNN-Att performs better than the no-attention model in most cases, improves 11% for Pearson@10, and 3% when considering the total rank. Our trio model performs well overall, gives best results for total rank. The duo models (combine base with either pretrained DW or PV) also deliver improvements over the sole temporal ones. We also observer additional gains while combining of temporal base with pretrained DW and PV altogether.

Results on Entity Recommendation

Here we report the results on the nDCG metrics. Table 4.5 (right-side) demonstrates the results for two ground-truth settings (proxy and human). We can observe the good performance of the baselines for this task over conventional temporal models, significantly for *proxy* setting. It can be explained that, 'static' entity relations are ranked high in the non time-aware baselines, hence are still rewarded when considering a fine-grained grading

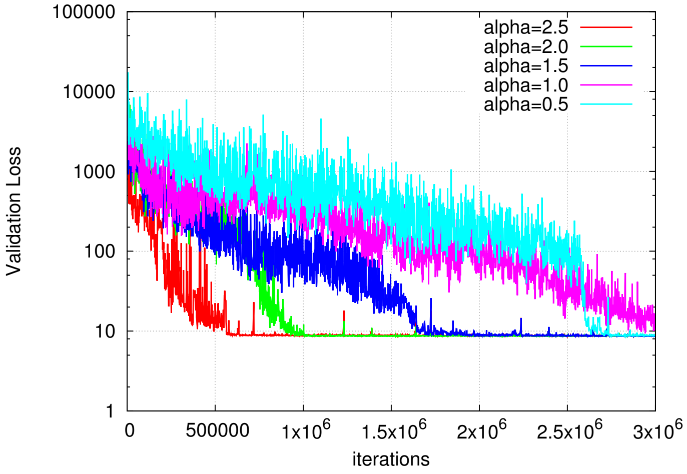


Figure 4.17: Convergence of decay parameters.

Models			
PV-DM	**TS-CNN-Att**	**Temp+PV**	**Trio**
Secret Service	Halle Berry	*Elton John*	Mark Strong
Spider-Man	*X-Men*	Taron Egerton	Jeff Bridges
Taron Egerton	Jeff Bridges	Edward Holcroft	Julianne More

Table 4.6: Different top-k rankings for entity *Kingsman: The Golden Circle*. Italic means irrelevance.

scale (100 levels). The margin becomes smaller when comparing in *human* setting, with the standard 5-level scale. All the models with pretrained representations perform poorly. It shows that for this task, early interaction-based approach is more suitable than purely based on representation.

Additional Analysis

We present an anecdotic example of top-selected entities for Kingsman: The Golden Circle in Table 4.6. While the content-based model favors old relations like the preceding movies, TS-CNN puts popular actress Halle Berry or the recent released X-men: Apocalypse on top. The latter is not ideal as there is not a solid relationship between the two movies. One implication is that the two entities are ranked high is more because of the popularity of themself than the strength of the relationship toward the source entity. The Trio model addresses the issue by taking other perspectives into account, and also balances out the recency and long-term factors, gives the best ranking performance.

Analysis on decay hyper-parameter. We give a study on the effect of decay parameter on performance. Figure 4.16a illustrates the results on $Pearson_{all}$ and nDCG@10 for the *trio* model. It can be seen that while nDCG slightly increases, Pearson score peaks while α in the range $[1.5, 3.5]$. Additionally, we show the convergence analysis on α for TS-CNN-

Att in Figure 4.17. Bigger α tends to converge faster, but to a significant higher loss when α is over 5.5 (omitted from the Figure).

Performances on different entity types. We demonstrate in Figures 4.16b and 4.16c the model performances on the *person* and *event* types. WLM performs poorer for the latter, that can be interpreted as link-based methods tend to slowly adapt for recent trending entities. The temporal models seem to capture these entites better.

4.2.6 Conclusion

In this work, we presented a trio neural model to solve the dynamic entity relatedness ranking problem. The model jointly learns rich representations of entities from textual content, graph and temporal signals. We also propose an effective CNN-based attentional mechanism for learning the temporal representation of an entity. Experiments on ranking correlations and top-*k* recommendation tasks demonstrate the effectiveness of our approach over existing baselines. For future work, we aim to incorporate more temporal signals, and investigate on different 'trainable' attention mechanisms to go beyond the time-based decay, for instance by incorporating latent topics.

5 Time-aware ranking for Web Archives

5.1 Temporal Anchor-text mining

Web search is good in delivering up-to-date or fresh information for topics of all types. Due to its vivid and wide used and participative content creation, the Web is in addition a good reflection of processes, practices, and topics in all areas of life including politics, society, science. When we regularly take snapshots of the Web at different times as it is done in Web archiving (at least for part of the Web), we can, thus, capture this world reflection at different times as well as its evolution via subsequent versions. Thus, web archives have the potential to provide a rich source of first-hand information from the past - and about how things evolved. It can, for example, be seen how topics such as integration, nuclear power or democracy where discussed in the early 90's - and how this discussion changes over time. In addition, looking at more mundane issues, in some decades from now we can see what people did wear, eat, and talk about in 2015 from archived evidences. Although such content might seem trivial in the first place, it accumulates into an unpreceded form grass-root historical records.

Although such information from the past might still be findable in the current Web, they are typically aggregated, filtered and interpreted from a current perspective. For experts and professionals such as journalists, researchers from political science and sociology, historians, a first-hand and unbiased reflection of the world opens up the investigation of own stories and completely new research questions. It enables them to better understand how and why issues, for example, controversial topics evolved over time. They can also see the context of such discussions and have a first-hand account of change such as the evolution of language.

Anchor texts have been shown as an important factor of the Web that can be used to mimic the behavior of the query logs [DC], representing documents [DDb] or subtopic mining [DHC$^+$]. However, none of the above have studied the temporal dynamics of the anchor texts for the subtopic mining tasks. In a recent work, Kanhabua and Nejdl [KNa] studied the terminology evolution of entities by means of anchor text in the context of

Wikipedia. However, different from Wikipedia, the timestamp annotations in the real web archives are a lot noisier due to the unreliability of the crawling time. In this section, we present our first study of mining the temporal dynamics of subtopics in the web archives for the time-based search result diversification task. The quality of timestamp annotations (i.e., crawling time) in the web archives at the document level is rather unpredictable. Table 5.1 illustrates an example of how unreliable the crawling time is for certain circumstances. Two documents that mentioned the incident related to the subtopic late-term abortion that involved with the Governor *Kathleen Sebelius* are only first crawled 3-4 years after. The actual timing for the trending of the subtopic is however in April 2009. This shows a significant lagging between the publishing date of a page and when it is actually crawled.

Our contributions in this section are: (1) we address the problem of mining temporal subtopic in a web archive, with the respect to time uncertainty, (2) we introduce a method to extract reliable publication dates from the web archive resources and (3) we exploit anchor text as a good source of mining temporal subtopics in the web archives and propose a method to infer relevant time (or a date) for the temporal subtopic.

Table 5.1: Example of relevant documents for the *late-term abortion* subtopic

URL	Crawling time	Actual date	Content
http://www.vitter.senate.gov/..sebelius-appointment	2013-02-22	2009-04-20	Vitter Voices Grave Concerns Over Sebelius Appointment Monday, April 20, 2009 'I was already concerned about the Governor's position on a number of issues, especially those relating to abortion,' said Vitter. 'The fact that Governor Sebelius has accepted thousands of dollars in campaign contributions from George Tiller - a highly-controversial individual who specializes in performing late-term abortions, a practice far beyond those performed in the majority of abortion clinics - provides some insight into her views on abortion that raise many important concerns about her nomination..
http://lamborn.house.gov/..pro-abortion-veto/	2012-11-09	2009-04-24	Lamborn Comments on Governor Sebelius Pro-Abortion Veto Apr 24 2009 Calls yesterday's action preview of extreme agenda Washington, Apr 24 - Congressman Doug Lamborn (CO-05) today released the following statement regarding President Obama's nomination for Secretary of the Department of Health and Human Services, Governor Kathleen Sebelius of Kansas, in response to her latest pro-abortion action. Yesterday, she vetoed a common sense bill that would have required doctors performing late-term abortions to report additional information on those procedures to the Kansas Department of Health and Environment. The bill would have also given women the right to sue the doctors, should they later believe their abortions were illegal..

5.1.1 Related Work

There have been several works on mining aspects from the anchor texts [DXC, DC, DHC$^+$], however they only mine on the current snapshot of the Web. The temporal dynamics of subtopics is first studied in [NK14] and is used to improve the ranking effectiveness of such queries at particular times. Dai et al. [DDb] also study the trending of the anchor texts by looking back at the historical web snapshot to improve the weighting function for document retrieval task. In the web archive context, there has been no existing work so far studied the temporal dynamics of subtopics.

5.1.2 Temporal subtopic mining from anchor text

Anchor text created by web content editors often reflect high quality summarizations of the destination pages. As anchor texts are often short and descriptive, it is shown to possess similar characteristics with web queries [DC]. Anchor text with regards to a topic/query (contain the query terms) often convey diverse aspects of the topic, hence is a good source of subtopic mining. For temporal subtopic mining in the web archives, with the absence of the query logs and the unreliability of state-of-the-art retrieval models (in retrieving top-K relevant documents at a time-period), we observe temporal anchor text reflects a good correlation with the temporal subtopic. The correlation is further elaborated in Section 5.1.5. For a given query q, we first get all anchor texts containing all query terms of q, weight them, and select the most important ones as subtopics. We follow the weighting mechanism proposed in [DHC$^+$], where they observe that the importance of an anchor text is usually proportional to its popularity on the Web, i.e., how frequent it is used among web pages. The importance score is also traded off against the length of the anchor text. The anchor text c with respect to query q is weighted as:

$$\begin{aligned} f(q,c) &= freq(c) \times rel(q,c) \\ &= [N_c^{site} + log(N_c^{page} N_c^{site} + 1)] \times \frac{1 + len(q)}{len(c)} \end{aligned} \tag{5.1}$$

whereas $freq(c)$ is the frequency of the anchor text c, N_c^{site} is the number of unique sites contain c and N_c^{page} is the number of pages contain c.

5.1.3 Subtopic Extraction

In order to construct the set of distinctive and high quality subtopics/aspects of a query, we also need to apply a clustering technique on the anchor texts, as follows [DXC]:

Similarity measures

Relevant models The similarity measure between anchor text pairs is not effective if based solely on the content of the anchor text, that often contains few terms. Instead, an anchor text is represented as the accumulated of top-k documents that are relevant to it. Specifically, each anchor text a is represented by the relevance model $P_a(w|R)$ estimated from the top-10 documents returned by the query likelihood retrieval model (we don't need a temporal retrieval model in this step) for a. The similarity of two subtopics c_1 and c_2 is then calculated as the KL-divergence between their relevance models $Pr_1(w|R)$ and $Pr_2(w|R)$. However, building relevance models for every anchor text is relatively computationally expensive.

Co-occurrence At Passage Level Follow Dang et al. [DXC], we also conduct a more efficient method based on passage analysis. The idea is that two anchor texts are more similar if they co-occur often in the same text passages. Therefore, for every pair of anchor texts c_i and c_j , we compute N_i and N_j - the number of passages in which each of them occurs, and N - the number of passages in which they co-occur. The similarity between c_i and c_j is given by the Jaccard score:

$$sim(c_i,c_j) = \frac{N}{N_i + N_j N} \tag{5.2}$$

Clustering Algorithms

Affinity Propagation Algorithm Follow [SZG$^+$14, NK14], we use Affinity Propagation (AP) for the clustering task. AP has an advantage over other clustering algorithm that it determines the number of clusters automatically.

5.1.4 Time inference for temporal subtopics

Due to the unreliability of the timestamp annotations in the web archives (i.e., the crawling time), detecting the trending period of a subtopic is not straight-forward. We use a brute-force approach (explained in detail in Section 5.1.5) to extract the most-reliable publication dates out of the web archives to acquire a substantial subset of documents [1] with highly-reliable dates. Our idea is to leverage this high quality temporal sub-collection to infer the relevant date for the temporal subtopics. One can think of mining the subtopics and its relevant times (e.g., via the frequency distribution) directly from this time-reliable sub-collection. However, beside the incompleteness of the sub-collection, it is difficult to infer trending behaviors of the temporal subtopics.

Temporal language model Given our temporal collection $\mathscr{C}$ with a set of time-partitions $\mathscr{T} = \{t_1, t_2, ..., t_n\}$, our task is to weight a temporal subtopic c with respect to each time-partition. This time-interval ranking approach is based on the temporal language model presented in [KNb]. The idea is to assign a probability to a time partition according to word usage or word statistics over time. A normalized log-likelihood ratio is used to compute the similarity between two language models. Here, we expand a subtopic c as the accumulated set of all the anchor text's terms in its cluster (explain in the previous section), removing all the duplicates.

$$S(c_i,t_j) = \sum_{word \in c_i} P(word|c_i) log \frac{P(word|t_j)}{P(word|(C)} \tag{5.3}$$

The $S(c_i,t_j)$ is the probability that a temporal subtopic c_i is relevant to the time period t_j.

Connection with the time-based search result diversification

[1] since our queries are *informational*, the revisions are mostly of duplicated content, hence we only consider at document-level. In detail, revisions with the same publication date are merged into the oldest revision.

Algorithm 1 *Time-based IA-Select($\mathscr{R}$)*

```
INPUT: $k, q, C(q), R(q), C(d), P(c|q), V(d|q,t)$
Output: set of documents $S$
$S = \emptyset$
$\forall t, U(t|q,S) = P(t|q)$
while $|S| < k$ do
    for $d \in R(q)$ do
        $q(d|q,t,S) = \sum_{t \in T(d)} U(t|q,S)V(d|q,t)$
    end for
    $d^* = argmax\ g(d|q,t,S)$
    $S = S \cap d^*$
    $\forall t \in T(d^*), U(t|q,S) = (1 - V(d^*|q,t))U(t|q,S \backslash d^*)$
    $R(q) = R(q) \backslash d*$
end while
return S
```

Search result diversification is meant for diversifying the result list so that the top-k covers all the aspects of an ambiguous query. In the web archive context, the requirement is rather more complicated as its also essential to cover the time-periods where the aspects/subtopics are trending, ranked based on the 'trending weight' of the subtopics. Hence, the objective function of the diversification ranking needs to be re-designed to take this temporal subtopic factor into account. Basically, it needs to diversify over two distinct dimensions (i.e., time and aspects) and present them in a comprehensive way (i.e., *federated/vertical* search). Previous works did not however take the two important factors into account in a unified framework. Berberich el al. [BB] only consider diversifying over the time dimension where they consider each time-period is a query aspect. Nguyen et al. [NK14] take both time and aspect into account but for their recency-favor ranking model.

Search result diversification is meant for diversifying the result list so that the top-k covers all the aspects of an ambiguous query. In our case, we adapt a state-of-the-art result diversification model, a.k.a IA-Select. [AGHI09b] and exploit the time-period ranking to achieve a list of documents where top-k documents covers top-n interesting time-points.

We derive $V(d|q,t)$ from the probability that a document d_i is relevent to the time period tj (Equation 5.3).

5.1.5 Experiments

Dataset

.gov domain collection We utilized a full corpus of archival web pages in .gov domain collected by the Internet Archive from January 1995 to September 2013. The corpus contains over 900 million of text captures and over 58.8 billion temporal links. Figures 6.2 and 5.2 show the document and document/revision ratio distribution of the collection. We

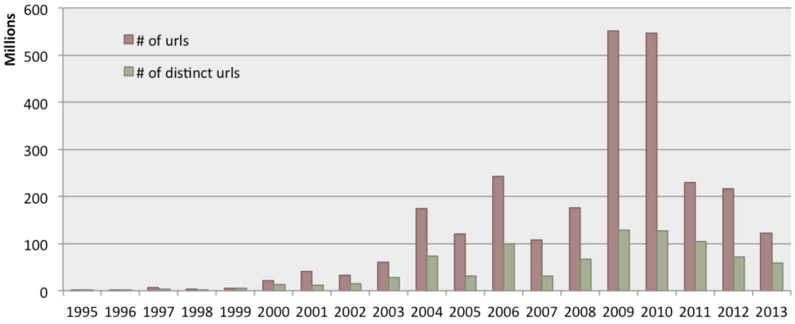


Figure 5.1: The document distribution

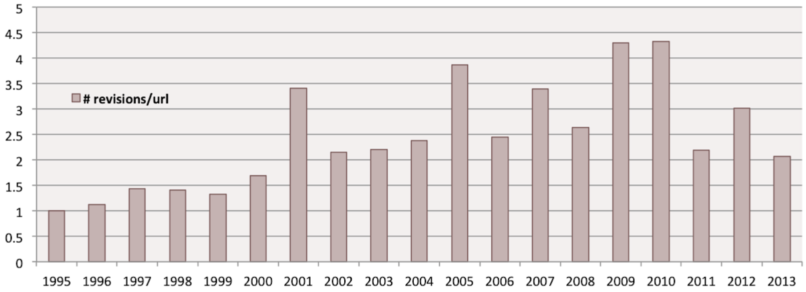


Figure 5.2: The document/revision ratio distribution

extracted the anchor text and its timestamp and built a temporal language model for every monthly bin.

Determining time of the crawled web document

Identifying the *publication time* for a crawled web document is a difficult task, as the crawling time is not a totally reliable source. We confide on 5 different sources (ranked by level of reliability). We judge the level of reliability by judging the accuracy level of different source (with random subset of 300 documents). The URL source is of 96%, content source is of 90%, whereas we dont have enough clue to judge the HTTP header sources. There are methods in related work [OPPSS16] for *inferring* the actual publication date of web pages based on their link-based neighborhood. However, to not rely on any proxy, we opt for not applying or addressing the issue in this work.

- Date extraction from URLs. Often, a web article's link contains the date of creation (e.g., http://www.whitehouse.gov/the-press-office/2011/12/24/blah). We define this source as *1st level* of confidence, called *very strong*. There are also cases that only dates at month granularity are provided in the URL, we mark it as *mildly strong* (that if day granularity- publication date can be extracted from the content then we will use it instead).
- Publication date from document content. The publication date often lies in the first

one-two lines of the article content. We use a temporal tagger called Heideltime [2] to exact the first date (if there is) out of this text snippet. This is *2nd level* of confidence, called *strong*.

- Last-Modified Date from HTTP Header. The Last-Modififed Date is however, often not quite different from the creation date in our case as the modification (if there is, occurs shortly after the article is online), called *acceptable*.
- Creation date from HTTP Header. This is ranked as *3-level*, called *weakly acceptable*.
- Crawling date, the *undetermined* source.

Table 5.4 describes the statistics of our date extraction method. Even the percent of number of *strongly*-confident date extractions are not high (approx. 15%), we believe that this is still accountable as the *important* documents appear in top-K results tends to have a good template and are easier to determine the publication time.

Preliminary results

Correlation with the query logs Figure 5.3 illustrates the time-series (represented as normalized frequency in monthly bins) between of the query electoral college, mining from three different sources (i.e., anchor text, content and query logs (from Google Trend)). We use *cross correlation* (ccf) $f \star g(\tau)$ to measure time series of the two time series. The lagging time τ_{delay} is calculated as $argmax_t(f \star g(t))$. This preliminarily shows the rather ineffectiveness of the accumulated document frequency in capturing the temporal dynamics of the controversial queries($ccf = 0.68$ with $\tau_{delay} = 9$). Instead, even with some lagging ($\tau_{delay} = 2$), the correlation between anchor text $f(t)$ and the query logs $g(t)$ is rather high ($ccf = 0.69$). This correlation is further illustrates in Figure 5.4. This rather shows the value of the anchor text in capturing the temporal dynamics in the web archives. The correlation is not clear for every queries, as it is affected by many factors (e.g., the event-relatedness and its impact). We leave a quantitative evaluation and deeper analysis for future work.

