Just-In-Time Information Retrieval and Summarization for Personal Assistance

Doctoral Dissertation submitted to the
Faculty of Informatics of the Università della Svizzera Italiana
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

presented by
Seyed Ali Bahrainian

under the supervision of
Fabio Crestani

August 2019

Dissertation Committee

Cathal Gurrin	Dublin City University, Dublin, Ireland
Marc Langheinrich	Università della Svizzera italiana, Lugano, Switzerland
David Losada	Universidade de Santiago de Compostela, Galicia, Spain
Fernando Pedone	Università della Svizzera italiana, Lugano, Switzerland

Dissertation accepted on 18th August 2019

Research Advisor	PhD Program Director
Fabio Crestani	**Walter Binder**

I certify that except where due acknowledgement has been given, the work presented in this thesis is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; and the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program.

Seyed Ali Bahrainian
Lugano, 18th August 2019

To my parents

Those who can imagine anything,
can create the impossible.
Alan Turing

Abstract

With the rapid development of means for producing user-generated data opportunities for collecting such data over a time-line and utilizing it for various human-aid applications are more than ever. Wearable and mobile data capture devices as well as many online data channels such as search engines are all examples of means of user data collection. Such user data could be utilized to model user behavior, identify relevant information to a user and retrieve it in a timely fashion for personal assistance.

User data can include recordings of one's conversations, images, biophysical data, health-related data captured by wearable devices, interactions with smartphones and computers, and more.

In order to utilize such data for personal assistance, summaries of previously recorded events can be presented to a user in order to augment the user's memory, send notifications about important events to the user, predict the user's near-future information needs and retrieve relevant content even before the user asks.

In this PhD dissertation, we design a personal assistant with a focus on two main aspects:

The first aspect is that a personal assistant should be able to summarize user data and present it to a user. To achieve this goal, we build a Social Interactions Log Analysis System (SILAS) that summarizes a person's conversations into event snippets consisting of spoken topics paired with images and other modalities of data captured by the person's wearable devices. Furthermore, we design a novel discrete Dynamic Topic Model (dDTM) capable of tracking the evolution of the intermittent spoken topics over time. Additionally, we present the first neural Customizable Abstractive Topic-based Summarization (CATS) model that produces summaries of textual documents including meeting transcripts in the form of natural language.

The second aspect that a personal assistant should be capable of, is proactively addressing the user's information needs. For this purpose, we propose a family of just-in-time information retrieval models such as an evolutionary model named Kalman combination of Recency and Establishment (K2RE) that can anticipate a

user's near-future information needs. Such information needs can include information for preparing a future meeting or near-future search queries of a user.

Acknowledgements

I am very grateful to my advisor Fabio Crestani for his constant support and guidance through my PhD. Fabio taught me how to be an independent researcher and gave me many great opportunities to grow as a researcher, and as a person. He supported and encouraged me to participate in several research visits that benefited my research and this thesis. I appreciate his continuous strive for making my PhD a success.

I would like to thank Marc Langheinrich, Cathal Gurrin, David Losada and Fernando Pedone for accepting to be on my PhD committee and providing me with their invaluable advice.

Marc Langheinrich was one of my mentors in the RECALL project. He constantly supported me with his invaluable advice and encouragement. I am grateful to Marc for his guidance throughout the project, advising me on human-computer interaction and privacy topics. He supported the collection of a dataset that greatly contributed to my PhD.

Cathal Gurrin invited me to visit his lab at Dublin City University and generously introduced me to the exciting lifelogging research in his lab. I learned a lot during this visit and was privileged to have invaluable discussions with a pioneer of lifelogging research which significantly influenced my thoughts and my thesis.

I would like to thank Carsten Eickhoff who welcomed me to his AI lab at Brown University. I am grateful for the amazing experience and the opportunity to work on cutting-edge research as a part of his team.

Pursuing a PhD can be difficult and with many ups and downs. I am thankful to the IR lab members at USI for their friendship and support during this time and many valuable scientific discussions. I also would like to thank my co-authors who helped improve my research.

Finally, I would like to thank my family for their love and support along the way. I would like to thank them all but more than anyone else my parents to whom I dedicate this thesis.

Contents

Figures

Tables

Chapter 1

Introduction

1.1 Motivation

With the rapid increase of the amount of information that people are exposed to on a daily basis, the need for utilizing new technologies to store, organize, and selectively retrieve relevant information at just the right time a user needs them becomes evident. Retrieval systems that can autonomously assist people to access their desired information at their finger tips and avoid time-consuming searches are of strong importance. The ultimate manifestation of information at your finger tips will be the use of mobile and ubiquitous computing to assist people address their information needs.

Virtual personal assistants are software agents that aim at assisting users with their information needs in a personalized and timely fashion. With the increasing popularity of smartphones, personal assistants have been built to operate on both stationary and mobile computing devices and assist users on-the-go. Personal assistants can benefit from personal lifelog datasets to assist a user in sophisticated ways [15]. Lifelogging represents a phenomenon whereby people can digitally record their own daily lives in varying amounts of detail, for different applications [60]. One important application of lifelogging is building personal assistants that can analyze various user-generated data modalities and result in much more powerful personal assistants than exists today.

Personal assistants are beneficial for personal information management (i.e., acquiring, organizing, retrieving and using information recorded about an individual), showing users automatic just-in-time reminders and notifications to address various information needs and assist users in accomplishing tasks. Such reminders and notifications can be presented to a user in the form of push notifications on smartphones, or via smart glasses (as in Figure 1.1), or via natural

Figure 1.1. Smart glasses

language voice notifications.

One early personal assistant was the Remembrance Agent [108] from MIT labs. This assistant was designed to augment human memory "by displaying a list of documents which might be relevant to the user's current context" without any user intervention. A mobile version of this system named the wearable Remembrance Agent was later introduced. The wearable Remembrance Agent utilized contextual information on-the-go, to carry out memory assistance tasks [109].

Rhodes et al. [107] defined Iust-In-Time Information Retrieval (JITIR) systems as *"software agents that proactively present potentially valuable information based on a person's local context in an easily accessible yet non-intrusive manner"*.

According to the above quote we derive two main notions that a personal assistant would be based on:

1. A personal assistant extracts useful information from unstructured user data and presents it to a user in an easy-to-understand and non-intrusive fashion.

2. Such assistant must be able to anticipate users' information needs and present relevant information to the users proactively in a JITIR manner.

These two notions inspire the new paradigm of personal assistants such as Google Now, Microsoft Cortana, or Apple's Siri that seek to "offer proactive experiences that aim to recommend the right information at just the right time and help you get things done, even before you ask" [127]. These personal assistants are increasingly built into various platforms such as smartphones, desktop computers as well as smart home devices. Such assistants aim at communicating with users by presenting them relevant content in the form of notifications but also via natural language dialogues.