Analysis on the temporal subtopic mining Figure 5.5 depicts the temporal dynamics - reflexed by the accumulated document frequency of the subtopic late-term abortion from two different time reliability sources. The first one is from the crawling time, the second one is extracted only from our date extraction process with strong level of confidence. Our assumption is high quality pages often follow standard templates and hence their publication dates are often easily extracted by our process. When an event happens (which leads to the trending of some subtopics), the amount of high quality pages issued also gets higher. Hence, we can partially use this second source of time reliability to represent the trending of a subtopic. This brings us a subtopic with the high reliability of timestamp. Figure 5.5

[2] https://code.google.com/p/heideltime/

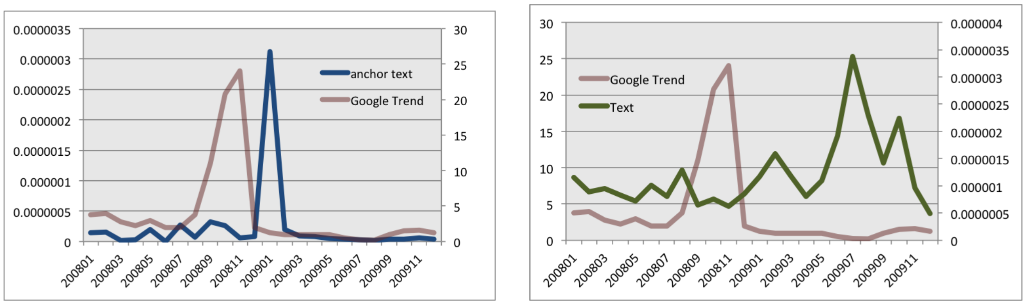


Figure 5.3: Correlation between time series mined from *anchor text* (left, $ccf = 0.69$, $\tau_{delay} = 2$), *content* (right, $ccf = 0.68$, $\tau_{delay} = 9$) to Google Trend for query electoral college

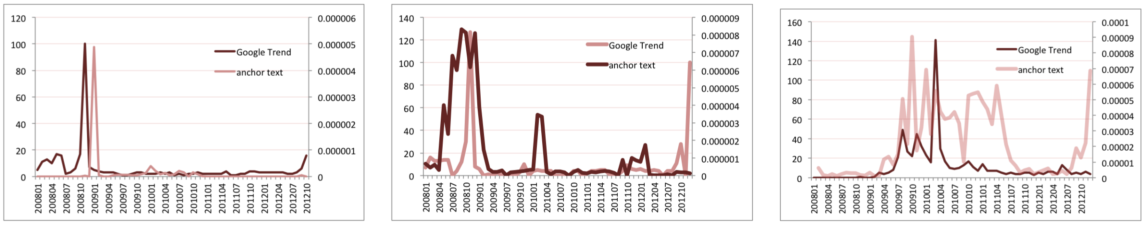


Figure 5.4: Time series of popular vote ($ccf = 0.94$, $\tau_{delay} = 2$), border fence ($ccf = 0.40$,$\tau_{delay} = 1$) and heath care reform ($ccf = 0.44$, $\tau_{delay} = 2$) from *anchor text* and Google Trend from left to right

shows that in the real query log, the subtopic *late-term abortion* get bursted starting from April 2009, and peaked in June 2009. While looking at the crawling time, it starts getting trend from May 2009 to September 2009, peaked in August 2009. However, investigating on the documents crawled in August 2009, we empirically found out that mostly they were published before and only being re-crawled or late-crawled till then.

Figure 5.6 shows the temporal dynamics of the subtopics underlined the query abortion. We see a clear alignment in peaks of the subtopic *late-term abortion* with the main query abortion in April 2012 (according to crawling time). It shows that there is temporal correlation in trending of both at a time-point but it is unsure when it is due to the time uncertainty (lagging) in the web archive.

We present another studied query, heath care in this analysis. Figure 5.7 depicts the temporal dynamics of the subtopics underlined *heath care* over the 2008-2012 period. Even though *health care* is a broad topic and being discussed all over again, its subtopics however are time-sensitive and trended at certain time-points. Although the crawling time does not provide any time-certainty but it can capture the dynamics of such subtopics, as shown in Section 5.1.5. Figure 5.8 then illustrates the development of the subtopic *health care reform* with two different time sources, crawling time and the strong confidence. Interestingly, both *crawling time* and the query log become bursty in January 2010. However, a deeper look into the development of the subtopic provided by the reliable time source show that the subtopic is already on trend 2 months earlier. Hence, both the real query log and the crawling time fail to detect the right relevant time for the subtopic. The lagging in the query log can be intuitively understood that the topic has been emerged and discussed in the .gov domain before it receives public attention.

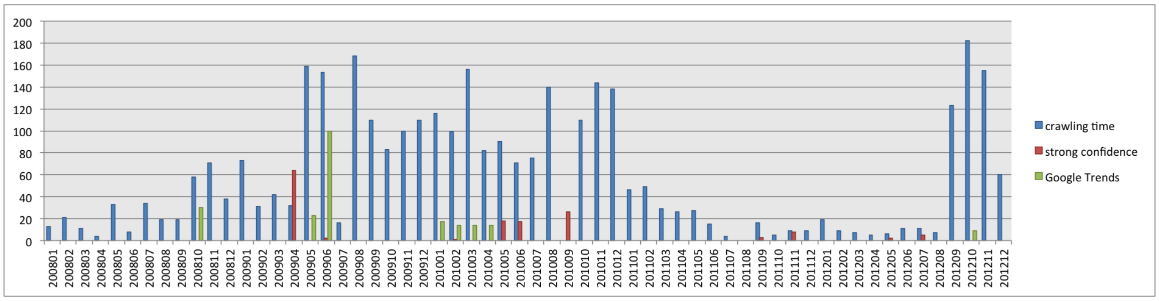


Figure 5.5: The temporal dynamics - reflexed by the accumulated document frequency of the subtopic late-term abortion from two different time reliability sources (crawling time and strong confidence) and from Google Trend.

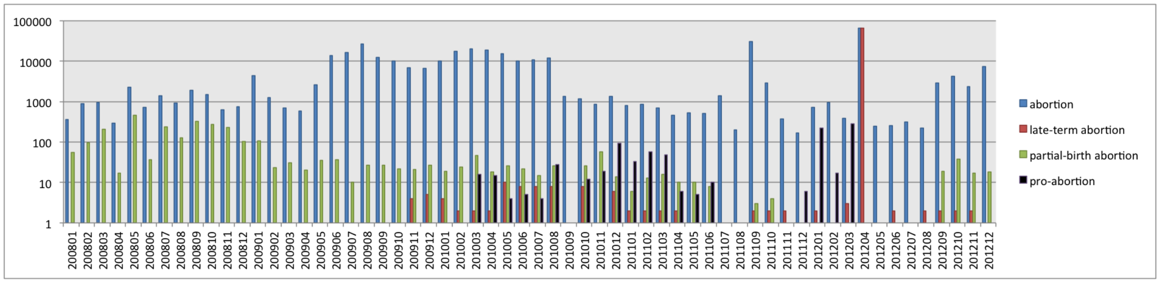


Figure 5.6: The temporal dynamics of the query abortion and its subtopics over time - reflexed by the accumulated frequency of anchor texts.

Inferring date for the temporal subtopics

This section provides some insights on determining the relevant time points for the temporal subtopics (mined by the temporal anchor texts), using our methods described in Section 5.1.4. Table 5.2 shows the temporal subtopic mining for 3 queries: *abortion*, *border fence* and *health care*. For each subtopic, we also show its corresponding temporal dynamics in Google Trend. For the subtopics of *border fence* the graphs are omitted due to the insufficiency of search volume. We can see that all the subtopics represented show a strong degree of burstiness and hence indicate their time sensitivity. However, identifying these time-points based solely on the timestamp annotations provided by the crawlers is difficult due to the natural lagging of the web archives. Our method that infers the relevant time periods by leveraging the part with strong level of time confidence is shown to be an effective indicator to solve the problem.

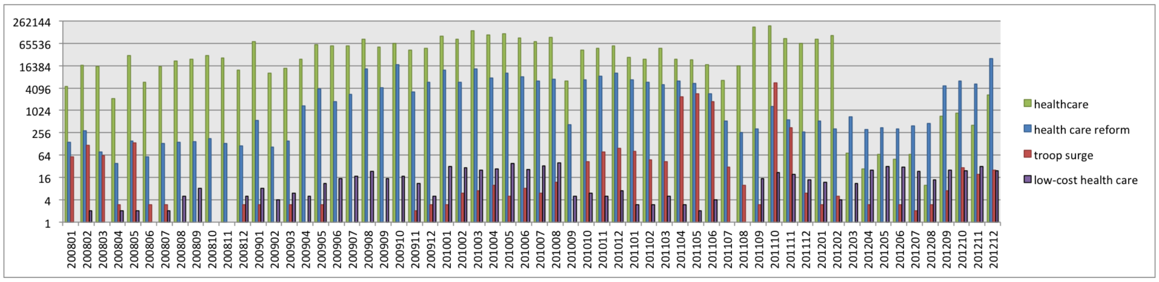


Figure 5.7: The temporal dynamics of the query health care reform and its subtopics over time - reflexed by the accumulated frequency of anchor texts.

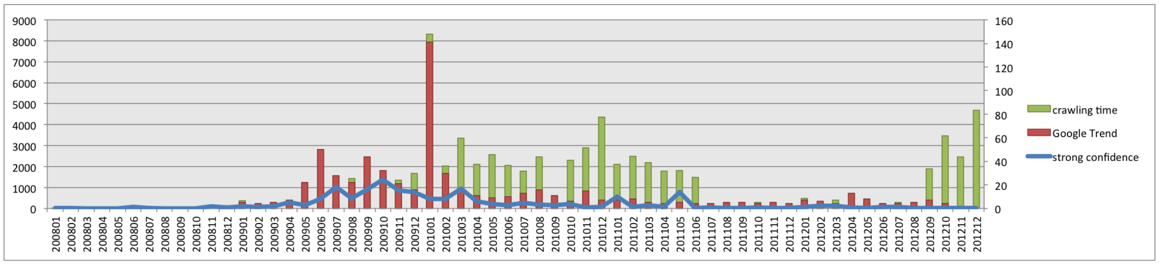


Figure 5.8: The temporal dynamics - reflexed by the accumulated document frequency of the subtopic health care reform from two different time reliability sources (crawling time and strong confidence) and from Google Trend.

5.1.6 Conclusions

In this section, we have studied the problem of mining temporal subtopics in the web archive. In future work, we will extend it to the time-aware search result diversification task in the web archive context. A further interesting problem is detecting the underlying 'topic drift' in this huge longitudinal of multi-modal data collection.

5.2 Temporal Graph-based Ranking

Web archives reflect nearly all types of social cultural, societal and everyday processes of our lives in the web as well as the exponential growth and continuous change in content and structure of the world wide web. Therefore, web archives from organizations such as the Internet Archive have the potential of becoming invaluable gold-mines for temporal content analytics of many kinds (e.g., politics and social issues, economics or media). First hand evidences about such processes are of great benefit for expert users such as journalists, economists, or historians. However, support for navigational search as it is, for example, offered by the Wayback machine[3], is not sufficient for tapping the full potential of web archives. Instead, search results should provide a good coverage of the query topic over time for enabling exploration of the topic and its evolution. Therefore, content relevance is

[3] http://archive.org/web/

not the only driver: time relevance and impact are other key factors. Further aspects, which make web archive search very different from web search are the high redundancy (pages of near-identical content are crawled all over again) and the special role that time/crawling time is playing in the web archive structure.

In this section, we tackle the problem of discovering important documents along the time-span of the web archives by a ranking approach. The intuition is that the impact/authority of a document in the web archives with regards to a query is strongly influenced by time. Hence, the temporal authority of a document should be accumulated over a surrounding time window (instead of considering only one or all temporal snapshots).

Temporal link analysis for web search improvement has been studied in previous work [YLL, YQZ$^+$, DDa, BVW05]. Their common goal is to improve state-of-the-art link-based algorithms (i.e., PageRank [PBMW99]) in favoring new web pages, instead of old, stale pages (that often ranked high due to its accumulated in-links). The most advanced approach in this direction is described in [DDa], where they track the authority of a page over multiple historical (past) web snapshots. This allows the incorporation of web freshness into authority propagation, and hence, boosts the authority of fresh (new) page at the querying time. In estimating the temporal authority along the time dimension, we adapt their strategy by propagating the authority of *past* and *future* snapshots. This backward propagation (from future snapshots) accounts for the 'lagging' time a new document needs to gather in-links. For example, a document about health care reform issued in March, 2010 is more relevant with the time point than April, 2010, where it has more number of in-links.

Graph-based diversity for ranking based on random walk are addressed in [ZGVGA, CDG$^+$13, MGR]. The first two utilized a greedy algorithm in transforming a picked node into an absorbing state. This punishes neighboring nodes but the random walk still lingers at away nodes, hence increases diversity. Mei et al. [MGR] introduce a vertex- reinforcement base on the intuition that nodes are visited many times tend to be more likely to be re-visited. This reinforcement on the transition probability has a strong theoretical foundation, brings less complexity and can be improved towards scalability in large graphs.

For temporal ranking in the web archive, we are the first ones to combine the problems of relevance, temporal authority, diversity and time in a unified framework. In more detail, we construct a temporal graph over the web archive. Using time preference and relevance as priors, in Section 5.2.1, we propose a novel random walk model on the temporal graph. The model accounts for the in-link and natural time lagging in the web archive in mining the temporal authority. We present two ways of injecting time preference in Section 5.2.1. Further more, we introduce a novel diversity mechanism that penalizes both neighbors in the same web snapshots and *across* snapshots in Section 5.2.1. Our experiments are conducted on a large-scale, real-world web archival dataset, which is further explained in Section 5.2.2.

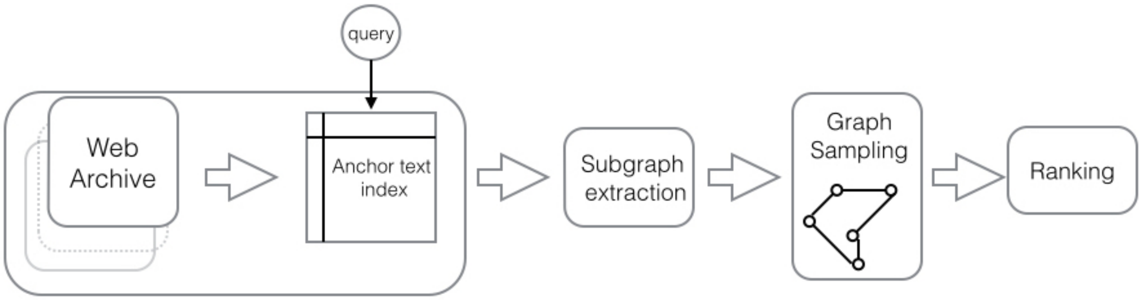


Figure 5.9: The graph-based ranking system pipeline.

5.2.1 Methodology

We demonstrate a possible system pipeline for the proposed ranking approach in Figure 5.9. To shrink down the large-scale graph, two possible components i.e., (1) Sub-graph extraction and (2) Graph sampling can be plugged into the pipeline. We will detail (1) in Section 4.1.3. For (2), a possible technique is describe in the following section.

Graph Sampling

Graph sampling for temporal graph is a complex task as we need to take into account the temporal structure as well as graph properties at different times. Leskovec et al. [LF06] propose a sampling method that attempts to match the graph temporal evolution. As described in [LF06], the *Back-in-time* sampling goal corresponds to traveling back in time and trying to mimic the past versions of graph G. Let G_n denote graph G at some point in time, when it had exactly n nodes. Now, we want to find a sample S on n nodes that is most similar to graph G_n , i.e. when graph G was of the same size as S. The challenge lies in the fact that the goal is trying to match patterns describing the temporal evolution together with the patterns defined on a single snapshot of a graph, which also change over time. If one would have node ages, then the best possible approach would be to simply roll-back the evolution (addition/deletion of nodes and edges over time).

Fire Forest Sampling. First, we give the exact definition and the algorithm for the Forest Fire sampling [LKF07]. We first choose node v uniformly at random. We then generate a random number x that is geometrically distributed with mean $p_f/(1p_f)$. Node v selects x out-links incident to nodes that were not yet visited. Let $w_1, w_2, ..., w_x$ denote the other ends of these selected links. We then apply this step recursively to each of $w_1, w_2, ..., w_x$ until enough nodes have been burned. As the process continues, nodes cannot be visited a second time, preventing the construction from cycling. If the fire dies, then we restart it, i.e. select new node v uniformly at random. We call the parameter p_f the forward burning probability.

PageRank

The PageRank algorithm is proposed by Page et al. [PBMW99], that is based on the random walk Markov process. At each step, the surfer either jumps to an arbitrary node or follows one of the out-going edges of the current node. Formally, the random walk process of PageRank can be defined as:

$$\pi_t = \alpha P_T \pi_{t-1} + (1-\alpha)v \tag{5.4}$$

where P is a transition matrix, π_t is the importance score vector of all pages at step t, and α is the damping factor, which controls how often the surfer jumps to an arbitrary node, v is a uniform distribution. The score of a node p at step t_i can also be represented as:

$$\pi_{p,i} = \sum_{q:q \to p} P(q,p) \cdot \pi_{p,i-1} \tag{5.5}$$

whereas the transition probability P(q,p) can be estimated by

$$P(q,p) = \begin{cases} (1-\alpha)\frac{1}{N} + \alpha\frac{1}{deg(q)} & if deg(q) > 0, \\ \frac{1}{N} & otherwise. \end{cases} \tag{5.6}$$

When one considers the non-uniform distribution of the ‘following out-egdes’ probability (denoted as $t(q,p)$, and jump probabilities $s(p)$, the formula 5.6 is generalized as:

$$P(q,p) = \begin{cases} (1-\alpha) \cdot s(p) + \alpha \cdot t(q,p) & if deg(q) > 0, \\ s(p) & otherwise. \end{cases} \tag{5.7}$$

Temporal Ranking Model

In this work, we re-design the traditional PageRank (at document/page level) to more precisely measure the temporal authority of the documents. We propose a new time-aware random surfer model, with the intuition that instead of jumping to a random node with equal probability, this traveler favors jumping to a node at a time period of interest.

Temporal Graph Model A temporal graph $\mathscr{G}$ consists of multiple graphs at different time points, called graph snapshots (snapshots for short). A snapshot G is a directed graph with time annotation, $G_t = (V_t, E_t, t)$, with $t \in T$, $V_t \in V$ and $E_t \in E$. A vertex $v \in V$ can belong to multiple snapshots $\{V_t\}$. A vertex $v \in V_{t_i}$ is connected with $v \in V_{t_j}$ by an inter-link $\{v_{t_i}, v_{t_j}\} \in \mathscr{I}$. In the web archive context, each vertex v in a graph snapshot is a revision of a document d, identified by an unique URL. The vertex is time-stamped by the crawling time. The edge between two vertices is the hyper-link between two revisions. It is time-stamped as the time of the source vertex.

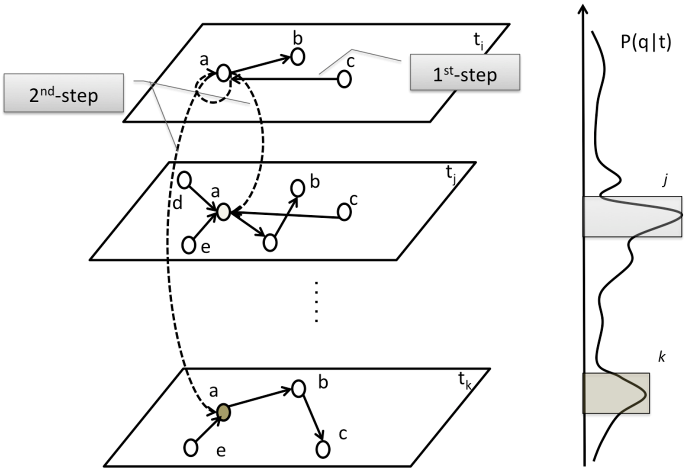


Figure 5.10: Time-travel web archive surfer. Solid lines indicate *within*-snapshot transitions, and dashed lines indicate the *across*-snapshot teleporting

Temporal Random Surfer Model We describe a 'time travel random surfer model', which redefines how an web archive searcher (so-called *time traveler*) surfs in the web archive. The 'time travel random surfer model' is initiated by the 'random surfer model', which explains underlying PageRank [PBMW99]. The original model describes the surfing behavior of a web surfer that after following the link structure starting from a page for several steps and then jump to a random page. However, this surfer model does not well-captured the temporal nature of a longitudinal web archive. Surfing in the web archive, we assume that a user prefers search results from interesting time points. Hence, the surfing behaviour of a time traveler should be adapted to incorporate this temporally important aspect.

In this work, we model a time-travel surfing as in moves that consist of two distinct steps (i.e., non-temporal and temporal, as illustrated in Figure 5.10). At each move, a traveler starts with a *non-temporal* step. A searcher chooses to either follow the out-links of a page or jump to a remote page. In order to achieve the precise temporal authority of a page at a time point, we employ the fresh favoring mechanism as in [DDa], so that old pages are degraded. A traveler in our model prefers a new/fresh page. As contrast to [DDa], we do not consider the editorial behaviour of the page because we focus on *informational* queries, where the content of a document is nearly *static* over time. In order that, we take into account the freshness linked with the age of a document a at time t_i, $\mathscr{F}(a,t_i)$, which is quantified as:

$$\mathscr{F}(a,t_i) = e^{-\beta_1(t_i - \mathscr{T}(a))} \tag{5.8}$$

where $\mathscr{T}$ denotes the first time a page appears in the collection. In the second (temporal) step, a traveler jumps to a snapshot of the page at different time points. To capture the temporal authority, we model the authority propagation flow among nodes at nearby time points. Here, we introduce two kinds of propagations: *forward-* (authority is propagated by past snapshots) and *backward*- (authority is propagated by future snapshots). The propagation is modeled in a decay fashion, and hence, in a time window with length controlled by a decay parameter. This authority propagation is different from [DDa] in the sense that, it helps capturing the precise temporal authority of a document at a given time point (instead of using for smoothing purpose). We model the length of the temporal propagation as controlled by the decay parameter β_2 (further explained in Section 5.2.1), that we model as a global parameter in this work.

Time-sensitive PageRank

In this section, we explain two novel methods to inject time preference into the PageRank: (1) the jumping probability at the 1^{st} step, so that the jumping scope is not restricted to within current snapshot but other snapshots and (2) via the transition probability between snapshots, at the 2^{nd} step.

In the normal case where no preferences are defined, the vector $\vec{v}$ which presents the jumping probability from node a to all the nodes N in the temporal graph is uniformly distributed ($= [\frac{1}{N}]N \times 1$). However, different from this ordinary behavior, we present a *query-dependent*, time-aware vector $\vec{v}_{temp}$ over the temporal graph as follows:

Time-aware Teleportation Instead of limiting the jumping scope in within a web snapshot, the traveler in this case can jump to any snapshot with a time preference. The probability of jumping from q to p at time t_i, $P_{t_i}(p|q, Jump)$, is dependent on the preference score of time t_i, $I(t_i)$. The probability that a traveler reaches the page p at snapshot t_i can be written as[4]:

$$\begin{aligned}\pi_{p,i} = \sum_{t_j \in T_i} P_{t_i|t_j}(p) \sum_{q:q \to p|t_j} P_{t_j}(Follow|q) P_{t_j}(p|q, Follow) \\ + \sum_{\forall q} P(Jump|q) I(t_i)\end{aligned} \tag{5.9}$$

Time-aware Transition Probability For this second type of embedding time preference into the model, we modify the transition probability across time snapshots. Intuitively, a snapshot at time t_i with high time preference will have higher transition probability. In this case, the jumping scope is restricted within the time snapshot. The probability that a

[4] One can introduce a parameter α in the formula so that $\sum_p \pi_{p,i} = 1$. However, to simplify the problem, we omit the parameter.

traveler reaches the page p at snapshot t_i can be written as:

$$\begin{aligned}\pi_{p,i} &= \begin{array}{l}\displaystyle\sum_{t_j\in S_i} P_{t_i|t_j}(p) \sum_{q:q\to p|t_j} P_{t_j}(Follow|q)P_{t_j}(p|q,Follow) \\ \displaystyle\qquad + \sum_{t_j\in S_i} P_{t_i|t_j}(p)\sum_{q|t_j} P_{t_j}(Jump|q)P_{t_j}(p|q,Jump)\end{array} \\ &= \begin{array}{l}\displaystyle\sum_{t_j\in S_i} P_{t_i|t_j}(p) \\ \displaystyle\cdot\left[(1-\alpha)\sum_{q:q\to p|t_j} F_{t_j}(p,q)\cdot\pi_{q,j} + \alpha\sum_{q|t_j}\frac{\pi_{q,j}}{N_{t_j}}\right]\end{array}\end{aligned}$$

where S_i is the set of snapshots which can directly distribute authority to t_i within one step. Even though presenting a similar generalization of the propagation model to us, the results in [DDa] indicate that the decay propagation is not most suitable for their task (normal web ranking). In our case, instead this transition probability (propagation) is strongly time-influenced. A node most propagates its authority to the nearest time of interest (or peak time). The transition probability $P_{t_i|t_j}(p)$ is derived from the interestingness measure of the two time points and is calculated as:

$$P_{t_i|t_j}(p) = \frac{I(t_j)}{\sum_{\forall t_k} I(t_k)} \cdot w(t_i,t_j) \tag{5.11}$$

where $w(t_i,t_j) = e^{\beta_2|t_i-t_j|}$. Hence the propagation scope is restricted to a time window $\mathscr{W}$ with the size (also size of S) is controlled by β_2. Within the time window $\mathscr{W}$, one with high time preference will be more likely to be propagated.

Time-based Diversity in Temporal Graph

In this section, we target another issue of the ranking problem, the time-based diversification of the top-k results.

Reinforcement in Random Walk

Mei et al. [MGR] introduce the integration of the vertex-reinforced random walk (VRRW) into the conventional PageRank to address the diversity ranking in graphs. Their intuition follows the ‘rich gets richer’ phenomenon, which specifically, the node that has been visited many times will have higher probability to be revisited again. Hence, the transition probability in the Markov random walk (to a state from others) is reinforced by the number of previous visits to that state. Our time-based diversity model follows the same intuition. The vertex reinforcement is applied within each snapshot, so that the *within*-snapshot neighbors of a popular node (visited many times) are penalized. For the authority propagation across time snapshots, however, this mechanism cannot be integrated directly. Instead, we follow a *voting* propagation mechanism. For every step, we check for the node snapshot with maximum number of visits over the propagation time window, and only this node snapshot got propagated from others. The other nodes receive no propagation from other nodes. Hence, this approach allows a partial *across*-snapshot penalty and helps the time-based diversity.

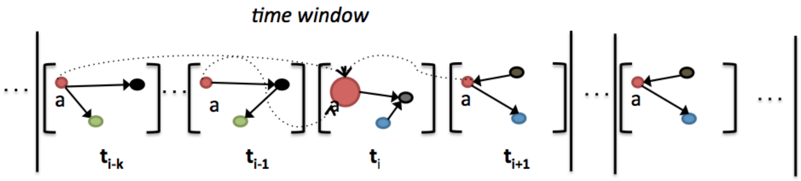


Figure 5.11: Reinforced/dynamic propagation over a time window. There is only one node gets all the propagations.

In a similar fashion to [MGR], the time-variant transition probability from q to p at step T (of the random walk) within a time snapshot t_i is defined as:

$$P^T_{t_i}(p,q) = (1-\alpha)\cdot s(p_{t_j}) + \alpha \cdot \frac{P^0_{t_i}(p,q)\cdot N^T_{t_i}(q)}{D^T_{t_i}(p)} \tag{5.12}$$

where $N^T_{t_i}(q)$ is the number of visits to q at step T. $D^T_{t_i}(p) = \sum_{q\in t_i} P^0_{t_i}(p,q)N^T_{t_i}(q)$. The cross snapshot- transition probability $P_{t_i|t_j}(p)$ at step T is:

$$P_{t_i|t_j}(p) = \begin{cases} \text{calculated as Equation 5.11} & if \\ \qquad p = \text{argmax}_{\forall t_i\in \mathscr{W}_{t_j}} N^T_{t_i}(p), & \\ 0 & otherwise. \end{cases} \tag{5.13}$$

where $\mathscr{W}_{t_j}$ is the time window of t_j, such that $\forall t_i \in \mathscr{W}_{t_j}$, $w(t_i,t_j) > 0$.

Zero Out-Link Problem

In the web archive context, the problem of having nodes with zero out-link is even more severe (than the whole web). Not only it is hard to crawl the whole graph but also the web archive is often domain-centric (e.g., .DE, .UK and .GOV). Thus, a substantial amount of a graph nodes are external nodes (specifically not being crawled, but is referred to by an internal node). This so-called problem of "dangling nodes" is addressed in a number of related work [PBMW99, DHH$^+$, BGS05, IS07]. In this work, we follow the approach that is proved to be most effective in [BGS05], such that:

$$\bar{M} = (1-\alpha)\cdot M + \bar{\alpha}\cdot U \tag{5.14}$$

where

$$\bar{\alpha}(i) = \begin{cases} \alpha & if \sum_j M_{ij} = 1, \\ 1 & otherwise. \end{cases} \tag{5.15}$$

The probability that a traveler reaches the page p at snapshot t_i can be written as [5]:

$$\pi_{p,i} = \sum_{t_j \in T_i} P_{t_i|t_j}(p) \sum_{q:q\rightarrow p|t_j} P_{t_j}(Follow|q)P_{t_j}(p|q,Follow) + \sum_{\forall q} P_{t_i}(Jump|q)P_{t_i}(p|q,Jump) \tag{5.16}$$

5.2.2 Experiments

In this section, we evaluate the performances of 5 different methods: temp-BM25, temp-PageRank[6](baseline), our approach with time-aware teleportation (ours), our approach with vertex-reinforcement random walk (ours+div) and our approach with time-aware vertex-reinforcement random walk (ours+tempdiv).

Experiment settings

Dataset We utilize a corpus of archival web pages in .gov domain collected by the Internet Archive from January 1995 to September 2013. The corpus contains over 900 million of text captures and over 58.8 billion temporal links. In order to shrink down the huge collection to extract a subset/sub-graph of interest, we follow the idea of recent work that exploiting the value of anchor text. First, we achieve over 60 related long-term controversial political topics from the debate website[7]. We then look into the document linking graph and extract the links with anchor text reflecting any of the topics (both lexicographically and semantically). We then captured the source and destination web pages of these links and treat them as the document seeds. We further capture the in-pages pointing to source pages and out-pages (which the destinations point to) to achieve a substantially large collection of over 100 million document revisions (approx. 40 million unique documents). We then picked 20 controversial/informational queries to conduct the experiment.

Ground Truth and Metrics Since there is no publicly available gold standard for our work, we rely on manual annotation for the 20 queries. For each document, we asked human experts, whether it is relevant to (1) one of the time-points and (2) any of the content-based subtopics. The scale for time is binary, whereas to account for authority, we use a scale from 0 (not related) to 4 (highly relevant) for the subtopic relevance judgment. A document with score larger than 2 is considered as relevant. For evaluation, follow the adopted setting of recent TREC web tracks (diversity task) as presented in [NK14], we use a generalization of a well-known result diversification metric (that accounts for both diversity and relevance), α-*nDCG* [CKC$^+$] (so that it takes into account the subtopic weights). We

[5] one can linearly introduce a parameter α in the formula so that $\sum_p \pi_{p,i} = 1$. However, to simplify the problem, we skip tuning the parameter.

[6] To incorporate temporal prior, we apply PageRank and BM25 at each time snapshot and then multiply them with the corresponding time preference score.

[7] http://www.debate.org/big-issues/

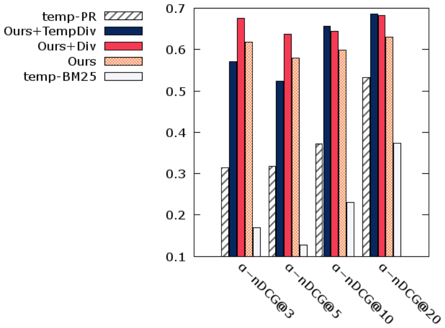


Figure 5.12: Performance of time-based subtopic diversity.

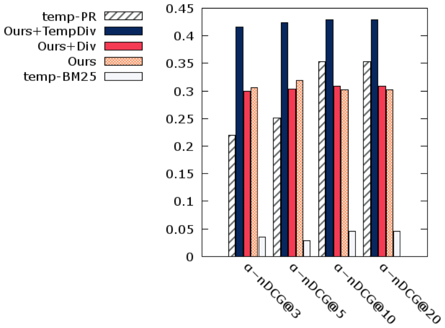


Figure 5.13: Performance of content-based subtopic diversity.

consider two different subtopic dimensions, (1) *time* - with associated weight mined from the anchor-text distribution and (2) subtopics mined from *content* - equally weighted.

Priors and Parameter Tuning For the temporal prior, we mined from the anchor text frequency distribution (to mimic the query logs, following [NKNN]). For the relevance prior, we utilize the scores from the BM25 retrieval model. For the random walk parameters, we set the jumping probability to be the default 0.15. All decay parameters are set to 0.4. All algorithms are run over Apache Giraph[8].

Experiment results

Figures 5.12 and 5.13 show the *time* and *subtopic* diversification results of the compared models respectively. For the *time* diversity, our models outperform the baseline significantly ($p < 0.05$) at $k = 3$ and $k = 5$. It is empirically found that, our propagation method helps identifying the temporal authority with regards to the relevant time more precisely. For example, given the query electoral college and a time period February 2009, a document issued in September 2008 has a high score for the traditional PageRank. However, our propagation accounts for both *freshness* and *lagging* ranks another document issued in February 2009 higher (that is more time relevant). The good performance of our approach

[8] http://giraph.apache.org/

shows that we capture better the temporal authority of documents with regards to the time preference. Our temporal diversity method also shows that it diversifies time effectively. The results for *content*-based subtopic diversity measurement also indicate a good performance of our method. Ours+tempdiv best performs (significant with $p < 0.05$) for all cases. This rather shows the effectiveness of our time-aware diversity approach.

5.2.3 Conclusions

For the work in this chapter, we have studied the problem of finding important document in the web archive and address it in a unified framework that integrates relevance, temporal authority, diversity and time together. In detail, we proposed a novel random walk model incorporating time and the link structure of the web archive. Our model is shown to outperform PageRank for both relevance and diversity tasks. For future work, we would like to investigate the application of the novel model on different open challenges of web archive, i.e., time-aware summarization. Scalability issues will also be another target.

Table 5.2: Examples of the subtopics extraction, weighting and time-relevant for 3 queries *abortion*, *border fence* (graphs not available) and *health care*.

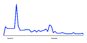
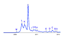
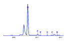
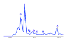
Query	Original anchor text	Subtopic	Weight	Most relevant time
abortion	Arizona governor signs law banning most late-term abortions	late-term abortion	0.15	2009-04
	Prohibit partial-birth abortion bill	partial-birth abortion	0.03	2011-04 
	Republicans attack Obama ahead of vote on bill to punish sex-selective abortion	sex-selective abortion	0.02	2010-02
border fence	Bush signs border fence funding into law	border fence funding	0.12	2008-03
	Construction of the Mexico border fence	border fence mexico	0.02	2006-10
	Five years ago, legislation was passed to build a 700-mile double-layer border fence along the southwest border	double-layer border fence	0.07	2011-11
health care	Obama promotes health care reform - at a grocery store	health care reform	0.13	2010-03 
	The health care vote	health care vote	0.11	2010-03 
	Freshmen propose health care amendments	health care amendment	0.03	2010-03 

Table 5.3: Statistics for date extraction method over revisions

Level	Number of revisions	Percentage
Very strong	1271124	1.23
Strong	12696686	12.22
Mildly strong	1173490	1.13
Acceptable	14179469	13.65
Weakly acceptable	74546697	71.35
Crawling time	435785	0.42

Table 5.4: Statistics for date extraction method over documents

Level	Number of documents	Percentage
Very strong	139426	0.48
Strong	2968054	10.32
Mildly strong	258591	0.89
Acceptable	5264909	18.32
Weakly acceptable	20102819	69.49
Crawling time	134852	0.46

6 Social network and Clinical Domain Studies

6.1 Enhancing temporal model performance in social media

Widely spreading rumors can be harmful to the government, markets and society and reduce the usefulness of social media channel such as Twitter by affecting the reliability of their content. Therefore, effective method for detecting rumors on Twitter are crucial and rumors should be detected as early as possible before they widely spread. As an example, let us recall of the shooting incident that happened in the vicinity of the Olympia shopping mall, Munich; in a summer day, 2016. Due to the unclear situation at early time, numerous rumors about the event did appear and they started to circulate very fast over social media. The city police had to warn the population to refrain from spreading related news on Twitter as it was getting out of control: *"Rumors are wildfires that are difficult to put out and traditional news sources or official channels, such as police departments, subsequently struggle to communicate verified information to the public, as it gets lost under the flurry of false information."* [1] Figure 6.1 shows the rumor *sub-events* in the early stage of the event Munich shooting. The first *terror-indicating* "news" –The gunman shouted 'Allahu Akbar'– was widely disseminated on Twitter right after the incident by an unverified account. Later the claim of three gunmen also spread quickly and caused public tension. In the end, all three information items were falsified.

We follow the rumor definition [QRRM11] considering a rumor (or fake news) as a statement whose truth value is unverified or deliberately false. A wide variety of features has been used in existing work in rumor detection such as [CMP11, GKCM14, JDS$^+$13, LNL$^+$15, MGM$^+$, MGW$^+$15, MPC10, WYZ15, YLYY12]. Network-oriented and other aggregating features such as propagation pattern have proven to be effective for this task. Unfortunately, the inherently accumulating characteristic of such features, which require some time (and Twitter traffic) to mature, does not make them very apt for early rumor detection. A first semi-automatic approach focussing on early rumor detection presented

[1] Deutsche Welle: http://bit.ly/2qZuxCN

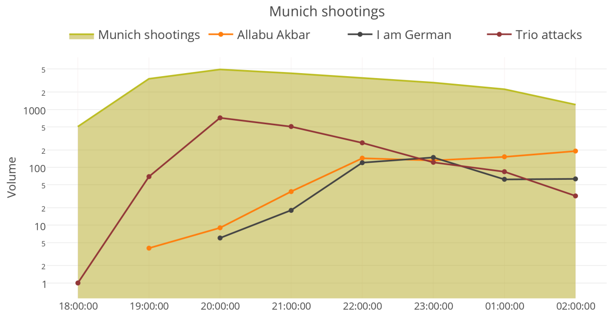


Figure 6.1: The *Munich shooting* and its sub-events burst after the first 8 hours, y-axis is English tweet volume.

by Zhao et al. [ZRM15], thus, exploits rumor signals such as enquiries that might already arise at an early stage. Our fully automatic, cascading rumor detection method follows the idea on focusing on early rumor signals on text contents; which is the most reliable source before the rumors widely spread. Specifically, we learn a more complex representation of single tweets using Convolutional Neural Networks, that could capture more hidden meaningful signal than only enquiries to debunk rumors. [CWL^{+}17, MGM^{+}] also use RNN for rumor debunking. However, in their work, RNN is used at *event-level*. The classification leverages only the deep data representations of aggregated tweet contents of the whole event, while ignoring exploiting other –in latter stage–effective features such as user-based features and propagation features. Although, tweet contents are merely the only reliable source of clue at early stage, they are also likely to have doubtful perspectives and different stands in this specific moment. In addition, they could relate to rumorous sub-events (see e.g., the Munich shooting). Aggregating all relevant tweets of the event at this point can be of noisy and harm the classification performance. One could think of a sub-event detection mechanism as a solution, however, detecting sub-events at real-time over Twitter stream is a challenging task [MNR^{+}15], which increases latency and complexity. In this section, we address this issue by deep neural modeling only at single tweet level. Our intuition is to leverage the "wisdom of the crowd" theory; such that even a certain portion of tweets at a moment (mostly early stage) are weakly predicted (because of these noisy factors), the ensemble of them would attribute to a stronger prediction.

In this section, we make the following contributions with respect to rumor detection:

- We develop a machine learning approach for modeling tweet-level credibility. Our CNN-based model reaches 81% accuracy for this novel task, that is even hard for human judgment. The results are used to debunk rumors in an ensemble fashion.

- Based on the credibility model we develop a novel and effective cascaded model for rumor classification. The model uses time-series structure of features to capture their temporal dynamics. Our model clearly outperforms strong baselines, especially for the targeted early stage of the diffusion. It already reaches over 80% accuracy in the first hour going up to over 90% accuracy over time.

6.1.1 Related Work

A variety of issues have been investigated using data, structural information, and the dynamics of the microblogging platform Twitter including event detection [Kim15], spam detection [AA12, Wan10], or sentiment detection [BF10]. Work on rumor detection in Twitter is less deeply researched so far, although rumors and their spreading have already been investigated for a long time in psychology [AP47, BHM12, Sun14]. Castillo et al. researched the information credibility on Twitter[CMP11, GKCM14]. The work, however, is based solely on people's attitude (trustful or not) to a tweet not the credibility of the tweet itself. In other words, a false rumor tweet can be trusted by a reader, but it might anyway contain false information. The work still provides a good start of researching rumor detection.

Due to the importance of information propagation for rumors and their detection, there are also different simulation studies [SMA12, TBM10] about rumor propagations on Twitter. Those works provide relevant insights, but such simulations cannot fully reflect the complexity of real networks. Furthermore, there are recent work on propagation modeling based on epidemiological methods [BYXD13, JDS$^+$13, KCJ$^+$13], yet over a long studied time, hence how the propagation patterns perform at early stage is unclear. Recently, [WYZ15] use unique features of Sina Weibo to study the propagation patterns and achieve good results. Unfortunately Twitter does not give such details of the propagation process as Weibo, so these work cannot be fully applied to Twitter.

Most relevant for our work is the work presented in [MGW$^+$15], where a time series model to capture the time-based variation of social-content features is used. We build upon the idea of their *Series-Time Structure*, when building our approach for early rumor detection with our extended dataset, and we provide a deep analysis on the wide range of features change during diffusion time. Ma et al. [MGM$^+$] used Recurrent Neural Networks for rumor detection, they batch tweets into time intervals and model the time series as a RNN sequence. Without any other handcrafted features, they got almost 90% accuracy for events reported in Snope.com. As the same disadvantage of all other deep learning models, the process of learning is a black box, so we cannot envisage the cause of the good performance based only on content features. The model performance is also dependent on the tweet retrieval mechanism, of which quality is uncertain for stream-based trending sub-events.

6.1.2 Tweet-level Credibility Model

Before presenting our Tweet-level Credibility Model, we will start with an overview of our overall rumor detection method. The processing pipeline of our classification approach is shown in Figure 6.2. In the first step, relevant tweets for an event are gathered. Subsequently, in the upper part of the pipeline, we predict tweet credibilty with our pre-trained

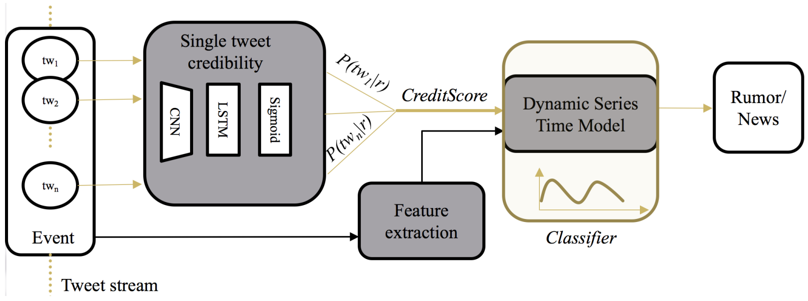


Figure 6.2: Pipeline of our rumor detection approach.

credibility model and aggregate the prediction probabilities on single tweets (CreditScore). In the lower part of the pipeline, we extract features from tweets and combine them with the creditscore to construct the feature vector in a time series structure called Dynamic Series Time Model. These feature vectors are used to train the classifier for rumor vs. (non-rumor) news classification.