We focus on both of these underlying notions of personal assistants in this PhD thesis. We present various personal assistant systems to process and analyze various modalities of personal data that together would be used to build a comprehensive personal assistant. To address problems related to the first notion, we present models for extracting the salient themes of meetings and to be presented to potential users in the form of event snippets evolving over time. Moreover, we present the first customizable abstractive summarization model that can selectively include certain topics or exclude them from generated output summaries. In order to address problems related to the second notion, we present a number of JITIR models that anticipate a user's information needs and address them in a timely fashion. These models are designed to anticipate the topics that one will discuss in a future meeting, and query topic that one searches for on a search engine. We elaborate on all these models in this thesis.

My PhD research started with the European Future and Emerging Technologies (FET) project named RECALL. This project, which aimed at building systems that augment and assist human memory, was the initial source of inspiration for this PhD thesis. The RECALL project strived to re-define the notion of memory augmentation. It relied on the recent developments in wearable sensors that capture various modalities of one's life and advancements in Information Retrieval (IR) to utilize continuous and automated recordings of many aspects of our everyday lives for assistance in accomplishing tasks. One important characteristic of the RECALL project was its aim to ameliorate memory augmentation technologies to present a departure from expensive clinical memory assistance technology to a mainstream technology, that would benefit users in their every day lives. The expected outcome of this project was mechanisms to automate the acquisition of personal memories, analyze those memories to present meaningful assistance to users, and help users by training their memories to live healthier and more fulfilling lives.

1.2 Thesis Aims and Objectives

Following the two main themes that we discussed in the previous section, my thesis is organized in two main parts:

Part I presents methods for extracting various patterns from unstructured user data, such as meetings transcripts, and showing them to the user in an easy-to-understand summary. To accomplish this, we designed a Social Interactions Log Analysis System (SILAS) that can retrieve summaries of the user's personal memories that if presented to the user can augment the user's memory and facilitate the recall of past events. These summaries are event snippets consisting of images captured by the user's wearable camera paired with topics which were spoken words extracted from transcribed text. However, in a subsequent effort, we go beyond these static topics and present a new temporal model that captures the evolution of a topic over time and chains similar topics together, facilitating exploring the discussed topics over time. We named this model the discrete Dynamic Topic Model (dDTM) as it tracks the evolution of intermittent topics over time. Additionally, we also design a customizable text summarization model which generates summaries in the form of natural language, allowing for transcripts of meetings being presented in a condensed version of the original transcript highlighting the gist of a meeting.

Part II presents novel models capable of anticipating users' near-future information needs and present relevant information to the users proactively in a JITIR manner. We designed a system that utilized biophysical sensor data as well as the sentiment signals expressed in transcriptions of meetings to predict the segments of a conversation that one is likely to forget within a week time. This research empirically showed that forgetting is often predictable, as expressed sentiment as well as biophysical sensor data can reveal whether a person is attentive during a meeting. Furthermore, we introduced several novel JITIR methods that, given the history of meetings of two individuals, aim at predicting what topics will be discussed in the next meeting. The initial results on meetings were promising and since just-in-time retrieval of information could have various applications with a range of user data modalities, we extended this work to other domains such as predicting the topic of one's next search query. Meetings datasets are very expensive and time-consuming to build. Specially, when it comes to gathering real-world meetings datasets, finding volunteer participants becomes very difficult. To carry out the experiments validating the efficacy of the proposed models we (a group of three students and two professors) gathered a dataset of meetings at USI. Despite the huge effort and time this dataset contains 28 meetings. Moreover, we went beyond only using our own dataset and gathered other existing datasets of meetings from previous research. After combining the available meetings from two different datasets (each involving a group of several people taking over a year to gather) the total number of meetings in our collective meetings dataset was a few hundred. To verify the performance and correctness of our proposed models rigorously, we used large datasets from other domains. For example, news data was used to extensively evaluate our proposed dDTM. Furthermore, conference proceedings and query logs were used to evaluate our JITIR models.

1.3 Thesis Outline

The outline of this thesis is as follows: As discussed the thesis is organized in two parts. Prior to the two parts, Chapter 2 presents related work.

Part I presents methods for extracting patterns from unstructured user data such as meetings transcripts and showing them to the user in an easy-to-understand summary. Part I includes the following three chapters:

In Chapter 3, we present SILAS, a system that retrieves summaries of a user's personal memories to be presented to the user as a form of memory augmentation to facilitate recall of past events. The system summarizes a meeting/conversation into event snippets containing images taken by the user's wearable camera paired with the corresponding topics of the meeting/conversation. We evaluate the effect of using SILAS in two user studies and show that reviewing events using SILAS can augment and assist human memory.

In Chapter 4, we present the dDTM topic model.

Definition 1. *Topic models are by definition hierarchical Bayesian models of discrete data [145], where each topic is a set of words, drawn from a fixed vocabulary, such that together they represent a high level concept.*

The dDTM captures topics of a temporal dataset and connects similar topics together over time. We evaluate dDTM, both in terms of standard metrics as well as qualitatively.

In Chapter 5, we present CATS, a Customizable Abstractive Topical Summarization. CATS is the first summarization model that can customize the generated summaries by including only certain topics of interest in the summaries. We evaluate CATS and show that it has several useful features and outperforms state of the art in terms of standard evaluation metrics.

Part II presents several models capable of anticipating users' near-future information needs and present relevant information to the users proactively in a JITIR manner. This part is organized in three Chapters as follows:

In Chapter 6, we present a model for a futuristic application that predicts the segments of a meeting/conversation that one is likely to forget within a week time. This model extracts the expressed sentiment as well as biophysical patterns derived from a wearable sensor to predict forgetting.

In Chapter 7, we present a family of JITIR models for predicting the near-future in two applications. We present models that, given the history of one's meetings strive to predict the topics that continue in a future meeting. This is achieved by tracking the probabilities of word frequencies in meetings and how these probabilities evolve over time. This has application in predicting topics that one should review in preparation for a future meeting.

In Chapter 8, in order to evaluate our JITIR models on a large-scale dataset, we take the conference proceedings of a large conference spread over 29 years. We design an evolutionary model that tracks the changes of topics over time and adapts itself to the behavior of the data in order to predict which topics appear in a future conference proceedings and which ones disappear. The latter has application in future research planning and funding.

In Chapter 9, we present another JITIR model that aims to predict the topic of one's next search query. This model utilizes various contextual data such as history of one's searches, collaborative patterns evident in all users (e.g. following certain trends), day and time for carrying out its prediction tasks.

Finally, in Chapter 10, we conclude the thesis and present a vision of an extended SILAS for a futuristic multimodal personal assistance. We then present insights into future work.

1.4 Contributions

The novel contributions of this thesis are as follows:

- We present SILAS, the novel memory augmentation system and show that reviewing a past event using SILAS has a significant positive effect in recalling the event.

- We present dDTM, a model that relaxes the assumption that all topics should be present in all time slices of a temporal corpus. This enables modeling intermittent topics which is useful for many user-centered applications such as capturing topic-threads in user conversations over time.

- We present CATS, the first customizable abstractive summarization model for summarizing meeting transcripts. This model achieved state-of-the-art performance on a huge news benchmark dataset

- We present a feasibility study and a model for a futuristic application, predicting the segments of a conversation that one is likely to forget within a week time.