Early in an event, the related tweet volume is scanty and there are no clear propagation pattern yet. For the credibility model we, therefore, leverage the signals derived from tweet contents. Related work often uses aggregated content [LNL$^+$15, MGW$^+$15, ZRM15], since individual tweets are often too short and contain slender context to draw a conclusion. However, content aggregation is problematic for hierarchical events and especially at early stage, in which tweets are likely to convey doubtful and contradictory perspectives. Thus, a mechanism for carefully considering the 'vote' for individual tweets is required. In this work, we overcome the restrictions (e.g., semantic sparsity) of traditional text representation methods (e.g., bag of words) in handling short text by learning low-dimensional tweet embeddings. In this way, we achieve a rich hidden semantic representation for a more effective classification.

Exploiting Convolutional and Recurrent Neural Networks

Given a tweet, our task is to classify whether it is associated with either a news or rumor. Most of the previous work [CMP11, GKCM14] on tweet level only aims to measure the *trustfulness* based on human judgment (note that even if a tweet is trusted, it could anyway relate to a rumor). Our task is, to a point, a reverse engineering task; to measure the probability a tweet refers to a *news* or *rumor* event; which is even trickier. We hence, consider this a weak learning process. Inspired by [ZSLL15], we combine CNN and RNN into a unified model for tweet representation and classification. The model utilizes CNN to extract a sequence of higher-level phrase representations, which are fed into a long short-term memory (LSTM) RNN to obtain the tweet representation. This model, called CNN+RNN henceforth, is able to capture both local features of phrases (by CNN) as well as global and temporal tweet semantics (by LSTM)(see Figure 6.3).

Representing Tweets: Generic-purpose tweet embedding in [DZF$^+$16, VVR16] use

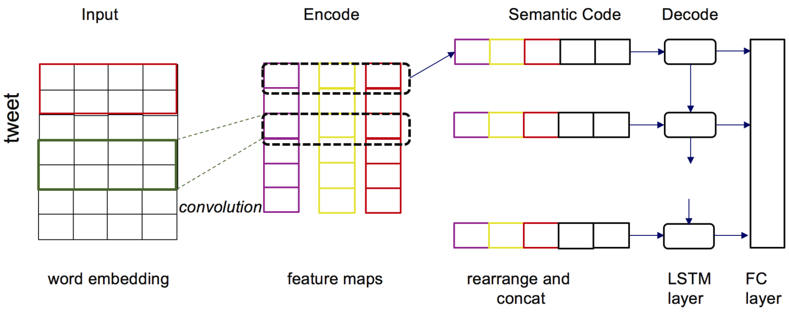


Figure 6.3: CNN+LSTM for tweet representation.

character-level RNN to represent tweets that in general, are noisy and of idiosyncratic nature. We discern that tweets for rumors detection are often triggered from professional sources. Hence, they are linguistically clean, making word-level embedding become useful. In this work, we do not use the pre-trained embedding (i.e., *word2vec*), but instead learn the word vectors from scratch from our (large) rumor/news-based tweet collection. The effectiveness of fine-tuning by learning task-specific word vectors is backed by [Kim14]. We represent tweets as follows: Let $x_i \in \mathscr{R}$ be the k-dimensional word vector corresponding to the i-th word in the tweet. A tweet of length n (padded where necessary) is represented as: $x_{1:n} = x_1 \oplus x_2 \oplus \cdots \oplus x_n$, where $\oplus$ is the concatenation operator. In general, let $x_{i:i+j}$ refer to the concatenation of words $x_i, x_{i+1}, ..., x_{i+j}$. A convolution operation involves a filter $w \in \mathscr{R}^{hk}$, which is applied to a window of h words to produce a feature. For example, a feature c_i is generated from a window of words $x_{i:i+h-1}$ by: $c_i = f(w \cdot x_{i:i+h-1} + b)$.

Here $b \in \mathscr{R}$ is a bias term and f is a non-linear function such as the hyperbolic tangent. This filter is applied to each possible window of words in the tweet $\{x_{1:h}, x_{2:h+1}, ..., x_{n-h+1:n}\}$ to produce a feature map: $c = [c_1, c_2, ..., c_{n-h+1}]$ with $c \in \mathscr{R}^{n-h+1}$. A max-over-time pooling or dynamic k-max pooling is often applied to feature maps after the convolution to select the most or the k-most important features. We also apply the 1D max pooling operation over the time-step dimension to obtain a fixed-length output.

Long Short-Term Memory Layer. RNNs are able to propagate historical information via a chain-like neural network architecture. While processing sequential data, it looks at the current input x_t as well as the previous output of hidden state h_{t-1} at each time step. The simple RNN hence has the ability to capture context information. However, the length of reachable context is often limited. The gradient tends to vanish or explode during the back propagation. With the memory cell in LSTMs [HS97], the gradient flow is continuous (errors maintain their value) which thus eliminates the vanishing gradient problem and enables learning from long sequences.

Optimization. We regard the output of the hidden state at the last step of LSTM as the final tweet representation and we add a softmax layer on top. We train the entire model by minimizing the cross-entropy error. Given a training tweet sample $x^{(i)}$, its true label $y_j^{(i)} \in \{y_{rumor}, y_{news}\}$ and the estimated probabilities $\tilde{y}_j^{(i)} \in [0..1]$ for each label $j \in \{rumor, news\}$, the error is defined as:

$$\mathsf{L}(x^{(i)}, y^{(i)}) = 1\{y^{(i)} = y_{rumor}\} log(\tilde{y}^{(i)}_{rumor}) + 1\{y^{(i)} = y_{news}\} log(\tilde{y}^{(i)}_{news}) \qquad (6.1)$$

where 1 is a function converts boolean values to $\{0,1\}$. We employ stochastic gradient descent (SGD) to learn the model parameters.

6.1.3 Time Series Rumor Detection Model

As observed in [MGM^{+}, MGW^{+}15], rumor features are very prone to change during an event's development. In order to capture these temporal variabilities, we adopt the Dynamic Series-Time Structure (DSTS) model [MGW^{+}15] (time series for short) for feature vector representation. We base our credibility feature on the time series approach and train the classifier with features from diffent high-level contexts (i.e., users, Twitter and propagation) in a cascaded manner. In this section, we first detail the employed Dynamic Series-Time Structure, then describe the high and low-level ensemble features used for learning in this pipeline step.

Temporal Model

For an event E_i we define a time frame given by $timeFirst_i$ as the start time of the event and $timeLast_i$ as the time of the last tweet of the event in the observation time. We split this event time frame into N intervals and associate each tweet to one of the intervals according to its creation time. Thus, we can generate a vector V(E_i) of features for each time interval. In order to capture the changes of feature over time, we model their differences between two time intervals. So the model of DSTS is represented as: $V(E_i) = (\mathbf{F}^D_{i,0}, \mathbf{F}^D_{i,1}, ..., \mathbf{F}^D_{i,N}, \mathbf{S}^D_{i,1}, ..., \mathbf{S}^D_{i,N})$, where $\mathbf{F}^D_{i,t}$ is the feature vector in time interval t of event E_i. $\mathbf{S}^D_{i,t}$ is the difference between $\mathbf{F}^D_{i,t}$ and $\mathbf{F}^D_{i,t+1}$. V(E_i) is the time series feature vector of the event E_i. $\mathbf{F}^D_{i,t} = (\widetilde{f}_{i,t,1}, \widetilde{f}_{i,t,2}, ..., \widetilde{f}_{i,t,D})$. And $\mathbf{S}^D_{i,t} = \frac{\mathbf{F}^D_{i,t+1} - \mathbf{F}^D_{i,t}}{Interval(E_i)}$. We use Z-score to normalize feature values; $\widetilde{f}_{i,t,k} = \frac{f_{i,t+1,k} - \overline{f}_{i,k}}{\sigma(f_{i,k})}$ where $f_{i,t,k}$ is the k-th feature of the event E_i in time interval t. The mean of the feature k of the event E_i is denoted as $\overline{f}_{i,k}$ and $\sigma(f_{i,k})$ is the standard deviation of the feature k over all time intervals. We can skip this step, when we use Random Forest or Decision Trees, because they do not require feature normalization.

Features for the Rumor Detection Model

In selecting features for the rumor detection model, we have followed two rationales: a) we have selected features that we expect to be useful in early rumor detection and b) we have collected a broad range of features from related work as a basis for investigating the time-dependent impact of a wide variety of features in our time-dependence study. In total, we have constructed over 50 features (c.f., Table 6.1) in the three main categories i.e., *Ensemble*, *Twitter* and *Epidemiological* features. We refrained from using network features, since they are expected to be of little use in early rumor detection [CLL+17], since user networks around events need time to form. Following our general idea, none of our features are extracted from the content aggregations. Due to space limitation, we describe only our main features as follows.

Category	Feature	Description
Twitter Features	Hashtag	% tweets contain #hashtag [CMP11][LNL+15][QRRM11][GKCM14][LNL+15]
	Mention	% tweets mention others @user [CMP11][LNL+15][QRRM11][GKCM14][LNL+15]
	NumUrls	# URLs in the tweet [CMP11][QRRM11][GKCM14][YLYY12][LNL+15]
	Retweets	average # retweets [LNL+15]
	IsRetweet	% tweets are retweeted from others [CMP11][GKCM14]
	ContainNEWS	% tweets contain URL and its domain's catalogue is News [LNL+15]
	WotScore	average WOT score of domain in URL [GKCM14]
	URLRank5000	% tweets contain URL whose domain's rank less than 5000 [CMP11]
	ContainNewsURL	% tweets contain URL whose domain is News Website
Text Features	LengthofTweet	average tweet lengths [CMP11][GKCM14]
	NumOfChar	average # tweet characters [CMP11][GKCM14]
	Capital	average fraction of characters in Uppercase [CMP11]
	Smile	% tweets contain $:->,:-),;->,;-)$ [CMP11][GKCM14]
	Sad	% tweets contain $:-<,:-(,;->,;-($ [CMP11][GKCM14]
	NumPositiveWords	average # positive words [CMP11][GKCM14][YLYY12][LNL+15]
	NumNegativeWords	average # negative words [CMP11][GKCM14][YLYY12][LNL+15]
	PolarityScores	average polarity scores of the Tweets [CMP11][YLYY12][LNL+15]
	Via	% of tweets contain via [GKCM14]
	Stock	% of tweets contain $ [CMP11][GKCM14]
	Question	% of tweets contain ? [CMP11][LNL+15]
	Exclamation	% of tweets contain ! [CMP11][LNL+15]
	QuestionExclamation	% of tweets contain multi Question or Exclamation mark [CMP11][LNL+15]
	I	% of tweets contain first pronoun like I, my, mine, we, our [CMP11][GKCM14][LNL+15]
	You	% of tweets contain second pronoun like U, you, your, yours [CMP11]
	HeShe	% of tweets contain third pronoun like he, she, they, his, etc. [CMP11]
User Features	UserNumFollowers	average number of followers [CMP11][GKCM14][LNL+15]
	UserNumFriends	average number of friends [CMP11][GKCM14][LNL+15]
	UserNumTweets	average number of users posted tweets [CMP11][GKCM14][YLYY12][LNL+15]
	UserNumPhotos	average number of users posted photos [YLYY12]
	UserIsInLargeCity	% of users living in large city [YLYY12][LNL+15]
	UserJoinDate	average days since users joining Twitter [CMP11][YLYY12][LNL+15]
	UserDescription	% of user having description [CMP11][YLYY12][LNL+15]
	UserVerified	% of user being a verified user[YLYY12][LNL+15]
	UserReputationScore	average ratio of #Friends over (#Followers + #Friends) [LNL+15]
Epidemiological Features	β_{SIS}	Parameter β of Model SIS [JDS+13]
	α_{SIS}	Parameter α of Model SIS [JDS+13]
	β_{SEIZ}	Parameter β of Model SEIZ [JDS+13]
	b_{SEIZ}	Parameter b of Model SEIZ[JDS+13]
	l_{SEIZ}	Parameter l of Model SEIZ [JDS+13]
	p_{SEIZ}	Parameter p of Model SEIZ [JDS+13]
	ε_{SEIZ}	Parameter ε of Model SEIZ [JDS+13]
	ρ_{SEIZ}	Parameter ρ of Model SEIZ [JDS+13]
	R_{SI}	Parameter R_{SI} of Model SEIZ [JDS+13]
SpikeM Model Features	P_s	Parameter P_s of Model Spike [KCJ+13]
	P_a	Parameter P_a of Model SpikeM [KCJ+13]
	P_p	Parameter P_p of Model SpikeM [KCJ+13]
	Q_s	Parameter Q_s of Model SpikeM [KCJ+13]
	Q_a	Parameter Q_a of Model SpikeM [KCJ+13]
	Q_p	Parameter Q_p of Model SpikeM [KCJ+13]
Crowd Wisdom	CrowdWisdom	% of tweets containing "Debunking Words" [LNL+15] [ZRM15]
CreditScore	CreditScore	average CreditScore

Table 6.1: Features of Time Series Rumor Detection Model

Ensemble Features. We consider two types of Ensemble Features: features accumulating crowd wisdom and averaging feature for the Tweet credit Scores. The former are extracted from the surface level while the latter comes from the low dimensional level of tweet embeddings; that in a way augments the sparse crowd at early stage.

CrowdWisdom: Similar to [LNL+15], the core idea is to leverage the public's common sense for rumor detection: If there are more people denying or doubting the truth of an event, this event is more likely to be a rumor. For this purpose, [LNL+15] use an extensive list of bipolar sentiments with a set of combinational rules. In contrast to mere *sentiment* features, this approach is more tailored *rumor* context (difference not evaluated in [LNL+15]). We simplified and generalized the "dictionary" by keeping only a set of carefully curated *negative words*. We call them "debunking words" e.g., *hoax*, *rumor* or *not true*. Our intuition is, that the attitude of doubting or denying events is in essence sufficient to distinguish rumors from news. What is more, this generalization augments the size of the crowd (covers more 'voting' tweets), which is crucial, and thus contributes to the quality of the crowd wisdom. In our experiments, "debunking words" is an high-impact feature, but it needs substantial time to "warm up"; that is explainable as the crowd is typically sparse at early stage.

CreditScore: The sets of single-tweet models' predicted probabilities are combined using an *ensemble averaging*-like technique. In specific, our pre-trained $CNN + LSTM$ model predicts the credibility of each tweet tw_{ij} of event E_i. The *softmax* activation function outputs probabilities from 0 (rumor-related) to 1 (news). Based on this, we calculate the average prediction probabilities of all tweets $tw_{ij} \in E_i$ in a time interval t_{ij}. In theory there are different sophisticated ensembling approaches for averaging on both training and test samples; but in a real-time system, it is often convenient (while effectiveness is only affected marginally) to cut corners. In this work, we use a sole training model to average over the predictions. We call the outcome CreditScore.

6.1.4 Experimental Evaluation

Data Collection

To construct the training dataset, we collected rumor stories from online rumor tracking websites such as **snopes.com** and **urbanlegends.about.com**. In more detail, we crawled 4300 stories from these websites. From the story descriptions we manually constructed queries to retrieve the relevant tweets for 270 rumors with high impact. Our approach to query construction mainly follows [GKCM14]. For the news event instances (non-rumor examples), we make use of the manually constructed corpus from Mcminn et al. [MMJ13], which covers 500 real-world events. In [MMJ13], tweets are retrieved via Twitter firehose API from 10^{th} of October 2012 to 7^{th} of November 2012. The involved events are manually verified and relate to tweets with relevance judgments, which results in a high quality corpus. From the 500 events, we select top 230 events with the highest tweet volumes (as a criteria for event impact). Furthermore, we have added 40 other news events, which happened around the time periods of our rumors. This results in a dataset of 270 rumors and

Type	Min Volume	Max Volume	Total	Average
News	98	17414	345235	1327.82
Rumors	44	26010	182563	702.06

Table 6.2: Tweet Volume of News and Rumors

270 events. The dataset details are shown in Table 6.2. To serve our learning task. we then constructs two distinct datasets for (1) single tweet credibility and (2) rumor classification.

Training data for single tweet classification. Here we follow our assumption that an event might include sub-events for which relevant tweets are rumorous. To deal with this complexity, we train our single-tweet learning model only with manually selected *breaking and subless* [2] events from the above dataset. In the end, we used 90 rumors and 90 news associated with 72452 tweets, in total. This results in a highly-reliable large-scale ground-truth of tweets labelled as *news*-related and *rumor*-related, respectively. Note that the labeling of a tweet is inherited from the event label, thus can be considered as an semi-automatic process.

Single Tweet Classification Experiments

For the evaluation, we developed two kinds of classification models: traditional classifier with handcrafted features and neural networks without tweet embeddings. For the former, we used 27 distinct surface-level features extracted from single tweets (analogously to the Twitter-based features presented in Section 6.1.3). For the latter, we select the baselines from NN-based variations, inspired by state-of-the-art short-text classification models, i.e., Basic tanh-RNN , 1-layer GRU-RNN, 1-layer LSTM, 2-layer GRU-RNN, FastText [JGBM16] and CNN+LSTM [ZSLL15] model. The hybrid model CNN+LSTM is adapted in our work for tweet classification.

Single Tweet Model Settings. For the evaluation, we shuffle the 180 selected events and split them into 10 subsets which are used for 10-fold cross-validation (we make sure to include near-balanced folds in our shuffle). We implement the 3 non-neural network models with Scikit-learn[3]. Furthermore, neural networks-based models are implemented with TensorFlow [4] and Keras[5]. The first hidden layer is an embedding layer, which is set up for all tested models with the embedding size of 50. The output of the embedding layer are low-dimensional vectors representing the words. To avoid overfitting, we use the 10-fold cross validation and dropout for regularization with dropout rate of 0.25.

Single Tweet Classification Results. The experimental results of are shown in Table 6.3. The best performance is achieved by the CNN+LSTM model with a good accuracy

[2] the terminology *subless* indicates an event with no sub-events for short.

[3] scikit-learn.org/

[4] https://www.tensorflow.org/

[5] https://keras.io/

of 81.19%. The non-neural network model with the highest accuracy is RF. However, it reaches only 64.87% accuracy and the other two non-neural models are even worse. So the classifiers with hand-crafted features are less adequate to accurately distinguish between rumors and news.

Model	Accuracy
CNN+LSTM	**0.8119**
2-layer GRU	0.7891
1-layer GRU	0.7644
1-layer LSTM	0.7493
Basic RNN with tanh	0.7291
FastText	0.6602
Random Forest	**0.6487**
SVM	0.5802
Decision Trees	0.5774

Table 6.3: Single Tweet Classification Performance

Feature	Importance
PolarityScores	0.146
Capital	0.096
LengthOfTweet	0.092
UserTweets	0.087
UserFriends	0.080
UserReputationScore	0.080
UserFollowers	0.079
NumOfChar	0.076
Stock	0.049
NumNegativeWords	0.030
Exclamation	0.023

Table 6.4: Top Features Importance

Discussion of Feature Importance For analyzing the employed features, we rank them by importances using RF (see 6.4). The best feature is related to sentiment polarity scores. There is a big difference between the sentiment associated to rumors and the sentiment associated to real events in relevant tweets. In specific, the average polarity score of news event is -0.066 and the average of rumors is -0.1393, showing that rumor-related messages tend to contain more negative sentiments. Furthermore, we would expect that verified users are less involved in the rumor spreading. However, the feature appears near-bottom in the ranked list, indicating that it is not as reliable as expected. Also interestingly, "IsRetweeted" feature is pretty much useless, which means the probability of people retweeting rumors or true news are similar (both appear near-bottom in the ranked feature list).

It has to be noted here that even though we obtain reasonable results on the classification task in general, the prediction performance varies considerably along the time dimension. This is understandable, since tweets become more distinguishable, only when the user gains more knowledge about the event.

6.1.5 Rumor Datasets and Model Settings

We use the same dataset described in Section 6.1.4. In total –after cutting off 180 events for pre-training single tweet model – our dataset contains 360 events and 180 of them are labeled as rumors. Those rumors and news fall comparatively evenly in 8 different categories, namely *Politics*, *Science*, *Attacks*, *Disaster*, *Art*, *Business*, *Health* and *Other*. Note, that the events in our training data are not necessarily subless, because it is natural for high-impact events (e.g., *Missing MH370* or *Munich shooting*) to contain sub-events. Actually, we empirically found that roughly 20% of our events (mostly news) contain sub-events. As a rumor is often of a long circulating story [FAEC14], this results in a rather long time span. In this work, we develop an event identification strategy that focuses on the first 48 hours after the rumor is peaked. We also extract 11,038 domains, which are contained in tweets in this 48 hours time range.

Rumor Detection Model Settings. For the time series classification model, we only report the best performing classifiers, SVM and Random Forest, here. The parameters of SVM with RBF kernel are tuned via grid search to $C = 3.0$, $\gamma = 0.2$. For Random Forest, the number of trees is tuned to be 350. All models are trained using 10-fold cross validation.

Rumor Classification Results

We tested all models by using 10-fold cross validation with the same shuffled sequence. The results of these experiments are shown in Table 6.5. Our proposed model (***Ours***) is the time series model learned with Random Forest including all ensemble features; $TS-SVM$ is the baseline from [MGW^{+}15], and $TS-SVM_{all}$ is the $TS-SVM$ approach improved by using our feature set. In the lower part of the table, RNN_{el} is the RNN model at event-level [MGM^{+}]. As shown in the Table 6.5 and as targeted by our early detection approach, our model has the best performance in all case over the first 24 hours, remarkably outperforming the baselines in the first 12 hours of spreading. The performance of RNN_{el} is relatively low, as it is based on aggregated *contents*. This is expected as the news (non-rumor) dataset used in [MGM^{+}] are crawled also from snopes.com, in which events are often of small granularity (aka. subless). As expected, exploiting contents solely at event-level is problematic for high-impact, evolving events on social media. We leave a deeper investigation on the sub-event issue to future work.

Model	Accuracy in hours								
	1	6	12	18	24	30	36	42	48
Ours	**0.82**	**0.84**	**0.84**	0.84	**0.87**	0.87	**0.88**	0.89	**0.91**
$TS-SVM_{all}$	0.76	0.79	0.83	0.83	0.87	**0.88**	0.86	0.89	0.90
$TS-SVM_{Credit}$	0.73	0.80	0.83	**0.85**	0.85	0.86	0.88	**0.90**	0.90
$TS-SVM$ [MGW^{+}15]	0.69	0.76	0.81	0.81	0.84	0.86	0.87	0.88	0.88
RNN_{el} [MGM^{+}]	0.68	0.77	0.81	0.81	0.84	0.83	0.81	0.85	0.86
$SVM_{static}+Epi$ [JDS^{+}13]	0.60	0.69	0.71	0.72	0.75	0.78	0.75	0.78	0.81
$SVM_{static}+SpikeM$ [KCJ^{+}13]	0.58	0.68	0.72	0.73	0.77	0.78	0.78	0.79	0.77
SVM_{static} [YLYY12]	0.62	0.70	0.70	0.72	0.75	0.80	0.79	0.78	0.77

Table 6.5: Performance of different models over time (bold for best accuracy, underlined for second-to-best). TS indicates time-series structure; we separate the TS models (upper) with the static ones (lower).

CreditScore and CrowdWisdom . As shown in Table 6.6, *CreditScore* is the best feature in overall. In Figure 6.4 we show the result of models learned with the full feature set with and without *CreditScore*. Overall, adding *CreditScore* improves the performance, especially for the first 8-10 hours. The performance of *all-but-CreditScore* jiggles a bit after 16-20 hours, but it is not significant. *CrowdWisdom* is also a good feature which can get

75.8% accuracy as a single feature. But its performance is poor (less than 70%) in the first 32 hours getting better over time (see Table 6.6). Table 6.6 also shows the performance of *sentiment* feature (*PolarityScores*), which is generally low. This demonstrates the effectiveness of our *curated* approach over the *sentiments*, yet the crowd needs time to unify their views toward the event while absorbing different kinds of information.

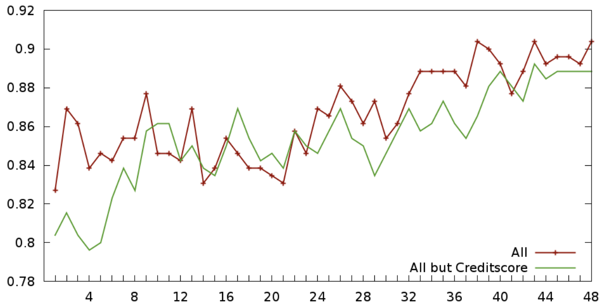


Figure 6.4: Accuracy: All features with and without CreditScore.

Features	**Ranks**									
Hours	1	6	12	18	24	30	36	42	48	AVG
CreditScore	1	0	0	0	0	0	0	0	0	0.08
CrowdWisdom	34	38	21	14	8	5	5	2	2	13.18
PolarityScores	12	15	23	28	33	33	34	31	32	28

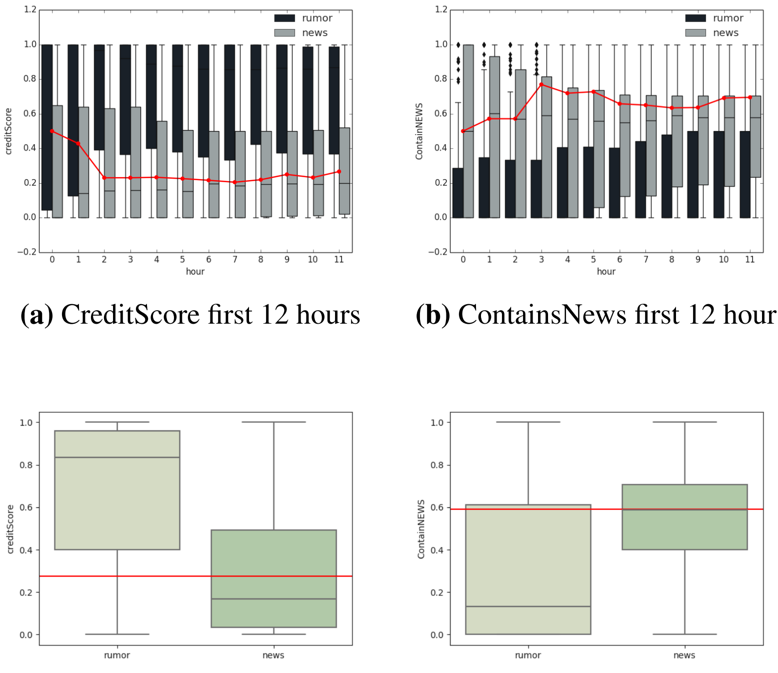
Table 6.6: Importance ranking of CreditScore, CrowdWisdom and PolarityScores over time; 0 indicates the best rank.

Case Study: Munich Shooting . We showcase here a study of the Munich shooting. We first show the event timeline at an early stage. Next we discuss some examples of misclassifications by our "weak" classifier and show some analysis on the strength of some highlighted features. The rough event timeline looks as follows.

- At 17:52 CEST, a shooter opened fire in the vicinity of the Olympia shopping mall in Munich. 10 people, including the shooter, were killed and 36 others were injured.
- At 18:22 CEST, the first tweet was posted. There might be some certain delay, as we retrieve only tweets in English and the very first tweets were probably in German. The tweet is *"Sadly, i think there's something terrible happening in #Munich #Munchen. Another Active Shooter in a mall. #SMH"*.
- At 18:25 CEST, the second tweet was posted: *"Terrorist attack in Munich????"*.
- At 18:27 CEST, traditional media (BBC) posted their first tweet. *"'Shots fired' in Munich shopping centre - http://www.bbc.co.uk/news/world-europe-36870800a02026 @TraceyRemix gun crime in Germany just doubled"*.

- At 18:31 CEST, the first misclassified tweet is posted. It was a tweet with shock sentiment and swear words: *"there's now a shooter in a Munich shopping centre.. What the f*** is going on in the world. Gone mad"*. It is classified as *rumor-related*.

We observe that at certain points in time, the volume of rumor-related tweets (for sub-events) in the event stream surges. This can lead to *false positives* for techniques that model events as the aggregation of all tweet contents; that is undesired at critical moments. We trade-off this by debunking at single tweet level and let each tweet vote for the credibility of its event. We show the *CreditScore* measured over time in Figure 6.5a. It can be seen that although the credibility of some tweets are low (rumor-related), averaging still makes the *CreditScore* of Munich shooting higher than the average of news events (hence, close to a *news*). In addition, we show the feature analysis for ContainNews (percentage of URLs containing news websites) for the event *Munich shooting* in Figure 6.5b. We can see the curve of *Munich shooting* event is also close to the curve of average news, indicating the event is more news-related.



(a) CreditScore first 12 hours

(b) ContainsNews first 12 hours

(c) CreditScore 48 hours

(d) ContainsNews 48 hours

Figure 6.5: Creditscore and ContainsNews for *Munich shooting* in red lines, compared with the corresponding average scores for *rumor* and *news*.

6.1.6 Conclusion

In this work, we propose an effective cascaded rumor detection approach using deep neural networks at tweet level in the first stage and wisdom of the "machines", together with a variety of other features in the second stage, in order to enhance rumor detection performance in the early phase of an event. The proposed approach outperforms state of the art methods for early rumor detection. There is, however, still considerable room to improve the effectiveness of the rumor detection method. The support for events with rumor sub-events is still limited. The current model only aims not to misclassify long-running, multi-aspect events where rumors and news are mixed and evolve over time as false positive.

6.2 Temporal model stability and predictivity at early-stage in clinical domain

Diabetes mellitus has been a major and global problem for a long time, as it is report that there are over 400 million patients over the world [6]. The knowledge of glucose concentration in blood is a key aspect in the diagnosis and treatment of diabetes. The use of signal processing techniques on glucose data started a long time ago, when glucose time-series in a given individual could be obtained in lab study from samples drawn in the blood at a sufficiently high rate. In particular, related work employed not only linear (e.g., correlation and spectrum analysis, peak detection), but also nonlinear (e.g., approximate entropy) methods to investigate oscillations present in glucose (and insulin) time-series obtained, during hospital monitoring, by drawing blood samples every 10-15 min for up to 48 h [SFC10]. In these settings, long term (e.g., days or months) studies resorted to self-monitoring blood glucose (SMBG) data, i.e., approx. 3 samples per day obtained by the patient herself by using fingerstick glucose meters. The retrospective analysis of SMBG time-series was used by physicians, together with the information taken from the ‘patient’s diary‘ (e.g., insulin dosage, meals intake, physical exercise) and some glycaemic indexes (typically HbA1c), to assess glucose control and the effectiveness of a particular therapy [SFC10].

With the support of continuous glucose monitoring (CGM) sensors, the development of new strategies for the treatment of diabetes has been accelerated in recent years. In particular, CGM sensors can be injected into ‘online‘ recommender systems that are able to generate alerts when glucose concentration is predicted to exceed the normal range thresholds. Recently, there has been a lot of complex data-driven prediction models [EOCQS09, PBM$^+$14, COV$^+$17, FMM$^+$17] that are built based on the CGM data, and have been shown to be effective. These data-driven models, or machine learning/deep learning are data-hungry, hence, its performance on **sparse / non-continuous data** is still a question. CGM data are still not always available for all diabetic patients for many reasons [7]; while a personalized or *patient-level* model that are trained on the same patient’s data is essential. In this section, we examine the performance of these machine leaning approaches on our real, limited data of a group of diabetic patients. Our contributions are two-fold: (1) we provide a quantitative study on the predictability of machine learned models on limited and sparse data; (2) we propose a prediction system that is robust on noisy data (based on prediction

[6] https://www.diabetes.co.uk/diabetes-prevalence.html

[7] http://time.com/4703099/continuous-glucose-monitor-blood-sugar-diabetes/

interval).

6.2.1 Dataset Overview

The data collection study was conducted from end of February to beginning of April 2017 and includes 9 patients who were given specially prepared smartphones. Measurements on carbohydrate consumption, blood glucose levels, and insulin intake were made with the Emperras Esysta system [8]. Measurements on physical activities were obtained using the Google Fit app. We use only steps information (number of steps) for our study.

We describe briefly here some basic patient information. Half of the patients are female and ages range from 17 to 66, with a mean age of 41.8 years. Body weight, according to BMI (Body mass index), is normal for half of the patients, four are overweight and one is obese. The mean BMI value is 26.9. Only one of the patients suffers from diabetes type 2 and all are in ICT therapy [9]. In terms of time since being diagnosed with diabetes, patients vary from inexperienced (2 years) to very experienced (35 years), with a mean value of 13.9 years. We anonymize the patients and identify them by IDs (from 8 to 17, we do not have information for patient 9).

Frequency of Measurements

We give an overview of the number of different measurements that are available for each patient. The study duration varies among the patients, ranging from 18 days, for patient 8, to 33 days, for patient 14. Likewise, the daily number of measurements taken for carbohydrate intake, blood glucose level and insulin units vary across the patients. The median number of carbohydrate log entries vary between 2 per day for patient 10 and 5 per day for patient 14. Median number of blood glucose measurements per day varies between 2 and 7. Similarly, insulin is used on average between 3 and 6 times per day. In terms of physical activity, we measure the 10 minute intervals with at least 10 steps tracked by the google fit app. This very low threshold for now serves to measure very basic movements and to check for validity of the data. Patients 11 and 14 are the most active, both having a median of more than 50 active intervals per day (corresponding to more than 8 hours of activity). Patient 10 on the other hand has a surprisingly low median of 0 active 10 minutes intervals per day, indicating missing values due to, for instance, not carrying the smartphone at all times.

Measurements per Hour of Day

Figure 6.8 show measurements of blood glucose, carbohydrates and insulin per hour of day for patient 13 and 14. Overall, the distribution of all three kinds of values throughout the

[8] https://www.emperra.com/en/esysta-product-system/

[9] describes as a model of an insulin therapy for the diabetics with two different types of insulin.

day roughly correspond to each other. In particular, for most patients the number of glucose measurements roughly matches or exceeds the number of rapid insulin applications throughout the days. Notable exceptions are patients 14, 15, and 17 (figures excluded). For patient 14, in the evening the number of meals and rapid insulin applications match but exceed the number of blood glucose measurements by far. Patient 17 has more rapid insulin applications than glucose measurements in the morning and particularly in the late evening. For patient 15, rapid insulin again slightly exceeds the number of glucose measurements in the morning. Curiously, the number of glucose measurements match the number carbohydrate entries – it is possible the discrepancy is a result of missing (glucose and carbohydrate) measurements. We further show the blood glucose distribution of each patient in Figure 6.6. The different lengths of the interquartile range for each distribution also reflects the difficulty of prediction problem on different patients.

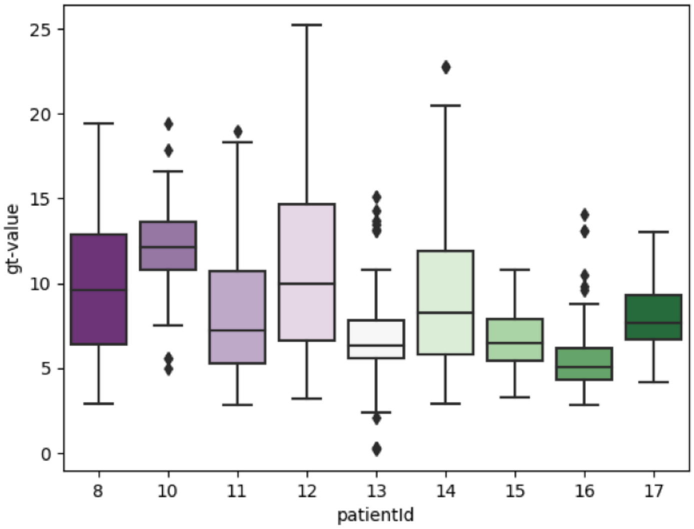


Figure 6.6: Blood glucose distribution for each patient.

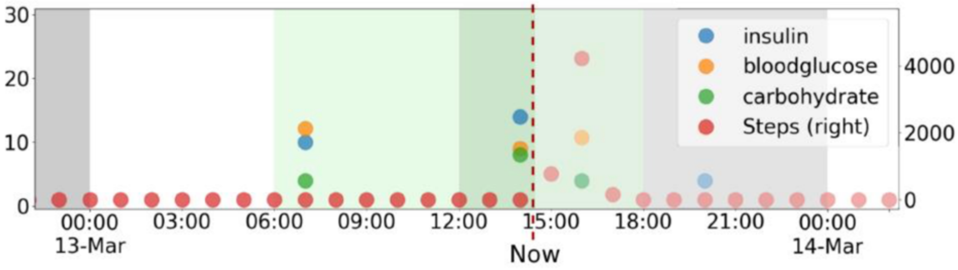
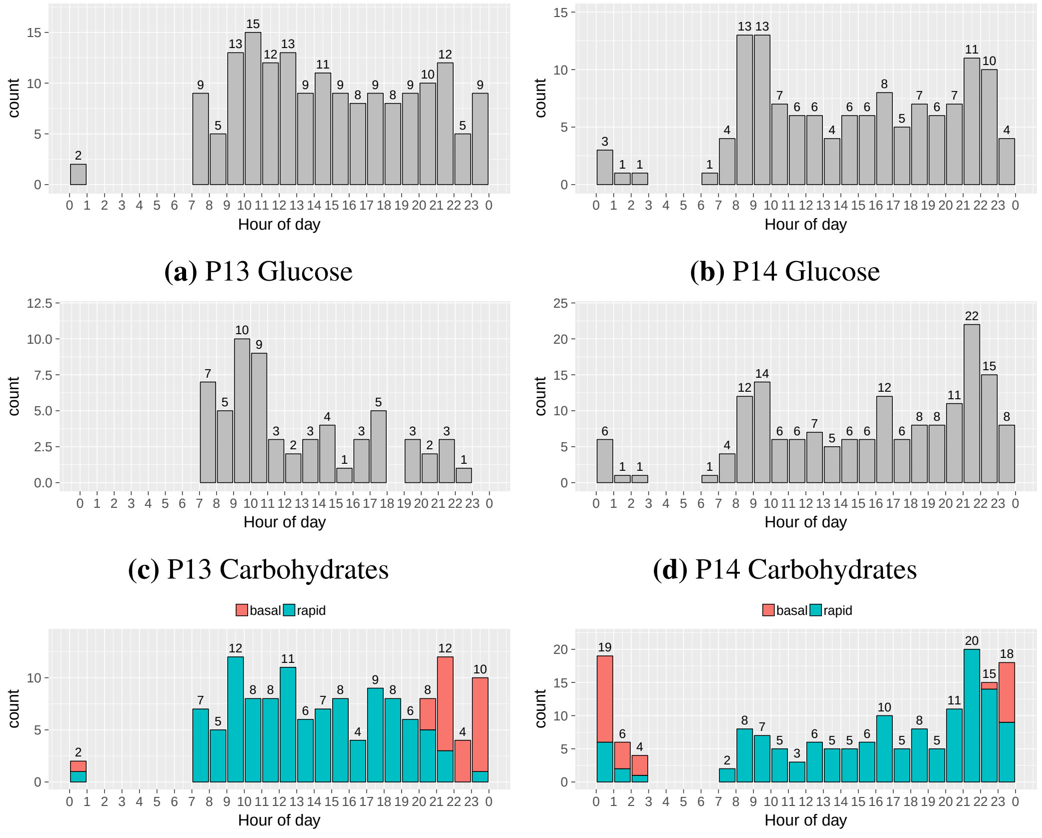


Figure 6.7: Blood glucose prediction scenario.



(a) P13 Glucose

(b) P14 Glucose

(c) P13 Carbohydrates

(d) P14 Carbohydrates

(e) P13 Insulin

(f) P14 Insulin

Figure 6.8: Glucose, carbohydrate and insulin values per hour of day for patients 13 and 14.

6.2.2 Prediction

Our first approach to blood glucose prediction is based on a regression type form of time series prediction. Given historical blood glucose data, we learn a model that predicts future glucose values based on a representation of the current situation (including the recent past), using information on patient context, recent insulin applications, carbohydrate intake, and physical activity levels.

Setup

Prediction task Our prediction task is a time series prediction of blood glucose values (in mmol/L) with a prediction horizon of 1 hour. Consequently, we can construct a data instance for each glucose measurement found in the dataset and use all information available up until 1 hour before the measurement for predicting the glucose value (c.f., Figure 6.7).

Evaluation Protocol Performance is evaluated on a per patient basis. In addition, we average performance over patients to get an overview. For each patient, we consider the first 66% of blood glucose measurements as training data to learn the models and the last 34% as test data to evaluate prediction performance.

Performance Measures Prediction performance is measured in terms of median absolute error (MdAE), root mean squared error (RMSE) and symmetric mean absolute percentage error (SMAPE). Given are ground truth values y_i and predictions $\hat{y}_i$, with $i \in [1,n]$. Median absolute error measures the median error made and is defined as

$$\text{MdAE} = \underset{i}{\text{median}}(|\hat{y}_i - y_i|).$$

Root mean squared error weighs larger errors more heavily and is defined as

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}.$$

Symmetric mean absolute percentage error relates prediction errors to predicted values. It is defined as

$$\text{SMAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\frac{|\hat{y}_i - y_i|}{(|y_i| + |\hat{y}_i|)/2}.$$

Note that this gives a result between 0% and 200%. Further, the measure penalizes a) deviating for low values and b) over-forecasting.

Algorithms

Simple Baselines As standard simple baselines, we use the last value observed one hour before the value that is being predicted (*Last*) and the arithmetic mean of glucose values in the training set.

Context-AVG As a more advanced baseline, we use a (temporal) context weighted average of previous glucose values. As our analysis showed differences in glucose values according to time of the day, we weigh previous glucose values base on temporal proximity, weighted exponentially decreasing in the difference of time of day.

Long-short-term-memory . LSTM is a recurrent neural network model that effectively accounts for the long-term sequence dependence among glucose inputs.

RandomForest The Random Forest Regressor (RF) is a meta estimator that learns an ensemble of regression trees [Bre01], averaging the output of individual regression trees to perform the prediction. We use a standard value of 500 estimators, as well as a minimal leaf size of 4 for the individual trees to reduce overfitting of the individual models.