- We present a family of JITIR models that track changes in data over time (e.g. changes of conversation topics) and model recent behaviors distinct from persistent ones. The best performing JITIR model that we introduce learns weights for recent behaviors and persistent ones in an evolutionary process. These JITIR models are designed for predicting topics of a future meeting, predicting topics of research papers in a future conference proceedings as well as predicting the topics of one's next search query.

1.5 Publications Overview

1.5.1 Publications in Thesis

1. **S.A. Bahrainian**, F. Crestani; Cued retrieval of personal memories of social interactions. *In Proceedings of the First Workshop on Lifelogging Tools and Applications at ACM Multimedia*, pages 3-12, 2016.

2. **S.A. Bahrainian**, I. Mele, F. Crestani; Modeling discrete dynamic topics. *In Proceedings of the 32nd ACM Symposium on Applied Computing*, pages 858-865, 2017.

3. **S.A. Bahrainian**, F. Crestani; Predicting the Topics to Review in Preparation of Your Next Meeting. *In Proceedings of the Italian Workshop on Information Retrieval*, 2017.

4. **S.A. Bahrainian**, F. Crestani; Towards the Next Generation of Personal Assistants: Systems that Know When you Forget. *In Proceedings of the ACM International Conference on Theory of Information Retrieval*, pages 169-176, 2017.

5. **S.A. Bahrainian**, F. Crestani; Are conversation logs useful sources for generating memory cues for recalling past memories? *In Proceedings of the Second Workshop on Lifelogging Tools and Applications at ACM Multimedia*, pages 13-20, 2017.

6. **S.A. Bahrainian**, F. Crestani; Tracking Smartphone App Usage for Time-Aware Recommendation. *In Proceedings of the 18th International Conference on Asia-Pacific Digital Libraries (ICADL'17*, pages 161-172, 2017.

7. **S.A. Bahrainian**, I. Mele, F. Crestani; Predicting Topics in Scholarly Papers. *In Proceedings of the European Conference on Information Retrieval (ECIR'18)*, pages 16-28, 2018.

8. **S.A. Bahrainian**, F. Crestani; Augmentation of Human Memory: Anticipating Topics that Continue in the Next Meeting. *In Proceedings of the 2018*

Conference on Human Information Interaction Retrieval (CHIIR'18), pages 150-159, 2018.

9. **S.A. Bahrainian**, F. Zarrinkalam, I. Mele, F. Crestani; Predicting the Topic of Your Next Query for Just-In-Time IR. *In Proceedings of the European Conference on Information Retrieval (ECIR'19)*. pages 261-275, 2019.

10. **S.A. Bahrainian**, C. Eickhoff, F. Crestani; Abstractive Summarization of Meeting Transcripts using Neural Sequence-to-Sequence Models. Submitted, 2019.

11. **S.A. Bahrainian**, G. Zerveas, C. Eickhoff, F. Crestani; CATS: Customized Abstractive Topic-based Summarization. Submitted, 2019.

1.5.2 Additional Publications

1. M. Aliannejadi, **S.A. Bahrainian**, A. Giachanou, F. Crestani; University of Lugano at TREC 2015: Contextual Suggestion and Temporal Summarization.

2. A. Bexheti, E. Niforatos, **S.A. Bahrainian**, M. Langheinrich, F. Crestani; Measuring the effect of cued recall on work meetings. *In Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing Adjunct*, pages 1020-1026, 2016.

3. I. Mele, **S.A. Bahrainian**, F. Crestani; Linking News across Multiple Streams for Timeliness Analysis. *In Proceedings of the 26th ACM International on Conference on Information and Knowledge Management (CIKM '17)*, pages 767-776, 2017.

4. I. Mele, **S.A. Bahrainian**, F. Crestani; Event mining and timeliness analysis from heterogeneous news streams. *Information Processing Management Journal*, vol 56, issue 3, pages 969-993, 2019.

5. E. Rissola, **S.A. Bahrainian**, F. Crestani; Anticipating Depression based on Online Social Media Behaviour. *In Proceedings of the 13th International Conference on Flexible Query Answering Systems (FQAS'19)*, accepted as long paper, 2019.

6. E. Rissola, **S.A. Bahrainian**, F. Crestani; Personality Recognition in Conversations using Capsule Neural Networks. *IEEE/WIC/ACM International*

Conference on Web Intelligence Web Intelligence (WIC'19), accepted as long paper, 2019.

Chapter 2

Related Work

In this chapter we review the most important previous related work by breaking them down into four main themes. The themes are: First, *augmentation of human memory and assisting people recalling and retrieving past events using lifelog data* is a notion closely connected with most of our research in this thesis. Second, *summarization* is the focus of the first part of this thesis. Third, *text representation* methods such as word embedding and topic models are central to both parts of this thesis. We elaborate on these methods in the third section of this chapter. Finally, the fourth important theme that we discuss here is *just-in-time IR* which is the focus of the second part of this thesis.

2.1 Augmentation of Human Memory using Lifelog Data

2.1.1 Augmenting Human Memory in Daily Life

Psychology of human memory has comprehensively studied how human memory recalls events or forgets them. One ground breaking work in this domain was the invention of the *forgetting curve* in 1885 by Ebbinghaus. The forgetting curve (which is an exponentially decreasing curve) shows that a human forgets on average about 77% of the details of what one has learned (for the first time) after six days (see Figure 2.1). This motivated our goal in augmenting human memory to assist one in recalling details of one's past events that one is likely to forget Also, this notion of exponential forgetting was previously used by information retrieval researchers [88] to develop a probabilistic recency-based model, for modeling the strength of memories of people in recalling their visits of places logged on websites such as Tripadvisor or Yelp. We also use an exponential forgetting factor in our JITIR future-state prediction models presented in Chapter

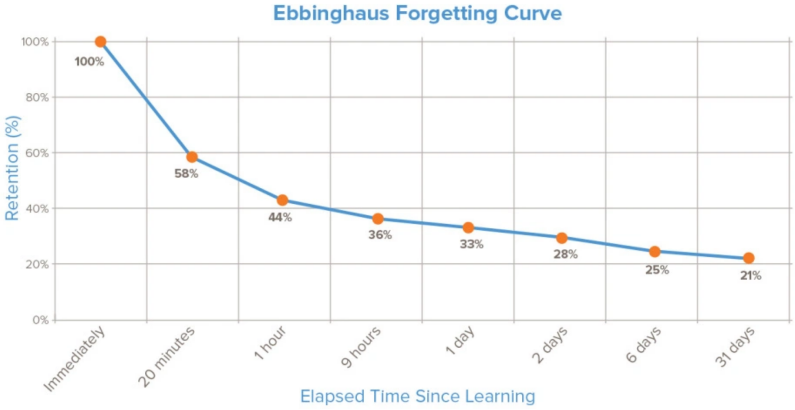


Figure 2.1. The Ebbinghaus forgetting curve

7.

Aiding Human Memory

We start our review by investigating memory enhancement studies in daily life and find out whether memory augmentation technologies can be beneficial for recalling past events. Sellen et al. [121], tested the effect of reviewing past events captured by SenseCam images on the recalling ability of a number of undergraduate student who they presumed had strong memories. SenseCam is a wearable lifelogging camera with fisheye lense from Microsoft. The test subjects were deliberately chosen to be undergraduate students in order to prove that "lifelogging technologies can benefit all of us, not just the elderly or memory-impaired". The findings of this study suggested that, with regard to recollection of past events, automatically captured images may help people remember more than if they had manually taken images on their own. They showed through a case study that images were effective memory cues for recalling memories. Moreover, the study of Lee et al. [85] showed the effect of replaying a person's significant outing recorded by a SenseCam camera, an audio recorder, and a GPS logger to elderly people with normal memories. The study showed that the test subjects experienced improvement in recalling the reviewed events. In a subsequent study [84], the same authors showed that a person with episodic memory impairment, reviewing memories of the person's significant outing could better remember the details of the event after 28 days, as compared with another person with memory impairment who was helped to remember the details of an outing by a caregiver who also attended the outing. Analogously, the study of Hallberg et al. [62] reports a memory support tool to increase autonomy for people with mild dementia. Their tool supports replaying images, video, audio, etc. to it's user in order to facilitate reminiscence of past memories.

All above-mentioned studies support the claim that a person reviewing past activities, regardless of whether they had intact memory, mild dementia, or even memory impairment, would benefit and would experience improvements in recalling memories. However, the problem with the methods presented in these studies is that they mostly present the raw recorded video, audio, or image to their users, and therefore reviewing an audio or video recording of a memory would require users to spend the same amount of time as the original experience to review it. Unlike these studies, in Chapter 3 of this thesis we develop a summarization system, which enables users to review their memories in a shorter amount of time. In order to do so, we extract the most significant topics from transcribed audio recordings to help users better recall past events.

One of the first works on automatic augmentation of human memory was the Remembrance Agent [108] from MIT labs. The Remembrance Agent is a 'program which augments human memory by displaying a list of documents which might be relevant to the user's current context'. This information retrieval system ran continuously with no user intervention and acted as a human memory aid. Every few seconds words around the user's cursor are collected, text queries related to these words are generated and given to a search engine which then retrieves relevant documents from a pool of previously indexed documents. The Forget-me-not system developed at the Rank Xerox Research Center [80] was another system that recorded where the user was, who the user was with, who the user called, and other such autobiographical information and stored and indexed them in a database for later retrieval. Rhodes et al. [107] defined a class of software programs with the goal of JITIR. These programs were 'software agents that proactively present potentially valuable information based on a person's local context in an easily accessible yet non-intrusive manner'. One of the versions of these software agents visualized content using a wearable head-up display. Such systems could empower humans to live more independent and healthy lives by assisting them in recalling important events which could otherwise complicate their lives and cause them embarrassment and problems. In this thesis, we present a number of JITIR models for augmentation of human memory.

Emotion, Biometrics and Memory

The work of Grossmann et al. [139] reveals that an emotion one experiences during an event does have an impact on recalling that memory later on. Furthermore, the level of attention during an experience is another important aspect that influences how much one can later recall that experience. To measure attention or emotional arousal researchers have used biophysical sensors such as Electro-Dermal Activity (EDA). Braithwaite et al. [35] states that 'the coupling between cognitive states, arousal, emotion and attention enables EDA to be used as an objective index of emotional states'. Additionally, the results of the research conducted by Yoshida et al. [151] confirm that EDA can be used as an attention index. Affective physiological signals such as EDA have been used by information retrieval researchers for understanding users' information needs as a form of implicit relevance feedback [11, 48, 95]. Moshfeghi et al. [95] reports that affective signals were an effective complementary source of information for relevance judgment prediction across a number of search intentions. Edwards et al. [48] also uses EDA to assess users search behaviors. In Chapter 6 of this thesis we use both EDA and sentiment as measures for predicting parts of a conversation that one may forget/remember.

2.1.2 Augmenting Human Memory in Meetings

Jaimes et al. [71], proposed a video summarization tool that summaries and indexes a meeting's video recording. The system then allows users to use various types of queries and memory cues to find certain content in the meeting. Chen et al. [38] also presents a memory augmentation tool and cites 'being reminded of information in a work situation (e.g. previous meetings with an individual' as an application of their system. A number of previous research proposed the use of summarization tools as opposed to indexing, by arguing that in many cases a user may not remember enough about a past meeting to even be able to formulate a query [13]. Thus, in such settings a summary of the meeting can reminisce the faint memories of that meeting.

Jaimes et al. [71] ran a user study with 15 participants to research the common issues that people forget regarding a past meeting. They asked the participants using a questionnaire a week after a meeting and assessed the accuracy of answers. Their results show that on a defined scale, about 35% of the questions they asked about the content of a meeting were answered incorrectly. We note that these were only 35% of the questions and not the actual meeting content. Motivated by the results of this work, in Chapter 6 we aim at predicting the parts of a meeting that one forgets.

Tucker et al. [138] present and compare various speech compression techniques including audio speed-up methods and excision methods. Speed-up methods are simply based on increasing the speed of audio playback to achieve a certain temporal compression rate, while excision methods are removal of insignificant utterances or insignificant words from the transcripts of speech signals. Their study shows that excision methods outperformed speed-up methods in representing an audio recording of a conversation. Based on these results, we believe that since topic models are in principle excision methods for extracting salient themes of texts, they can serve as viable solutions to summarize and present textual content. Therefore, in Chapter 3, we develop a system named SILAS for automatically summarizing social interactions log data, using topics and images.

SILAS is capable of summarizing conversations of a person for later reviewing of those conversations. We show through case studies, that reviewing a past week using SILAS has significant effect on recalling events and enhancing one's episodic memory.

In summary, development of memory augmentation models with a focus on workplace meetings is presented in Chapters 3, 6, and 7.

2.1.3 Challenges of Lifelog-Related Research

Evaluation Challenges

Since evaluation of lifelog analysis systems is a major challenge for most studies in this field, including our work, we would like to explain some of the challenges as reflected in previous related work. Gurrin et al. [60] state that "there are currently no defined holistic benchmarking evaluation tests in lifelogging". Since lifelog data presents a single person's personal life, and also due to the fact that wearable devices are still in their infancy, there are various issues that have caused lifelogging evaluation very difficult. The first issue is that our daily lives endure so many different aspects (i.e., so many different modalities of data), that there are many different views on what kind of information is important to be collected and analyzed. This has resulted in limited availability of lifelogging datasets. The second issue is that most of the available datasets that reflect an average person's daily life contain merely images, captured at regular time intervals, or videos throughout a short period of time such as the AIHS dataset [72] or the multimodal NTCIR 2016 dataset [61], thus limiting progress of other media types that could potentially be suitable candidates for capturing a person's life. The third issue as reflected in the work of Gurrin et al. [60] is that "a lifelog is most meaningful to only one person, the data gatherer. While anyone could examine the lifelog dataset and generate queries and relevance judgments, the actual usefulness of lifelog retrieval system is not simply to generate a good quality ranked list; it is to support the individual in accessing past memories using carefully considered reasons". They further state that "the only truly reliable judge of the accuracy of many potential retrieval approaches is the individual who gathered the data in the first place". Given that gathering such datasets collected by even a small number of participants is very expensive, there are still no standard benchmarks for lifelogging. In this thesis, we collected our own meeting/conversation datasets in an attempt to focus on other untouched aspects of lifelogging with a focus on human memory enhancement.