ExtraTrees The Extra-Trees Regressor (ET) is a variation on RandomForest that uses a different base learner: Extremely randomized trees [GEW06]. In contrast to regular regression trees, best split values per feature are chosen randomly. We use 300 estimators and a minimum leaf size of 2.

Overall Results

In this section we report aggregate results, averaged over all patients. Table 6.7 shows regression performance averaged over all patients. Performance is based on 42 test instances on average. The simple baselines *Last* and *AVG* achieve median errors of 3.3 and 2.5 mmol/L. Weighing previous glucose values based on time of the day (*Context-AVG*) improves average median errors to 2.28 mmol/L. The Extra-Trees Regressor achieves the lowest MdAE of 2.16 and similarly slightly outperforms Context-AVG in terms of RMSE and SMAPE. In comparison to predicting the arithmetic mean (*AVG*), however, RMSE does not improve by much (12.15 vs 12.96), indicating that the ensemble is not able to predict extreme errors well on average. We additionally report the performance of a neural-network based model, the Long-short-term-memory (LSTM), trained with 10 and 100 epochs. LSTM seems to be quite stable for MdAE but varies substantially for RMSE and SMAPE. The performance of LSTM actually gets much worse after 100 epochs, that indicates the prone to *overfitting*. This show the instability of the model towards our dataset, and thus we do not consider the LSTM results for model comparison in Table 6.7.

Method	MdAE	RMSE	SMAPE
Last	3.28	25.71	40.96
AVG	2.51	12.96	31.42
Context-AVG	2.28	12.53	29.71
ARIMA	2.40	13.88	31.61
LSTM (10 iter)	*2.24*	*10.41*	*29.02*
LSTM (100 iter)	*2.76*	*19.24*	*35.64*
RandomForest	2.27	**12.05**	29.98
Extremely (randomized) Trees	**2.16**	12.15	**29.56**

Table 6.7: Overall regression performance averaged over all patients. Best performance per measure is marked in bold (results in italic are not considered for comparison).

6.2.3 Prediction confidence

In this section, we study the confidence of our best performed prediction tree-based models, RandomForest and ExtraTrees. This would, to an extent, facilitate us to answer an important question, **when** the system is reliable enough to give out predictions. Thus, we study the variability of predictions and *estimate* standard errors for the prediction model.

Prediction intervals When looking at two regression models, while the model predictions could be similar, confidence in them would vary if we look at the training data, a less and more spread out data could bring a low confidence. Hence, a prediction returning a single value (typically meant to minimize the squared error) likewise does not relay any information about the underlying distribution of the data or the range of response values. We hence, leverage the concept of **prediction intervals** to supplement for the noisy data and enhance the end model, in the sense that it can **refuse** to give prediction at certain time when the *confidence* is low.

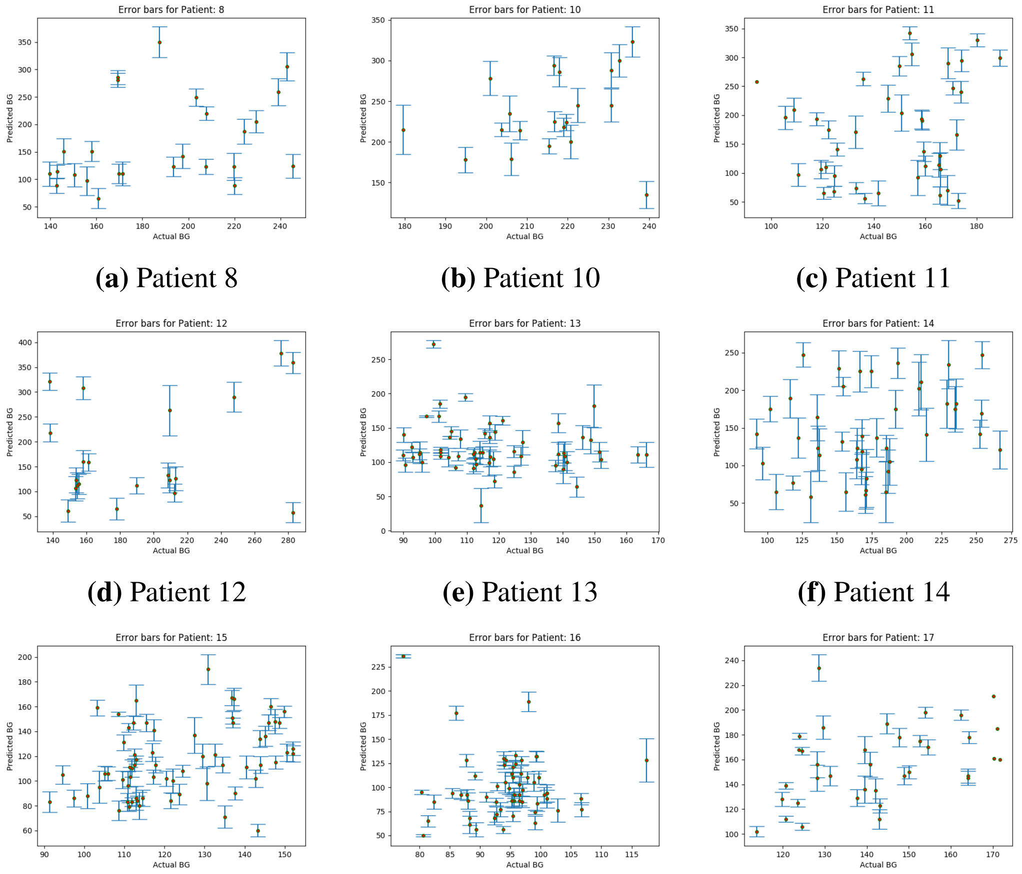
A prediction interval or confidence interval is an estimate of an interval into which the future observations will fall with a given probability. In other words, it can quantify our confidence or certainty in the prediction. Unlike confidence intervals from classical statistics, which are about a parameter of population (such as the mean), prediction intervals are about individual predictions [dat]. We leverage the confidence interval estimations for Random Forests, proposed in [WHE14], that account for certain *variability* estimation (of individual trees) bias to conduct the experiments.

Estimating the variance of bagged learners based on the preexisting bootstrap replicates can be challenging, as there are two distinct sources of noise. In addition to the sampling noise (i.e., the noise arising from randomness during data collection), we also need to control the Monte Carlo noise arising from the use of a finite number of bootstrap replicates. We study the *bias-correction* methods of both sampling noise and Monte Carlo noise as a filtering technique.

Regression evaluation

We report here the variablity evaluation across all patients for the regression task. Figure 6.9 show the error bars using unbiased variance for all patients. We then show in Figures 6.10 the error bar graphs for patient 8 in an *incremental* training size setting – meaning that we keep the same actual test set, but training on only part of the training data. E.g., $1/4$ training data indicates that we 'look back' on only $1/4$ of the available past data. The more dots that near the diagonal show the more 'accurate' is our prediction model. And the error bars show the 'confidence' interval. Figure 6.10a indicates the high 'confidence' in the predictions with little training data, yet the dots are far away from the diagonal.

when to predict: on the training size evaluation. To answer this question, we set up an evaluation setting with increasing size of number of instances, order by time. Each training point is evaluated by leave-one-out validation. We show in Figure 6.11 the results for patient 8. The general conclusion is the that the more training data, the better the performance is, as witness for patient 13, 15 or 17. However, the results for such patients e.,g patient 8, 11 or 16 show that the training size increment could also bring more noise and decrease the results. We envision that it could because the learned model is not stabilized yet with the limited number of instances in our experiment. In addition, training size is not the only factor to decide when to predict. We hence move on to examine the other two



(a) Patient 8 **(b)** Patient 10 **(c)** Patient 11

(d) Patient 12 **(e)** Patient 13 **(f)** Patient 14

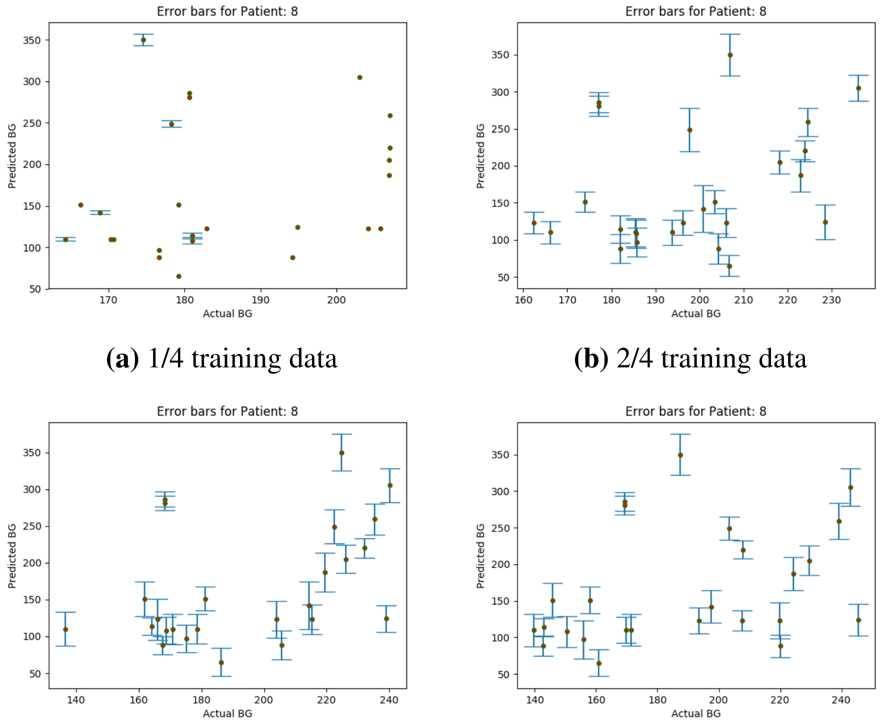
(g) Patient 15 **(h)** Patient 16 **(i)** Patient 17

Figure 6.9: Error bar graphs for predicted BG using unbiased variance.

factors: (1) model stability - via std. dev. and (2) prediction confidence toward coming instances.

when to predict: on the model stability. To answer this question, we measure the stability of the model by the standard deviation of the k-fold cross validation with incremental training size. Figure 6.15 indicate on MAE and RMSE metrics, the model seems to be more stabilized with the more number of training data. This is a good indicator for the *when to predict* questions.

when to predict: on the prediction confidence. We show in Figure 6.13 and Figure 6.14 the confidence distribution at each run of the 5-fold CV for different patients based on *bias* and *no-bias* confidences respectively. The results show the confidence distributions are rather similar across different run, indicating that the temporal order of the instances does not impact much on the model performance. Base on the distribution, we move on the the threshold parameter tuning for the data filtering using confidence interval. The idea is to



(a) 1/4 training data

(b) 2/4 training data

(c) 3/4 training data

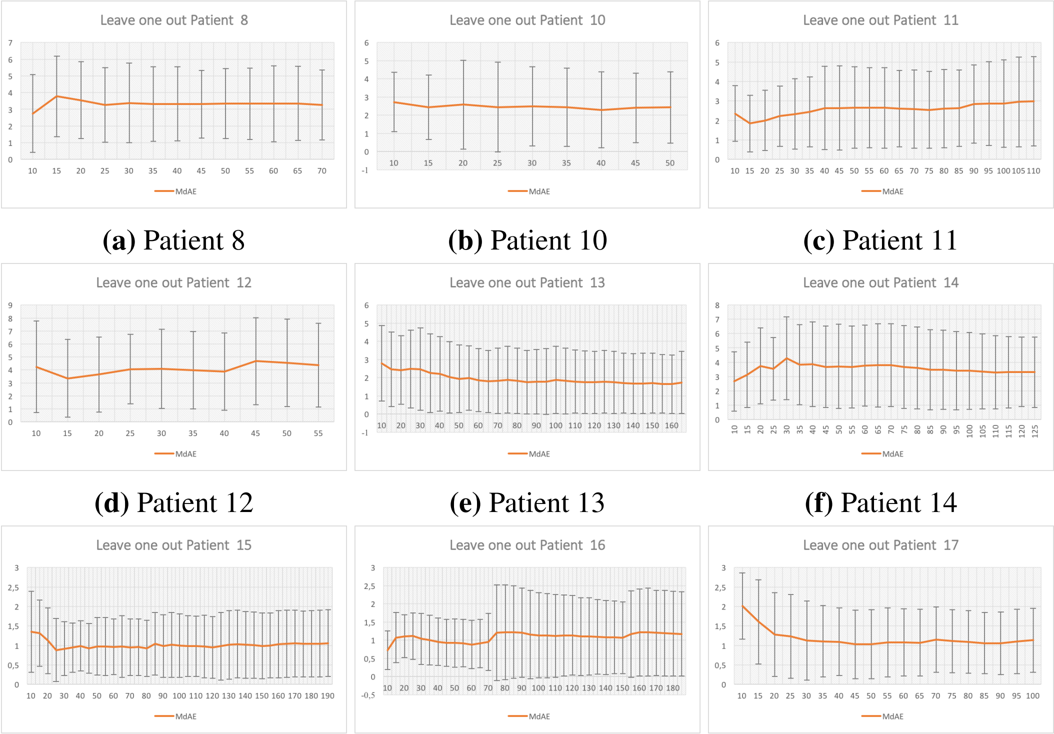
(d) 4/4 training data

Figure 6.10: Incremental training size - error bar graphs for predicted BG using unbiased variance for patient 8.

answer the question, "if we filter low confidence instances (high confident interval), will the model perform better?" We found that, for some patients, i.e., patient 10 and 13, the filtering technique substantially enhance the model performances on MAE and RMSE (not shown) metrics. It is witness that the *biased* confidence measure somewhat works better than *non-biased* one across patients. However, for some patients i.e., patient 8 it seems does not bring any effects.

when to predict: combined factors. Figure 6.12 show some highlighted combined filtering techniques. In general, combining the aforementioned factors together does improve the model performance. However, the combination is not straightforward, e.g., *confidence interval* filtering lower the performance at the starting time when the model is unstable aka. *cold start*. Hence, there is not enough evidence for us to make a *hard* decision. The more *trial-and-error* attempts on the fly or a bigger dataset however will be at ease to be built on these as a foundation.

Overall results with Filtering methods We show in Table 6.8 the overall results of our models with different filtering approaches for all patients. We use 2 different filtering approaches: (1) Sanity filter, heuristics (e.g., remove out wrongly input measurement or moments when the last glucose level input is too far) that remove noise and (2) Stability filter: prediction confidence (std. dev is not needed when the training size is large enough). The results show that the *stability filter* (based on bias and bias-corrected) achieve the best performance, without the need of human efforts on sanity filter. Sole stability filter also provide more predictions (avg. 24) than other filtering combination.

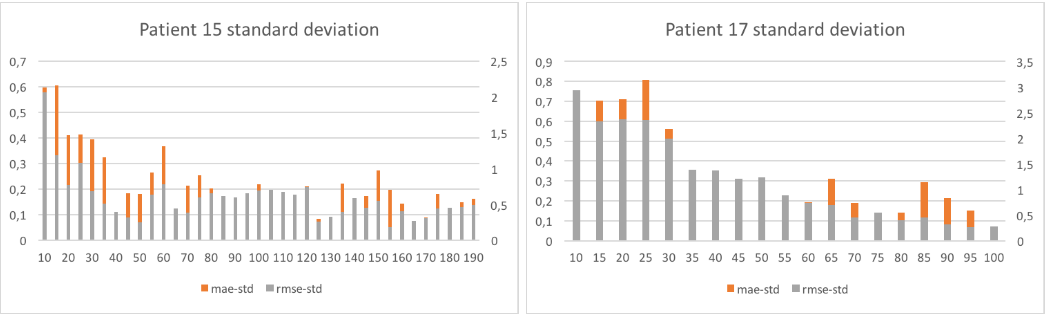


(a) Patient 8 **(b)** Patient 10 **(c)** Patient 11

(d) Patient 12 **(e)** Patient 13 **(f)** Patient 14

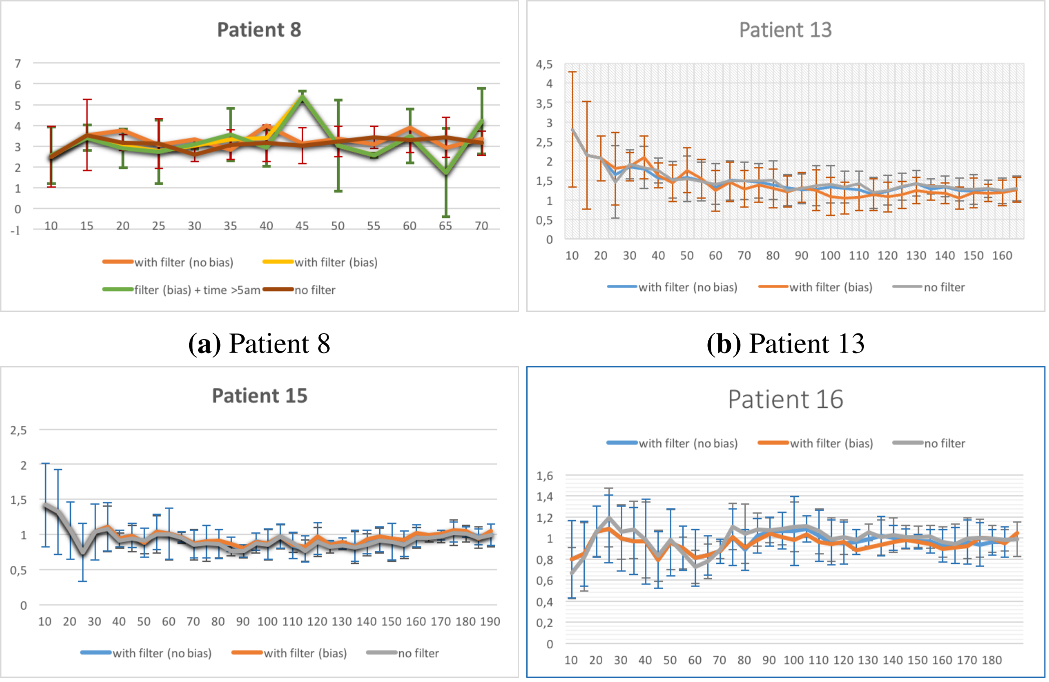
(g) Patient 15 **(h)** Patient 16 **(i)** Patient 17

Figure 6.11: Leave-one-out cross validation with incremental training size.



(a) Patient 15 **(b)** Patient 17

Figure 6.15: Standard deviation with incremental training size.



(a) Patient 8

(b) Patient 13

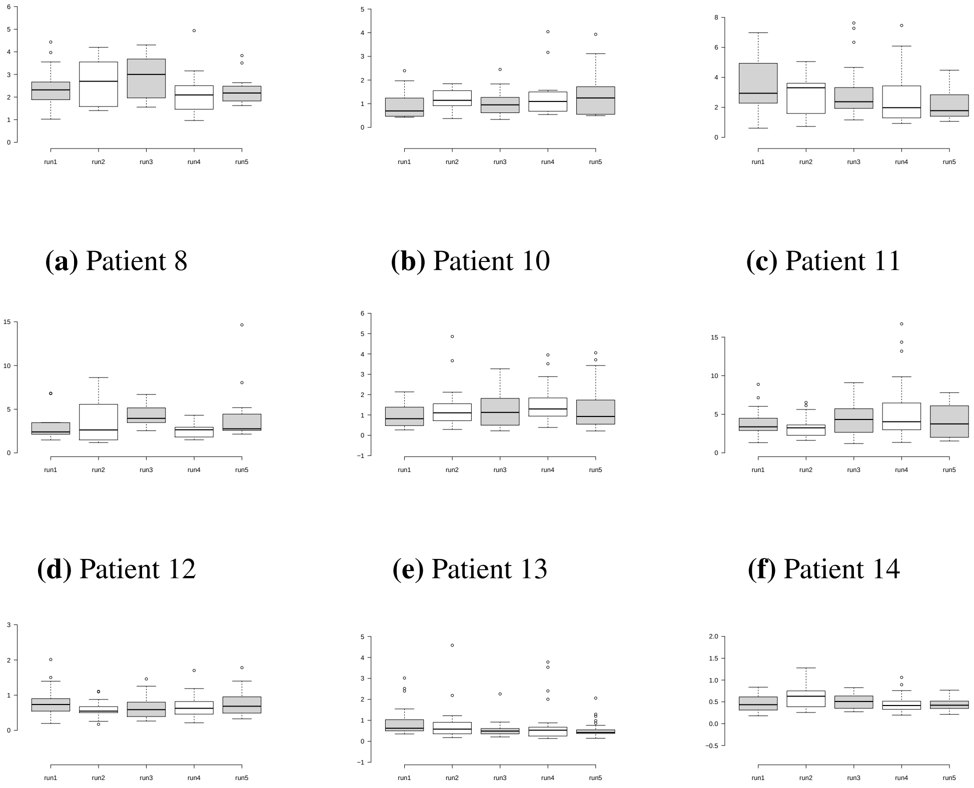
(c) Patient 15

(d) Patient 16

Figure 6.12: 5-fold cross validation with incremental training size.

6.3 Conclusion

We studied the predictability of machine-learning models in the scenarios of non-continuous blood glucose tracking. Additionally, we studied the stability and robustness of the learned model over time. We show that Random Forest and Extra Tree ensemble-based models are the most suitable models for this case, as they can account for the outliers as well as overfitting problems when the data are limited. Our further study on the prediction confidence show that the model can give reliable predictions after acquiring 25-30 instances.