Privacy Challenges

Serious privacy concerns are raised regarding the use of wearable cameras and other recording tools for life-logging, that allow seamless capture of an experience in a public or private setting. Photos taken in a public setting can reveal sensitive information about bystanders, and photos taken in private settings might contain sensitive information of the individual lifelog user, such as photos of smartphone screen or computer screen. In order to address such privacy concerns, various systems were designed that can control access to a lifelog dataset, enable secure sharing with other parties as well as maintaining privacy of strangers while recording lifelog images at the first place. In the following, we briefly introduce some of these works:

Bexheti and Langheinrich [30] discuss designing an access control mechanism that prevents unauthorized access to captured media and memory cues in a memory augmentation system and the challenges that arise when sharing the memory cues with other co-located people. They suggest that this control mechanism should provide continuous evaluation of access policies and monitoring the data usage with the flexibility to grant and revoke customized access among the sharing parties. In a later work, Bexheti et al. [32] present a device named MemStone, that is a tangible user interface for controlling data capture and sharing activities of a memory augmentation system using five physical gestures. Saran [115] developed an Android photography app that can blur faces of strangers and bystanders in photos taken in public settings, preserving their privacy. In our experience, recording audio is perceived more intrusive as compared with images or video. As such, we believe that there are possibilities for future research on privacy-preserving audio recording. Audio can capture rich semantics and communications between different parties and therefore is of strong importance. The development of such privacy-preserving systems allows for enhanced control over what data is stored and who has access to it. Technologies that address the above privacy concerns are required for further advancement of lifelogging research.

2.2 Summarization

As previously stated, one of the main themes of this thesis is summarization including topic summaries extracted in the form of a group of words that represent a high-level semantic concept as well as generated texts presenting a condensed version of a source document. In the following sections we first discuss related work of the former approach followed by the latter.

2.2.1 Content Extraction From Text and Retrieval

There has been little previous work on generating narratives describing daily life. Most previous narratives were generated manually. For example, the MyLifeBits project [54] was using simple manual narratives of time and location to describe data. Riedl et al. [110] define narrative as "a sequence of events that describes how the story world change over time". Textual narratives allow lifelog data to be easily searched by keywords. Therefore, developing methods which would automatically generate textual keywords to be utilized for searching a social interactions dataset is of strong importance.

Orad et al. [99] demo the implementation of an interesting tool which enables users to search the conversations recorded on the Apollo mission. They used several large collections of data which were available online and attempted to automate temporal, spatial, and topical content alignment. Since most of the communications were precisely time stamped, automated alignment of data across multiple sources of data was easily feasible. The difference between their work and ours presented in Chapter 3, is that our data contains also images and summarizes lifelog data. Furthermore, our system creates topic models and tracks the evolution of topics over time, links topics to people, and to locations.

Different meeting search, browsing, and summarization systems have been developed for aiding people to reuse content of meeting at a later time [71, 137]. As an example, [71] proposed a video summarization tool that summarizes and indexes a meeting's video recording. The system then allows users to use various types of queries and memory cues to find certain content in the meeting. Tucker et al. [137] investigated several meeting browsing systems. They state that most meeting browsers have ignored semantic techniques such as topic tracking.

This is while other research showed that extractive summaries [69, 96] and in particular Latent Dirichlet Allocation (LDA) [34] topics extracted from meetings [146] provide users with an efficient and effective way of browsing meeting content. We introduce LDA in Section 2.3.1. Thus far, focusing on a user's informational needs when going through previous recorded meetings has been mostly limited to taking a user's search keywords as feedback or highlighting the decisions made during a meeting. However, one common informational need of a user can be to use the previous meetings for preparing a future meeting. Therefore, there is a need for tracking topics and semantic information to produce summaries which are important for the near future of a user. Addressing such need is the focus of Chapter 7.

The rationale behind using topics is that they can effectively summarize long conversations and at the same time backtrack from a topic to its sentences of origin. This gives the user the opportunity to review the exact sentences from which a topic was extracted.

Moreover, studies [69, 96] show that extractive summaries provide a more efficient way of navigating meeting content than simply reading transcript and using audio-video record, or navigating via keyword search. Similarly, minute taking in meetings although useful, requires more work from the user side, both for writing and later reading it. Additionally, in certain situations (e.g. less formal meetings) such facilities may not be available. Furthermore, if a user needs to know about a topic in more detail, SILAS uses LDA to trace back the words of a topic, to the spoken sentences where the topic was extracted from. However, the drawback of most memory augmentation tools is that with the growing number of meetings that one participates in and with the growing number of topics discussed, there are thousands of topics to be reviewed every week in preparation for future meetings that might touch only a part of the many topics discussed in previous meetings.

2.2.2 Generating Summaries in the form of Natural Language

Automatic document summarization is defined as producing a shorter, yet semantically highly related, version of a source document. Solutions to this task are typically classified into two categories: extractive summarization and abstractive summarization.

Extractive summarization refers to methods that select sentences of a source text based on a scoring scheme, and eventually combine those exact sentences in order to produce a summary. Conversely, abstractive summarization aims at producing shortened versions of a source document by *generating* sentences that do not necessarily appear in the original text. Recent advances in neural sequence-to-sequence modeling have sparked interest in abstractive summarization due to its flexibility and broad range of applications.

Abstractive Summarization is often treated as a sequence-to-sequence problem, meaning that a sequence of words in a source document is mapped to a

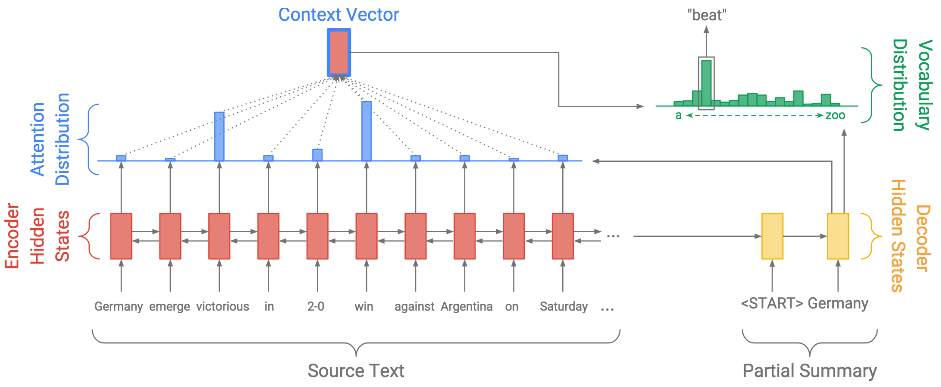


Figure 2.2. The sequence-to-sequence baseline model [120]

shorter sequence of words constituting a summary. This section presents the related work associated with our research on customizable abstractive summarization presented in Chapter 5.

Prior to the rise of neural sequence-to-sequence models, there had been relatively little work in the area of abstractive summarization. TOPIARY was an abstractive model proposed in 2004 by Zajic et al. [152] which showed superior results in the DUC-2004 summarization task. This model used a combination of linguistically motivated compression techniques and an unsupervised topic detection algorithm that inserts keywords extracted from the article into the compressed output. Some other notable work in the task of abstractive summarization includes using traditional phrase table-based machine translation approaches [26] and compression using weighted tree transformation rules [41].

Recent work approaches abstractive summarization as a sequence-to-sequence problem. One of the early deep learning architectures that was shown to be effective in the task of abstractive summarization was the attention-based encoder-decoder proposed by Bahdanau et al. [12]. This model had originally been designed for machine translation problems, where it defined the state of the art. Due to strong similarities between machine translation and abstractive summarization (i.e., both tasks aim at mapping an input sequence to a semantically related output sequence), this architecture showed similarly strong performance in the summarization domain [97]. Owing to the influential role this model has been playing for a number of sequence-to-sequence problems, we include it as a baseline in our experiments in Chapter 5. Figure 2.2 depicts a schematic illustration of this model.

Attention mechanisms were shown to enhance the basic encoder-decoder model [12]. The main bottleneck of the basic encoder-decoder architecture was its fixed-sized representation (“thought vector”), which was unable to capture all the relevant information of the input sequence as the model or input scaled up. However, the attention mechanism relies on the notion that at each generation step, only a part of the input is relevant. This mechanism helps the model decide, at each generation step, which part of the input encoding to focus on in order to generate output words.

In the following, we review some of the most important neural architectures applied to abstractive summarization, which were inspired by the attention encoder-decoder architecture described above and often bring improvements to the basic architecture.

The Pointer Generator Network (PGN) was an architecture originally proposed in the machine translation domain [141] and subsequently applied by See et al. [120] to the task of abstractive summarization. This model aims at solving the challenge of out-of-vocabulary (OOV) words and factual errors. The main idea behind this model is to choose between either generating a word from the fixed vocabulary or copying one from the source document at each step of the generation process. It incorporates the power of extractive methods by “pointing” [141]. At each step, a generation probability is computed, which is used as a switch to choose words from the target vocabulary or the source document. Under this model, OOV words can be directly copied from the source text. This model is the current state of the art on abstractive summarization. For this reason, and due to the strong similarity between our proposed model presented in Chapter 5 and PGN [120], we include it as one of the performance baselines in our experiments. Our summarization model differs from the PGN firstly in the use of a different attention mechanism which forces the model to focus on certain topics when generating an output summary. Secondly, our model enables the selective inclusion or exclusion of certain topics in a generated summary, which can have several potential applications. This is done by incorporating information from an unsupervised topic model.

The coverage mechanism [135] is another improvement over the attention-based encoder-decoder architecture which keeps track of the global level of attention given to each word at all steps in time. In other words, by summing up the attention at all steps, the model keeps track of how much coverage each encoding has received. Subsequently, this is given as input to the attention model so that it penalizes attending to parts of the input that have already been sufficiently covered. We include a PGN combined with coverage [120] as a further baseline in our experiments, and also add this mechanism to our proposed model.

Hierarchical attention is yet another component effectively used in text summarization [39]. The main idea behind hierarchical attention models is to extract more structural information from a source document by processing it both at the sentence level as well as the word level, usually by two different Recurrent Neural Networks (RNNs). They are motivated by the intuition that documents are compositions of words, sentences, paragraphs and even larger units of text [39]. Other analogous research [97] in this domain also utilizes POS tags, named entities and *TF-IDF* information, apart from *word2vec* [94] and *Glove* [103] word embeddings. Word embeddings are vectors of numbers which represent a word in its context, such that each number presents a different similarity dimension.

The drawback of such models based on hierarchical attention is limited abstractive capability as a lot of content in the summary is directly extracted from the source document. Due to the use of an extractive framework, the work presented in [39] is not directly comparable with ours. However, hierarchical attention is a promising advancement that may be effectively used in an abstractive summarization model.

Intra-Attention [101] is another approach which utilizes the attention mechanism in a different way. The basic attention encoder-decoder approaches attend merely to the encoder output states. Instead, the intuition behind Intra-Attention is that a word being currently generated also depends on previously generated words. Therefore, Intra-Attention was introduced to apply attention to the decoder outputs. Analogously to the coverage mechanism described above, the Intra-Attention mechanism also aims at preventing repetition of words and sentences in the output summary.

The work of Paulus et al. [101] acknowledges the various ways in which a document can be correctly summarized, with the reference summary being just one of those possible ways. Therefore, there should be some room for variation in the summary. Based on the notion of reinforcement learning, their model produces multiple summaries of a source document and by defining a loss function it determines how good the produced summaries are. Similar to [120] they also use the pointer generator mechanism to switch between generating a token or extracting it from the source.

Existing neural models do not directly take advantage of the latent topic structure underlying input texts. To the best of our knowledge, our model presented in Chapter 5 is the first work including this source of information explicitly into the neural abstractive summarization model. The experimental section of Chapter 5 will demonstrate the merit of this approach empirically.

2.3 Text Representation

2.3.1 Representations of Text Documents: Topics and Beyond

As explained previously, in the literature [145], topic models are defined as hierarchical Bayesian models of discrete data, where each topic is a set of words that together represent a high-level semantic concept. According to this definition LDA [34] was introduced. In both parts of this thesis we use LDA. In the following we explain LDA in detail:

Latent Dirichlet Allocation

LDA is a generative probabilistic topic model which discovers topics present in a given text corpus. LDA represents each topic as a probability distribution over words in the documents. The generative process of LDA is as follows:

1. For each topic $\beta_i, for\ i \in \{1, \ldots, K\}$:
2. For each document d:
   - (a) Draw topic proportions $\theta_d \sim \ Dir(\alpha)$.
   - (b) For each word:
Draw $z_{d,n} \sim \mathrm{Mult}(\theta_d)$,
Draw $w_{d,n} \sim \mathrm{Mult}(\beta_{zd,n})$.

The graphical model of LDA is shown in Figure 2.3.

LDA is the basis for a number of other topic models including ours presented in Chapter 4. LDA is a Bayesian network that generates each document from a corpus using a mixture of topics. For each document, a multinomial distribution θ over topics is randomly sampled from a Dirichlet function with parameter α (which influences the shape of the distribution). Moreover, to generate each word, a topic z is chosen from this topic distribution and a word, w, is generated by randomly sampling from a per-topic multinomial distribution β.

In Chapter 3, similarly to [146], we use LDA topics as summaries. That is because the use of LDA topics as representations of documents is theoretically motivated and endorsed by previous related work on temporal compression of recordings of conversations [138].

In Chapters 7, 8, and 9 we use LDA topics to predict their continuation in future time slices in a JITIR setting. However, our proposed methods in these chapters are generic enough that any vector representation of documents could

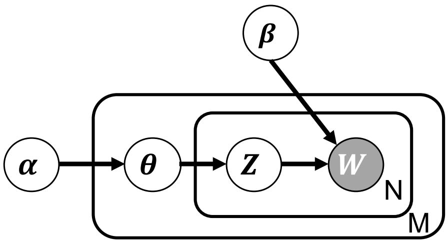


Figure 2.3. Graphical model of LDA.

be used and predicted by the methods. This advantage of our models is very important because there are many human-centered studies such as [136] which produce various textual summaries of conversations other than LDA topics. The only requirement for using our models is to have a vector containing a set of words where each word is associated with a corresponding probability score showing the strength of its presence in a given document.