(a) Patient 8 **(b)** Patient 10 **(c)** Patient 11

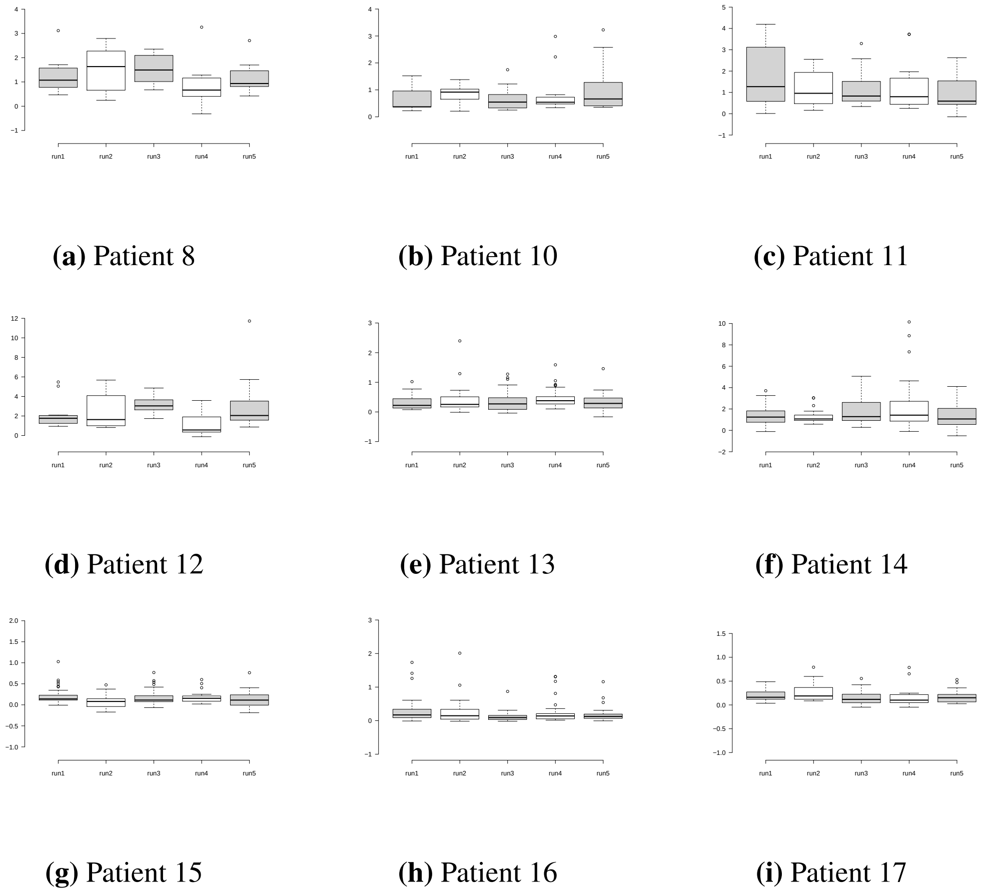
(d) Patient 12 **(e)** Patient 13 **(f)** Patient 14

(g) Patient 15 **(h)** Patient 16 **(i)** Patient 17

Figure 6.13: Confidence distributions at each run of 5-fold CV for predicted BG using biased variance.

Table 6.8: Average performance of different filtering approaches for all patients.

Model	**# predictions**	**MAE**	**MdAE**	**RMSE**	**SMAPE**
rf	42	2.58	2.27	12.05	29.98
et	42	2.55	2.16	12.15	29.56
rf + sanity filter	16	2.22	2.01	8.80	28.10
et + sanity filter	16	2.29	2.06	9.01	29.36
rf + sanity + stability filter	15	2.22	1.92	8.71	27.82
rf + stability filter	24	**1.92**	**1.77**	**7.57**	**22.65**



(a) Patient 8 **(b)** Patient 10 **(c)** Patient 11

(d) Patient 12 **(e)** Patient 13 **(f)** Patient 14

(g) Patient 15 **(h)** Patient 16 **(i)** Patient 17

Figure 6.14: Confidence distributions at each run of 5-fold CV for predicted BG using unbiased variance.

7 Conclusion and Future Work

7.1 Conclusion and Contributions

In this thesis, we answer the research questions formalized in the previous section. The contribution of this thesis is on providing effective methods for mining, ranking and recommedation in the Web.

Recommendation and Ranking for Web Search: In Chapter 3, we address the issues for (1) suggesting entity-centric queries and (2) ranking effectiveness surrounding the happening time period of an associated event. In particular, we propose an multi-objective optimization framework that faciliates the combination of multiple temporal models in (1) and a probabilistic approach for search result diversification of temporally ambiguous queries for (2).

Entity relatedness in Wikipedia: In Chapter 4, we study the long-term dynamics of Wikipedia as a global memory place for high-impact events, specifically the reviving memories of past events. Additionally, we propose a neural network-based model to measure the temporal relatedness of entities and events. The model engages different latent representations of an entities (i.e., from time, link-based graph and content) and use the *collective attention* from user navigation as the supervision.

Ranking in Web Archives: In Chapter 5, we tackle the problem of discovering important documents along the time-span of Web Archives, leveraging the link graph. Specifically, we combine the problems of relevance, temporal authority, diversity and time in a unified framework. The model accounts for the incomplete link structure and natural time lagging in Web Archives in mining the temporal authority.

Methods for *enhacing predictive models*: In Chapter 6, we investigate several methods to control model instability and enrich contexts of predictive models at the "cold-start" period. We demonstrate their effectiveness for the *rumor detection* and *blood glucose prediction* cases.

7.2 Open Research Directions

While we address several major issues with mining, ranking and recommending approaches for temporal IR, there are several issues that still need to be addressed. While we propose to enhance the effectiveness of ranking models in the re-ranking manner, the scalability and efficiency of such systems have not yet been touched in the scope of this thesis. How such models inference for new data in online context or can be designed efficiently or combine with lower-level components i.e., retrieval model is still an interesting open question. For Web Archives, while we exploit a graph-based method, how the full-text content and other text-based metadata could be used to improve ranking effectiveness as well as efficiency is also not addressed. We still do not have any actual query logs to fully understand the expert intents in Web Archives is another missing part to mention. We address the possible research directions as follows:

Approximate similarity (entity) search - For a given meaning space, searching for similar embeddings is one of the most basic operations in NLP and IR and can be applied to various applications, e.g., extracting synonyms, inferring the meanings of polysemous words, solving analogical reasoning questions, and searching for documents related to a query. How to quickly and accurately find similar embeddings in a continuous space is important from a practical standpoint, e.g., when we want to develop a real-time query expansion system on a search engine on the basis of an embedding similarity. Since word / entity embedding is often low-dimensional dense (in contrast to traditional sparse count vector), the inverted-index is not suitable for the task. [SKI16] studied different indexing strategies (i.e., hash-, tree- and graph-based) for this task and show their effectiveness for the task at different aspects. However, how time can be encoded for this task is still an interesting and important question.

(Time-aware) Bayesian neural ranking - Neural ranking is a very trending approach for IR in the hype of deep learning and have shown its effectiveness over many basic ranking tasks. However, such approaches always are data-hungry and needs large amount of training data / supervision. Thus, for advanced cases where supervision is not available at large-scale, such as for time-aware retrieval tasks in Web Archives or Expert-based systems, this *discriminative* approach has yet shown its advantages. For such cases, a *generative* approach that increases interpretability and handles uncertainty is considered to be more appropriate.

A
Curriculum Vitae

Tu Nguyen, born on 28 September 1986, in Hanoi, Vietnam.

Studies

11/2013 - 12/2018	Ph.D. studies. Gottfried Wilhelm Leibniz Universität Hannover, Germany.
04/2011 - 07/2013	M.Sc. in Computer Science. Gottfried Wilhelm Leibniz Universität Hannover, Germany.

Professional Experience

01/2019 - now **Data and applied scientist**	Daimler AG, Berlin, Germany Working as a applied scientist for machine learning / computer vision applications to autonomous services.
04/2011 - 12/2018	L3S Research Center, Gottfried Wilhelm Leibniz Universität Hannover, Germany.
Research staff member	EU Projects: ”QualiMaster”, ”Alexandria”, ”EUMSSI”, BMBF Project: ”Glycorec”.
Research assistant	BMBF Project ”ASEV”
06/2018 - 09/2018 **Research Intern**	IBM Research, Dublin, Ireland Collarborated as a visiting researcher in the Natural Language Processing Group, mainly responsible for developing deep learning methods for information extraction.

Bibliography

[AA12] Faraz Ahmed and Muhammad Abulaish. An mcl-based approach for spam profile detection in online social networks. In *Proceedings of TrustCom*, pages 602–608. IEEE, 2012.

[AB14] Nitish Aggarwal and Paul Buitelaar. Wikipedia-based distributional semantics for entity relatedness. In *2014 AAAI Fall Symposium Series*, 2014.

[AC95] Jan Assmann and John Czaplicka. Collective memory and cultural identity. *New German Critique*, (65):pp. 125–133, 1995.

[AGHI09a] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. Diversifying search results. In *Proceedings of WSDM '09*, 2009.

[AGHI09b] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 5–14. ACM, 2009.

[AHSW11] Sitaram Asur, Bernardo A. Huberman, Gábor Szabó, and Chunyan Wang. Trends in social media: Persistence and decay. In *Proceedings of ICWSM '11*, 2011.

[AP47] Gordon W Allport and Leo Postman. The psychology of rumor. 1947.

[AS91] J. R. Anderson and L. J. Schooler. Reflections of the environment in memory. *Psychological Science*, 6(2):396–408, 1991.

[ASVMNM10] R. Arun, V. Suresh, C. E. Veni Madhavan, and M. N. Narasimha Murthy. On finding the natural number of topics with latent dirichlet allocation: some observations. In *Proceedings of PAKDD '10*, 2010.

[AYJ11] Ching-man Au Yeung and Adam Jatowt. Studying how the past is remembered: towards computational history through large scale text mining. In *Proceedings of CIKM '11*, 2011.

[BB] Klaus Berberich and Srikanta Bedathur. Temporal diversification of search results. In *TAIA'2013*.

[BB13] Klaus Berberich and Srikanta Bedathur. Temporal diversification of search results. In *SIGIR 2013 Workshop on Time-aware Information Access (TAIA'2013)*, 2013.

[BCMT13] Roi Blanco, Berkant Barla Cambazoglu, Peter Mika, and Nicolas Torzec. Entity recommendations in web search. In *ISWC*, pages 33–48. Springer, 2013.

[BDDI] Krisztian Balog, Jeffrey Dalton, Antoine Doucet, and Yusra Ibrahim. Report on esair'15. In *ACM SIGIR Forum*.

[BF10] Luciano Barbosa and Junlan Feng. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of ACL*, pages 36–44, 2010.

[BGS05] Monica Bianchini, Marco Gori, and Franco Scarselli. Inside pagerank. *ACM Transactions on Internet Technology (TOIT)*, 5(1):92–128, 2005.

[BHM12] Javier Borge-Holthoefer and Yamir Moreno. Absence of influential spreaders in rumor dynamics. *Physical Review E*, 85(2):026116, 2012.

[BLL$^+$] Jiang Bian, Xin Li, Fan Li, Zhaohui Zheng, and Hongyuan Zha. Ranking specialization for web search: A divide-and-conquer approach by using topical ranksvm. In *WWW' 10*.

[BNJ03] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[Bre01] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[BSF94] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[BVW05] Klaus Berberich, Michalis Vazirgiannis, and Gerhard Weikum. Time-aware authority ranking. *Internet Mathematics*, 2(3):301–332, 2005.

[BYK] Ziv Bar-Yossef and Naama Kraus. Context-sensitive query auto-completion. In *WWW' 11*.

[BYXD13] Yuanyuan Bao, Chengqi Yi, Yibo Xue, and Yingfei Dong. A new rumor propagation model and control strategy on social networks. In *Proceedings of ICWSM*, pages 1472–1473. ACM, 2013.

[CBB10] Emanuele Coluccia, Carmela Bianco, and Maria A. Brandimonte. Autobiographical and event memories for surprising and unsurprising events. *Applied Cognitive Psychology*, 24(2):177–199, 2010.

[CC09] Ben Carterette and Praveen Chandar. Probabilistic models of ranking novel documents for faceted topic retrieval. In *Proceedings of CIKM '09*, 2009.

[CCS09] Charles L. A. Clarke, Nick Craswell, and Ian Soboroff. Overview of the TREC 2009 web track. In *TREC*, 2009.

[CCSV11] Charles L. A. Clarke, Nick Craswell, Ian Soboroff, and Ellen M. Voorhees. Overview of the TREC 2011 web track. In *TREC*, 2011.

[CDG$^+$13] Xue-Qi Cheng, Pan Du, Jiafeng Guo, Xiaofei Zhu, and Yixin Chen. Ranking on data manifold with sink points. *Knowledge and Data Engineering, IEEE Transactions on*, 25(1):177–191, 2013.

[CG98] Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of SIGIR '98*, 1998.

[CHL05] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 539–546. IEEE, 2005.

[CHWW12] Weiwei Cheng, Eyke Hüllermeier, Willem Waegeman, and Volkmar Welker. Label ranking with partial abstention based on thresholded probabilistic models. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2501–2509. Curran Associates, Inc., 2012.

[CKC$^+$] Charles LA Clarke, Maheedhar Kolla, Gordon V Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of SIGIR'2008*, pages 659–666.

[CLK$^+$16] Fernando Chirigati, Jialu Liu, Flip Korn, You Will Wu, Cong Yu, and Hao Zhang. Knowledge exploration using tables on the web. *Proceedings of the VLDB Endowment*, 2016.

[CLL$^+$17] Mauro Conti, Daniele Lain, Riccardo Lazzeretti, Giulio Lovisotto, and Walter Quattrociocchi. It's always april fools' day! on the difficulty of social network misinformation classification via propagation features. *CoRR*, abs/1701.04221, 2017.

[CLO$^+$13] Diego Ceccarelli, Claudio Lucchese, Salvatore Orlando, Raffaele Perego, and Salvatore Trani. Learning relatedness measures for entity linking. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 139–148. ACM, 2013.

[CMP11] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. Information credibility on twitter. In *Proceedings of WWW*, pages 675–684. ACM, 2011.

[CN10] Marek Ciglan and Kjetil Nørvåg. WikiPop: personalized event detection system based on Wikipedia page view statistics. In *Proceedings of CIKM '10*, 2010.

[COV$^+$17] Iván Contreras, Silvia Oviedo, Martina Vettoretti, Roberto Visentin, and Josep Vehí. Personalized blood glucose prediction: A hybrid approach using grammatical evolution and physiological models. *PloS one*, 12(11):e0187754, 2017.

[CS07] Nick Craswell and Martin Szummer. Random walks on the click graph. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 239–246, New York, NY, USA, 2007. ACM.

[CSHK09] Andrew R. A. Conway, Linda J. Skitka, Joshua A. Hemmerich, and Trina C. Kershaw. Flashbulb memory for 11 september 2001. *Applied Cognitive Psychology*, 23(5):605–623, 2009.

[CWL$^+$17] Tong Chen, Lin Wu, Xue Li, Jun Zhang, Hongzhi Yin, and Yang Wang. Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection. *arXiv preprint arXiv:1704.05973*, 2017.

[DA16] Andrea Dessi and Maurizio Atzori. A machine-learning approach to ranking rdf properties. *Future Generation Computer Systems*, 54:366–377, 2016.

[dat] Prediction intervals for random forests. http://blog.datadive.net/prediction-intervals-for-random-forests/. Accessed: 2017-17-09.

[DC] Van Dang and Bruce W Croft. Query reformulation using anchor text. In *Proceedings of WSDM'2010*.

[DDa] Na Dai and Brian D Davison. Freshness matters: in flowers, food, and web authority. In *Proceedings SIGIR'2010*, pages 114–121.

[DDb] Na Dai and Brian D Davison. Mining anchor text trends for retrieval. In *Proceedings of ECIR'2010*.

[DHC$^+$] Zhicheng Dou, Sha Hu, Kun Chen, Ruihua Song, and Ji-Rong Wen. Multi-dimensional search result diversification. In *Proceedings of WSDM'2011*.

[DHC$^+$11] Zhicheng Dou, Sha Hu, Kun Chen, Ruihua Song, and Ji-Rong Wen. Multi-dimensional search result diversification. In *Proceedings of WSDM '11*, 2011.

[DHH$^+$] Chris Ding, Xiaofeng He, Parry Husbands, Hongyuan Zha, and Horst Simon. Pagerank, hits and a unified framework for link analysis.

[DKL] Hongbo Deng, Irwin King, and Michael R. Lyu. Entropy-biased models for query representation on the click graph. In *Proceedings of SIGIR' 09*.

[DNLH16] Alexander Dallmann, Thomas Niebler, Florian Lemmerich, and Andreas Hotho. Extracting semantics from random walks on wikipedia: Comparing learning and counting methods. 2016.

[DOL15] Andrew M Dai, Christopher Olah, and Quoc V Le. Document embedding with paragraph vectors. *arXiv preprint arXiv:1507.07998*, 2015.

[DSW] Zhicheng Dou, Ruihua Song, and Ji-Rong Wen. A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of WWW' 07*.

[DXC] Van Dang, Xiaobing Xue, and W Bruce Croft. Inferring query aspects from reformulations using clustering. In *Proceedings of CIKM'2011*.

[DZF$^+$16] Bhuwan Dhingra, Zhong Zhou, Dylan Fitzpatrick, Michael Muehl, and William W Cohen. Tweet2vec: Character-based distributed representations for social media. *arXiv preprint arXiv:1605.03481*, 2016.

[Ebb85] Hermann Ebbinghaus. *Über das Gedächtnis. Untersuchungen zur experimentellen Psychologie*. Duncker & Humblot, Leipzig, 1885.

[EOCQS09] Meriyan Eren-Oruklu, Ali Cinar, Lauretta Quinn, and Donald Smith. Estimation of future glucose concentrations with subject-specific recursive linear models. *Diabetes technology & therapeutics*, 11(4):243–253, 2009.

[FAEC14] Adrien Friggeri, Lada A Adamic, Dean Eckles, and Justin Cheng. Rumor cascades. 2014.

[FBMB15] Lorenz Fischer, Roi Blanco, Peter Mika, and Abraham Bernstein. Timely semantics: a study of a stream-based ranking system for entity relationships. In *ISWC*, 2015.

[FM12] Michela Ferron and Paolo Massa. Psychological processes underlying wikipedia representations of natural and manmade disasters. In *Proceedings of WikiSym '12*, 2012.

[FMM$^+$17] Samuele Fiorini, Chiara Martini, Davide Malpassi, Renzo Cordera, Davide Maggi, Alessandro Verri, and Annalisa Barla. Data-driven strategies for robust forecast of continuous glucose monitoring time-series. In *Engineering in Medicine and Biology Society (EMBC), 2017 39th Annual International Conference of the IEEE*, pages 1680–1683. IEEE, 2017.

[FOM$^+$14] Michele Fenzi, Jörn Ostermann, Nico Mentzer, Guillermo Payá-Vayá, Holger Blume, Tu Ngoc Nguyen, and Thomas Risse. Asev—automatic situation assessment for event-driven video analysis. In *Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on*, pages 37–43. IEEE, 2014.

[GB10] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[GB14] Zhaochen Guo and Denilson Barbosa. Robust entity linking via random walks. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 499–508. ACM, 2014.

[GEW06] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

[GFAC16] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 55–64. ACM, 2016.

[GKCM14] Aditi Gupta, Ponnurangam Kumaraguru, Carlos Castillo, and Patrick Meier. Tweetcred: Real-time credibility assessment of content on twitter. In *SocInfo*. Springer, 2014.

[GKK$^+$13] Mihai Georgescu, Nattiya Kanhabua, Daniel Krause, Wolfgang Nejdl, and Stefan Siersdorfer. Extracting event-related information from article updates in wikipedia. In *Proceedings of ECIR '13*, 2013.

[GM07] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 1606–1611, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[GM09] Evgeniy Gabrilovich and Shaul Markovitch. Wikipedia-based semantic interpretation for natural language processing. *Journal of Artificial Intelligence Research*, 34:443–498, 2009.

[Halon] Maurice Halbwachs. *On collective memory*. The University of Chicago Press, Chicago, 1950 (Translation).

[HBB] Faegheh Hasibi, Krisztian Balog, and Svein Erik Bratsberg. Dynamic factual summaries for entity cards. In *SIGIR'17*.

[HGSK09] Geremy Heitz, Stephen Gould, Ashutosh Saxena, and Daphne Koller. Cascaded classification models: Combining models for holistic scene understanding. In *NIPS*, 2009.

[HHD$^+$15] Zhiting Hu, Poyao Huang, Yuntian Deng, Yingkai Gao, and Eric Xing. Entity hierarchy embedding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1292–1300, 2015.

[HHG$^+$13] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 2333–2338. ACM, 2013.