Other Word Representation Methods

Recently, there has been interesting work on mapping semantically related words to nearby positions in the vector space in an unsupervised way. Some example approaches are the well known word2vec model [94], Glove [103], in addition to other probabilistic word embedding methods such as [140] which uses a Gaussian distribution for modeling each word. In this chapter, we build on the same concept by using a Gaussian Mixture Model (GMM) for modeling each word in each of its contexts. We define a context of a target word as a word co-occurrence in the same vicinity. For example, the word 'book' can mean making a reservation or it can also mean a bound collection of pages depending on the context.

Temporal Word Representation Methods

Temporal topic models are capable of tracking the evolution of topics over time and model probability of words over time. The Dynamic Topic Model (DTM) [33] is the state of the art in this domain. It is based on the LDA model and requires as input a sequential corpus of documents. It uses a linear Kalman filter [73] to compute the evolution of each topic over time. The authors showed that, on a sequential dataset, DTM outperformed LDA in terms of log likelihood. We elaborate on the details of this model in the background of Chapter 4.

We use DTM as a baseline model in Chapters 4 and 8. In Chapter 4, we compare this model against our temporal topic model capable of tracking intermittent topics over time. In Chapter 8, we compare DTM with our proposed model capable of predicting topics that continue in a future time-slice.

We note that a temporal topic model such as the DTM is not a suitable option to be included in our benchmark in Chapter 7 for predicting continuing conversation topics, because: (1) it assumes that all topics are present over all time slices of a given dataset, which does not hold in the case of conversation logs (2) its not capable of tracking textual representations of conversations other than topics.

Another notable temporal approach is the continuous-time dynamic topic modeling [145]. The model relaxes the assumption made by DTM, which is, all documents are exchangeable in each time slice. For this purpose, it replaces the state space model used by DTM in order to model Brownian motion [82]. The model is able to capture continuous topics by taking into account the timestamps of documents within a collection with different levels of granularity. Unlike this work, in Chapter 4, we use a Markovian state space model [55] for tracking intermittent topics over time. This means that the evolution of a topic will be a discrete process. That is because a Markovian state space is based on the Markov assumption which states that data in each time step merely depends on its previous time step.

Additionally, another topic model that tracks the evolution of topics is the Topics Over Time model [147]. This model uses the timestamps of documents as observations for the latent topics, and each topic is associated with a continuous distribution over timestamps. Thus, for each generated document, the mixture distribution over topics is influenced by both the word co-occurrences and the document's timestamp. Although this model allows to account time jointly with word co-occurrence patterns, it does not discretize time and does not make Markov assumptions over state transitions in time. However, the topics are constant and the timestamps can be used to explore them.

Evaluation of Topic Models

Blei et al. [33] and many other previous work in the domain of topic modeling, use likelihood on held-out data as a standard measure of evaluation. Similarly to the common practice, in Chapter 4 we also use likelihood to evaluate and compare our model against DTM. Additionally, we present various qualitative results showing the effectiveness of our model in identifying patterns and emerging trends in the data.

2.4 Just-In-Time Retrieval of Information

Addressing the users' near-future information needs has been studied in the context of personal assistants [15] such as Google Now, Microsoft Cortana, or Apple's Siri. These systems offer proactive experiences and aim to recommend the right information at just the right time [123, 127]. As an example, Sun et al. [127] proposed an approach for tracking the context and intent of a user leveraging smartphone data [14] in order to discover complex co-occurring and sequential correlations between different signals. Then, they utilized the discovered patterns to predict the users' intents and to address their information needs.

In the context of proactive search and recommendation, Song and Guo [125] aimed at predicting task repetition for offering a proactive search experience. They focused on predicting when and what type of tasks will be repeated by the users in the future. Their model was based on time series and classification. They tested the effectiveness of their approach for future query and future app predictions. Our work differs from their work since we take a collaborative time-series approach for predicting the topics of future user queries. Moreover, our goal is to predict the topic of one's next query and not only predicting the repetition of a search task.

Agichtein et al. [2] tried to predict the continuation of a previously started task within the next few days. Similarly to [125], they defined the prediction of the continuation of a task as a classification problem. They used an extensive set of features for training the classifiers. Such features include query topics, level of user engagement and focus, user profile features such as total number of unique topics in prior history, and repeating behavior among others. Our work differs from this work as we do not simply try to predict the search task continuation in the future but we also aim at predicting the day of the week and the approximate time of the day when a query topic will occur. Moreover, unlike their model which is a classifier based on a number of hand-engineered features, our model has a time-series structure and it evolves over time by learning from the data and correcting itself over time.

Furthermore, another interesting but different work consists in the identification of recurrent event queries and was presented by Zhang et al. [154]. In this work, the authors aimed at identifying search queries that occur at regular and predictable time intervals. To accomplish this, they train various classifiers such as Support Vector Machines and Naïve Bayes, on a number of proposed features such as query frequency, click information, and auto correlation. They conclude that a combination of all features leads to the highest performance.

Part I

Content Extraction, Summarization and Representation for Personal Assistance

Chapter 3

A Social Interactions Log Analysis System

3.1 Introduction

In recent years, the introduction of various wearable data capture devices has created new opportunities to utilize the collected data for various human-aid applications. Among all the interesting lifelogging applications are health-care applications such as reviewing memories for human memory enhancement and support of failing memories by being able to search these archives. In addition to memory enhancement and treating memory deficiencies, a person reviewing summaries of one's social interactions could be experience increased self-awareness, being able to plan the future better, etc.

In this chapter, we present a novel Social Interactions Log Analysis System (SILAS) which summarizes one's daily social interactions, and connects the conversations with similar topics over time for a later presentation to a user. Such system can assist the user by augmenting the user's memory, by allowing the user to review summaries of past events. Such system can benefit people by helping them reminisce faint memories of past events or to ameliorate their episodic memories. Despite whether these people are young or old, their memories are intact or not, they are living independently or are assisted by nurses at a nursing home, studies have shown that replaying their lives to them have significant effect in helping them better recall past events of their personal lives [76, 85, 121]. Therefore, developing technologies that help people to live a more independent and autonomous life with respect to their memory strength is of crucial importance.