[HPB$^+$09] W Hirst, E.A Phelps, RL Buckner, A.E Budson, A Cuc, J.D.E Gabrieli, M.K Johnson, C Lustig, K.B Lyle, M Mather, R Meksin, K.J. Mitchell, K. N. Ochsner, D.L. Schacter, J.S Simons, and C.J. Valdya. Long-term memory for the terrorist attack of september 11: Flashbulb memories, event memories, and the factors that influence their retention. *Journal of Experimental Psychology: General*, 138(2):161–76, 2009.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[HSN$^+$12] Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. Kore: keyphrase overlap relatedness for entity disambiguation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 545–554. ACM, 2012.

[Int] Introduction to optimization. https://ipvs.informatik.uni-stuttgart.de/mlr/marc/teaching/13-Optimization/04-secondOrderOpt.pdf.

[IS07] Ilse CF Ipsen and Teresa M Selee. Pagerank computation, with special attention to dangling nodes. *SIAM Journal on Matrix Analysis and Applications*, 29(4):1281–1296, 2007.

[IS15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[JD07] Rosie Jones and Fernando Diaz. Temporal profiles of queries. *ACM Trans. Inf. Syst.*, 25, July 2007.

[JDS$^+$13] Fang Jin, Edward Dougherty, Parang Saraf, Yang Cao, and Naren Ramakrishnan. Epidemiological modeling of news and rumors on twitter. In *Proceedings of SNA-KDD*, 2013.

[JGBM16] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[JLG$^+$16] Tingsong Jiang, Tianyu Liu, Tao Ge, Lei Sha, Baobao Chang, Sujian Li, and Zhifang Sui. Towards time-aware knowledge graph completion. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1715–1724, 2016.

[Joa06] Thorsten Joachims. Training linear svms in linear time. In *Proceedings of KDD '06*, 2006.

[KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[KCJ$^+$13] Sejeong Kwon, Meeyoung Cha, Kyomin Jung, Wei Chen, and Yajun Wang. Prominent features of rumor propagation in online social media. In *Proceedings of ICDM*, 2013.

[KGC12] Brian Keegan, Darren Gergle, and Noshir Contractor. Staying in the loop: Structure and dynamics of wikipedia's breaking news collaborations. In *Proceedings of WikiSym '12*, 2012.

[KGC13] Brian Keegan, Darren Gergle, and Noshir Contractor. Hot off the wiki - structures and dynamics of Wikipedia's coverage of breaking news events. *American Behavioral Scientist*, 57(5):595–622, 2013.

[Kim14] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[Kim15] David Kimmey. Twitter event detection. 2015.

[KKN+16] Nattiya Kanhabua, Philipp Kemkes, Wolfgang Nejdl, Tu Ngoc Nguyen, Felipe Reis, and Nam Khanh Tran. How to search the internet archive without indexing it. In *Research and Advanced Technology for Digital Libraries - 20th International Conference on Theory and Practice of Digital Libraries, TPDL 2016, Hannover, Germany, September 5-9, 2016, Proceedings*, pages 147–160, 2016.

[KLL+] Weize Kong, Rui Li, Jie Luo, Aston Zhang, Yi Chang, and James Allan. Predicting search intent based onl pre-search context. In *SIGIR '15*.

[KMT+13] Sanjay Ram Kairam, Meredith Ringel Morris, Jaime Teevan, Daniel J. Liebling, and Susan T. Dumais. Towards supporting search over trending events with social media. In *ICWSM*, 2013.

[KNa] Nattiya Kanhabua and Wolfgang Nejdl. On the value of temporal anchor texts in wikipedia. In *TAIA'2014*.

[KNb] Nattiya Kanhabua and Kjetil Nørvåg. Determining time of queries for re-ranking search results. In *Proceedings of ECDL'2010*.

[KN10] Nattiya Kanhabua and Kjetil Nørvåg. Determining time of queries for re-ranking search results. In *Proceedings of ECDL'2010*, 2010.

[KNN14a] Nattiya Kanhabua, Tu Ngoc Nguyen, and Claudia Niederée. What triggers human remembering of events? A large-scale analysis of catalysts for collective memory in wikipedia. In *IEEE/ACM Joint Conference on Digital Libraries, JCDL 2014, London, United Kingdom, September 8-12, 2014*, pages 341–350, 2014.

[KNN14b] Nattiya Kanhabua, Tu Ngoc Nguyen, and Claudia Niederée. What triggers human remembering of events? a large-scale analysis of catalysts for collective memory in wikipedia. In *Digital Libraries (JCDL), 2014 IEEE/ACM Joint Conference on*, pages 341–350. IEEE, 2014.

[KNNN15] Nattiya Kanhabua, Tu Ngoc Nguyen, and Wolfgang Nejdl. Learning to detect event-related queries for web search. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1339–1344. ACM, 2015.

[KRM16] Nattiya Kanhabua, Huamin Ren, and Thomas B. Moeslund. Learning dynamic classes of events using stacked multilayer perceptron networks. *CoRR*, abs/1606.07219, 2016.

[KSLP$^+$] Shubhra Kanti Karmaker Santu, Liangda Li, Dae Hoon Park, Yi Chang, and Chengxiang Zhai. Modeling the influence of popular trending events on user search behavior. In *WWW '17*.

[KTSD] Anagha Kulkarni, Jaime Teevan, Krysta M. Svore, and Susan T. Dumais. Understanding temporal query dynamics. In *Proceedings of WSDM' 11*.

[KYZ$^+$15] Changsung Kang, Dawei Yin, Ruiqiang Zhang, Nicolas Torzec, Jianzhang He, and Yi Chang. Learning to rank related entities in web search. *Neurocomputing*, 166:309–318, 2015.

[LF06] Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 631–636. ACM, 2006.

[LGA17] Tao Lin, Tian Guo, and Karl Aberer. Hybrid neural networks for learning the trend in time series. 2017.

[LGRC] Janette Lehmann, Bruno Gonçalves, José J. Ramasco, and Ciro Cattuto. Dynamical classes of collective attention in twitter. In *Proceedings of WWW' 12*.

[LKF07] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):2, 2007.

[LL13] Zhengdong Lu and Hang Li. A deep architecture for matching short texts. In *Advances in Neural Information Processing Systems*, pages 1367–1375, 2013.

[LM14] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196, 2014.

[LNL$^+$15] Xiaomo Liu, Armineh Nourbakhsh, Quanzhi Li, Rui Fang, and Sameena Shah. Real-time rumor debunking on twitter. In *Proceedings of CIKM*, pages 1867–1870. ACM, 2015.

[LPG$^+$] Thomas Lin, Patrick Pantel, Michael Gamon, Anitha Kannan, and Ariel Fuxman. Active objects: Actions for entity-centric search. In *WWW'12*.

[LPM15] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[MBL15] Iris Miliaraki, Roi Blanco, and Mounia Lalmas. From selena gomez to marlon brando: Understanding explorative entity search. In *Proceedings of the 24th International Conference on World Wide Web*, pages 765–775. International World Wide Web Conferences Steering Committee, 2015.

[MC17] Bhaskar Mitra and Nick Craswell. Neural models for information retrieval. *arXiv preprint arXiv:1705.01509*, 2017.

[MGM$^+$] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. Detecting rumors from microblogs with recurrent neural networks.

[MGR] Qiaozhu Mei, Jian Guo, and Dragomir Radev. Divrank: the interplay of prestige and diversity in information networks. In *Proceedings of SIGKDD'2010*, pages 1009–1018.

[MGW$^+$15] Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu, and Kam-Fai Wong. Detect rumors using time series of social context information on microblogging websites. In *Proceedings of CIKM*, 2015.

[MM31] John A. McGeoch and William T. McDonald. Meaningful relation and retroactive inhibition. *American Journal of Psychology*, 43(4):579–588, 1931.

[MMJ13] Andrew J McMinn, Yashar Moshfeghi, and Joemon M Jose. Building a large-scale corpus for evaluating event detection on twitter. In *Proceedings of CIKM*, 2013.

[MNR$^+$15] Polykarpos Meladianos, Giannis Nikolentzos, François Rousseau, Yannis Stavrakas, and Michalis Vazirgiannis. Degeneracy-based real-time subevent detection in twitter stream. In *Proceedings of ICWSM*, 2015.

[MPC10] Marcelo Mendoza, Barbara Poblete, and Carlos Castillo. Twitter under crisis: can we trust what we rt? In *Proceedings of the first workshop on social media analytics*, pages 71–79. ACM, 2010.

[MRN14] Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244, 2014.

[MSC$^+$13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[MSP$^+$12] Yasuko Matsubara, Yasushi Sakurai, B Aditya Prakash, Lei Li, and Christos Faloutsos. Rise and fall patterns of information diffusion: model and implications. In *Proceedings of KDD*. ACM, 2012.

[Ngu17] Tu Nguyen. A comprehensive low and high-level feature analysis for early rumor detection on twitter. *arXiv preprint arXiv:1711.00726*, 2017.

[NK14] Tu Ngoc Nguyen and Nattiya Kanhabua. Leveraging dynamic query subtopics for time-aware search result diversification. In *European Conference on Information Retrieval*, pages 222–234. Springer, 2014.

[NKN18a] Tu Ngoc Nguyen, Nattiya Kanhabua, and Wolfgang Nejdl. Multiple models for recommending temporal aspects of entities. In Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna Tordai, and Mehwish Alam, editors, *The Semantic Web*, pages 462–480, Cham, 2018. Springer International Publishing.

[NKN18b] Tu Ngoc Nguyen, Nattiya Kanhabua, and Wolfgang Nejdl. Multiple models for recommending temporal aspects of entities. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, pages 462–480, 2018.

[NKNN] Tu Ngoc Nguyen, Nattiya Kanhabua, Wolfgang Nejdl, and Claudia Niederée. Mining relevant time for query subtopics in web archives. In *Proceedings of WWW'2015 companion*, pages 1357–1362.

[NKNN15] Tu Ngoc Nguyen, Nattiya Kanhabua, Wolfgang Nejdl, and Claudia Niederée. Mining relevant time for query subtopics in web archives. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 1357–1362, New York, NY, USA, 2015. ACM.

[NKNZ15] Tu Ngoc Nguyen, Nattiya Kanhabua, Claudia Niederée, and Xiaofei Zhu. A time-aware random walk model for finding important documents in web archives. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 915–918, New York, NY, USA, 2015. ACM.

[NLN17] Tu Ngoc Nguyen, Cheng Li, and Claudia Niederée. On early-stage debunking rumors on twitter: Leveraging the wisdom of weak learners. In *Social Informatics - 9th International Conference, SocInfo 2017, Oxford, UK, September 13-15, 2017, Proceedings, Part II*, pages 141–158, 2017.

[NR18] Tu Nguyen and Markus Rokicki. On the predictability of non-cgm diabetes data for personalized recommendation. *arXiv preprint arXiv:1808.07380*, 2018.

[NS13] Tu Ngoc Nguyen and Wolf Siberski. Slubm: an extended lubm benchmark for stream reasoning. In *Proceedings of the 2nd International Conference on Ordering and Reasoning-Volume 1059*, pages 43–54. CEUR-WS. org, 2013.

[NTN18] Tu Ngoc Nguyen, Tuan Tran, and Wolfgang Nejdl. A trio neural model for dynamic entity relatedness ranking. In *Proceedings of the 22nd SIGNLL Conference on Computational Natural Language Learning*, pages 297–300. ACL, 2018.

[NXC$^+$16] Yuan Ni, Qiong Kai Xu, Feng Cao, Yosi Mass, Dafna Sheinwald, Hui Jia Zhu, and Shao Sheng Cao. Semantic documents relatedness using concept graph representation. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16, pages 635–644, New York, NY, USA, 2016. ACM.

[OPPSS16] Liudmila Ostroumova Prokhorenkova, Petr Prokhorenkov, Egor Samosvat, and Pavel Serdyukov. Publication date prediction through reverse engineering of the web. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 123–132. ACM, 2016.

[OR16] Francisco Javier Ordóñez and Daniel Roggen. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115, 2016.

[PARS14] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

[PBM$^+$14] Kevin Plis, Razvan Bunescu, Cindy Marling, Jay Shubrook, and Frank Schwartz. A machine learning approach to predicting blood glucose levels for diabetes management. 2014.

[PBMW99] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. 1999.

[Pen09] Christian Pentzold. Fixing the floating gap: The online encyclopaedia wikipedia as a global memory place. *Memory Studies*, 2(2):255–272, 2009.

[PFC17] Marco Ponza, Paolo Ferragina, and Soumen Chakrabarti. A two-stage framework for computing entity relatedness in wikipedia. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, pages 1867–1876, New York, NY, USA, 2017. ACM.

[PMZ] Jeffrey Pound, Peter Mika, and Hugo Zaragoza. Ad-hoc object retrieval in the web of data. In *WWW'10*.

[QRRM11] Vahed Qazvinian, Emily Rosengren, Dragomir R Radev, and Qiaozhu Mei. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of EMNLP*, 2011.

[RBS10] Davood Rafiei, Krishna Bharat, and Anand Shukla. Diversifying web search results. In *Proceedings of WWW '10*, 2010.

[RMdR15] Ridho Reinanda, Edgar Meij, and Maarten de Rijke. Mining, ranking and recommending entity aspects. In *Proceedings of SIGIR*, pages 263–272. ACM, 2015.

[RSD$^+$] Kira Radinsky, Krysta Svore, Susan Dumais, Jaime Teevan, Alex Bocharov, and Eric Horvitz. Modeling and predicting behavioral dynamics on the web. In *Proceedings of WWW '12*.

[SFC10] Giovanni Sparacino, Andrea Facchinetti, and Claudio Cobelli. "smart" continuous glucose monitoring sensors: on-line signal processing issues. *Sensors*, 10(7):6751–6772, 2010.

[SGJ11] Jannik Strötgen, Michael Gertz, and Conny Junghans. An event-centric model for multilingual document similarity. In *Proceedings of SIGIR '11*, 2011.

[Sho] Milad Shokouhi. Detecting seasonal queries by time-series analysis. In *SIGIR' 11*.

[Sil10] Fabrizio Silvestri. Mining query logs: Turning search usage data into knowledge. *Foundations and Trends in Information Retrieval*, 4(1-2):1–174, 2010.

[SKI16] Kohei Sugawara, Hayato Kobayashi, and Masajiro Iwasaki. On approximately searching for similar word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2265–2275, 2016.

[SMA12] Eunsoo Seo, Prasant Mohapatra, and Tarek Abdelzaher. Identifying rumors and their sources in social networks. In *SPIE*, 2012.

[SMDR$^+$] Damiano Spina, Edgar Meij, Maarten De Rijke, Andrei Oghina, Minh Thuong Bui, and Mathias Breuss. Identifying entity aspects in microblog posts. In *SIGIR'12*.

[SMO10] Rodrygo L.T. Santos, Craig Macdonald, and Iadh Ounis. Exploiting query reformulations for web search result diversification. In *Proceedings of WWW '10*, 2010.

[SR] Milad Shokouhi and Kira Radinsky. Time-sensitive query auto-completion. In *IGIR '12*.

[SRVS11] Andrey Styskin, Fedor Romanenko, Fedor Vorobyev, and Pavel Serdyukov. Recency ranking by diversification of result set. In *Proceedings of CIKM '11*, 2011.

[Sun14] Cass R Sunstein. *On rumors: How falsehoods spread, why we believe them, and what can be done*. Princeton University Press, 2014.

[SW12] Marc Spaniol and Gerhard Weikum. Tracking entities in web archives: the lawa project. In *Proceedings of the 21st International Conference on World Wide Web*, pages 287–290. ACM, 2012.

[SZG$^+$11] Wei Song, Yu Zhang, Handong Gao, Ting Liu, and Sheng Li. HITSCIR system in NTCIR-9 subtopic mining task, 2011.

[SZG$^+$14] Wei Song, Yu Zhang, Handong Gao, Ting Liu, and Sheng Li. HITSCIR system in NTCIR-9 subtopic mining task. 2014.

[TANN] Giang Tran, Mohammad Alrifai, Tu Ngoc Nguyen, and Wolfgang Nejdl. Wikitimess knowledge extraction and enrichment process.

[TBM10] Rudra M Tripathy, Amitabha Bagchi, and Sameep Mehta. A study of rumor control strategies on social networks. In *Proceedings of CIKM*, pages 1817–1820. ACM, 2010.

[TN14] Tuan Tran and Tu Ngoc Nguyen. Hedera: scalable indexing and exploring entities in wikipedia revision history. In *Proceedings of the 2014 International Conference on Posters & Demonstrations Track-Volume 1272*, pages 297–300. CEUR-WS. org, 2014.

[TTN] Nam Khanh Tran, Tuan Tran, and Claudia Niederée. Beyond time: Dynamic context-aware entity recommendation. In *ESWC'17*.

[Tul02] Endel Tulving. Episodic memory: From mind to brain. *Annual review of psychology*, 53(1):1–25, 2002.

[Und57] Benton Underwood. Interference and forgetting. *Psychological Review*, 64(1):49–60, 1957.

[VSP$^+$17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[VTN$^+$16] Khoi Duy Vo, Tuan Tran, Tu Ngoc Nguyen, Xiaofei Zhu, and Wolfgang Nejdl. Can we find documents in web archives without knowing their contents? In *Proceedings of the 8th ACM Conference on Web Science*, pages 173–182. ACM, 2016.

[VTS16] S. Vadrevu, Y. Tu, and F. Salvetti. Ranking relevant attributes of entity in structured knowledge base, January 5 2016. US Patent 9,229,988.

[VVR16] Soroush Vosoughi, Prashanth Vijayaraghavan, and Deb Roy. Tweet2vec: Learning tweet embeddings using character-level cnn-lstm encoder-decoder. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 1041–1044. ACM, 2016.

[Wan10] Alex Hai Wang. Don't follow me: Spam detection in twitter. In *Proceedings of SECRYPT*, pages 1–10. IEEE, 2010.

[WHE14] Stefan Wager, Trevor Hastie, and Bradley Efron. Confidence intervals for random forests: the jackknife and the infinitesimal jackknife. *Journal of Machine Learning Research*, 15(1):1625–1651, 2014.

[WJ] Stewart Whiting and Joemon M. Jose. Recent and robust query auto-completion. In *WWW '14*.

[WM08] Ian H Witten and David N Milne. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. 2008.

[WNS$^+$] Gerhard Weikum, Nikos Ntarmos, Marc Spaniol, Peter Triantafillou, András A Benczúr, Scott Kirkpatrick, Philippe Rigaux, and Mark Williamson. Longitudinal analytics on web archive data: It's about time!

[WYZ15] Ke Wu, Song Yang, and Kenny Q Zhu. False rumors detection on sina weibo by propagation structures. In *Proceedings of ICDE*, pages 651–662. IEEE, 2015.

[WZJL13] Stewart Whiting, Ke Zhou, Joemon Jose, and Mounia Lalmas. Temporal variance of intents in multi-faceted event-driven information needs. In *Proceedings of SIGIR '13*, 2013.

[YAGJ14] Taha Yasseri, Spoerri Anselm, Mark Graham, and Kertész Janos. The most controversial topics in wikipedia: A multilingual and geographical analysis. In P. Fichman and N Hara, editors, *Global Wikipedia: International and cross-cultural issues in online collaboration*. Scarecrow Press, 2014.

[YLL] Philip S Yu, Xin Li, and Bing Liu. On the temporal dimension of search. In *Proceedings of WWW'2004*, pages 448–449.

[YLYY12] Fan Yang, Yang Liu, Xiaohui Yu, and Min Yang. Automatic detection of rumor on sina weibo. In *Proceedings of MDS*. ACM, 2012.

[YMHH14] Xiao Yu, Hao Ma, Bo-June Paul Hsu, and Jiawei Han. On building entity recommender systems using user click log and freebase knowledge. In *Proceedings of WSDM*, pages 263–272. ACM, 2014.

[YQZ$^+$] Lei Yang, Lei Qi, Yan-Ping Zhao, Bin Gao, and Tie-Yan Liu. Link analysis using time series of web graphs. In *Proceedings of CIKM'2007*, pages 1011–1014.

[YSXZ16] Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association of Computational Linguistics*, 4(1):259–272, 2016.

[ZGVGA] Xiaojin Zhu, Andrew B Goldberg, Jurgen Van Gael, and David Andrzejewski. Improving diversity in ranking using absorbing random walks. In *Proceedings of HLT-NAACL'2007*, pages 97–104.

[ZLC$^+$14] Yi Zheng, Qi Liu, Enhong Chen, Yong Ge, and J Leon Zhao. Time series classification using multi-channels deep convolutional neural networks. In *International Conference on Web-Age Information Management*, pages 298–310. Springer, 2014.

[ZLS15] Yu Zhao, Zhiyuan Liu, and Maosong Sun. Representation learning for measuring entity relatedness with rich information. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[ZRM15] Zhe Zhao, Paul Resnick, and Qiaozhu Mei. Enquiring minds: Early detection of rumors in social media from enquiry posts. In *Proceedings of WWW*, 2015.

[ZRZ16] Lei Zhang, Achim Rettinger, and Ji Zhang. A probabilistic model for time-aware entity recommendation. In *ISWC*, pages 598–614. Springer, 2016.

[ZSLL15] Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis Lau. A c-lstm neural network for text classification. *arXiv preprint arXiv:1511.08630*, 2015.

[ZWJL13] Ke Zhou, Stewart Whiting, Joemon M. Jose, and Mounia Lalmas. The impact of temporal intent variability on diversity evaluation. In *Proceedings of ECIR '13*, 2013.

[ZYL$^+$16] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 353–362. ACM, 2016.