SILAS takes as input transcriptions of audio recordings of one's conversations,

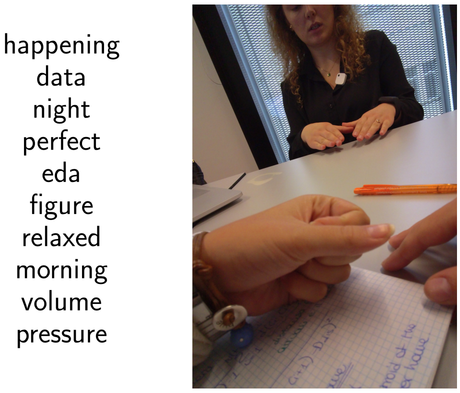


Figure 3.1. Illustration of a randomly chosen event snippet produced by our memory augmentation system.

images taken automatically by one's wearable camera at fixed time intervals of 30 seconds, and continuous recordings of one's GPS coordinates. All three media types are synchronized by their temporal state (i.e., point in time). SILAS then processes the data by modeling the topics of the transcribed conversations, detecting and recognizing faces in the images, and connecting the topics with their corresponding images, and locations. We refer to a topic connected with its corresponding image, location and time as an *event snippet*. Figure 3.1 illustrates a randomly chosen event snippet consisting of an image and a topic from our dataset.

Our work is firstly motivated by studies of Berry et al. [29], Pauly-Takacs et al. [102], and Silva et al. [105] which illustrate that after replaying of a given day's activities to a number of participants there were measurable effects of memory improvement. However, since replaying a raw video of an event to a person would require the same amount of time as the original experience, we aimed at developing a system that could summarize a person's day/week/month/year. Secondly, we aim at developing a system that enables search through daily social interactions data by querying people who are identified in images, topics which are extracted from audio signals, and locations where conversations about certain topics took place.

The most important contributions of this chapter are:

- To the best of our knowledge, this is the first work which develops a summarization system of one's daily social interactions that combines topics of conversations with their corresponding images.

- We develop a system for indexing and searching personal memories, that

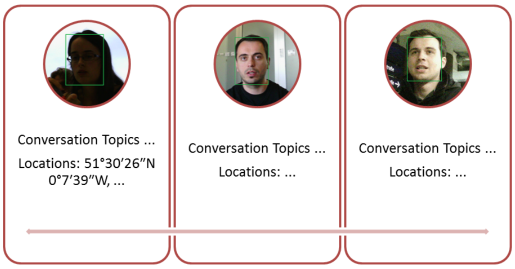


Figure 3.2. Illustration of the Features of Our Novel System

can connect people with topics and vice versa.

- We build a domain-independent summarization system which can handle different types of conversations (i.e., with respect to topics and lengths).
- We thoroughly evaluate our system, both in terms of its accuracy in computing event snippets as well as enhancing human memory.

3.2 Definitions

Episodic Memory is defined by Schacter et al. as "the collection of past personal experiences that occurred in a special time and place" [116]. In this chapter, we show that reviewing personal memories of daily social interactions using our memory augmentation tool enables better recall of past conversations.

Memory Cue is a stimuli that triggers a memory in human mind. For example, if several central words of a conversation between you and a colleague would be shown to you, the memory of that conversation might be triggered in your mind. Schacter et al. defines memory cue as "external information that is associated with stored information and helps bring it to mind" [116]. In this study, we utilize the notion of memory cue in our memory augmentation system to help trigger personal memories of the past in one's mind. Those personal memories that their occurrence is associated with a specific time and place can be classified as episodic memories.

3.3 Social Interactions Log Analysis System

We start introducing our system by first presenting a general picture of it. Figure 3.2 illustrates the features of SILAS. Over a time-line one meets many people, including family members, friends, acquaintances, or even strangers. The face detection and recognition module detects faces in images taken by a small wearable camera and then recognizes faces of people known to the system. We assume that the names and a number of images of each contact person are given to the system. In the real-world this is a valid assumption, because as the user meets more people, the system also detects new faces and may ask the user their names, and tag the faces with the corresponding names. Furthermore, the transcribed audio recording of conversations with other people are analyzed and topics are extracted. Topics or in other words "concepts" are probability distributions over words of documents with a fixed vocabulary. Additionally from the extracted topics, the most similar ones are connected to one another over time and as a result chains of evolving topics are derived. At the same time the physical location(s) where a conversation takes place is/are recorded, meaning that not only the conversations that are stationary in a single location, but also those that the people involved are moving, will be processed by the system. Finally, since all data types are time-stamped, faces will be connected with conversations and locations with whom and where they took place.

SILAS indexes such dataset based on the known people, topics, and locations. This enables manual search of a social interactions log archive as well as automatic summarization of the archive. Although, SILAS currently processes the three mentioned data types, its architecture is designed in a way to support other sensor data (e.g. accelerometer, heart rate, etc.) as well. In the next section we describe all the components of the system in detail.

3.4 System Components

As shown in Figure 3.2, our system currently handles three types of data, namely, conversation transcripts, images, and GPS coordinates. In the following we describe each module of our system:

3.4.1 Text Processing

The audio signals are converted to text in order to be processed. We transcribe the speech signals manually using an online transcription service, thus we merely deal with text. In the future, an ASR (Automatic Speech Recognition) component can automatically transcribe recorded conversations/meetings. In

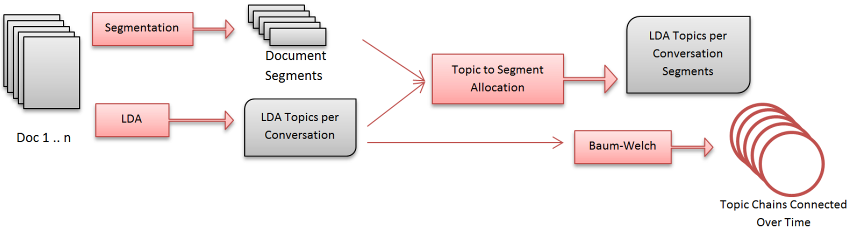

Segmentation
Document Segments
Doc 1 .. n
LDA
LDA Topics per Conversation
Topic to Segment Allocation
LDA Topics per Conversation Segments
Baum-Welch
Topic Chains Connected Over Time

Figure 3.3. Components of Our System Handling Transcribed Conversations

the meanwhile, since an ASR system would inevitably add noise and errors to the system, in this study we have assumed perfect transcription. This will help us only evaluate SILAS and not incorporate noise from an ASR system in our evaluation. The text processing module is the most essential module of our system. The data that is processed by this system contains transcriptions of audio recordings of one's daily conversations in English language. Figure 3.3 demonstrates all the components of our text processing module. We explain the figure by discussing each of illustrated components along with our intuitions behind our model.

In the figure, we refer to each conversation as a document. Thus, the left-most item in the figure shows all the documents. We note that since SILAS can summarize one's daily social interactions at different levels of granularity, i.e., day, week, or month, etc., the definition of documents and time slices will change accordingly. Documents could be of varying length. Each turn in the conversation ends with a paragraph break in the documents. Since some turns in a given conversation can be very long or short in size, we break down the conversation to smaller segments. For this purpose we use the texttiling algorithm [65] which has shown a good performance in other previous work such as in [46] and is suitable for our purpose.

Segmentation of a Transcript using Texttiling: Texttiling [65] is "a technique for subdividing texts into multi-paragraph units that represent passages or subtopics". It utilizes patterns of lexical co-occurrence and distribution as discourse cues for identifying major subtopic shifts. We note that the texttiling algorithm segments documents at end of paragraph breaking points. Therefore, one segment would at least contain one paragraph, which as mentioned above is one turn in a conversation. Our purpose behind using texttiling is to connect an image (based on its time stamp) to a topic which has the highest probability in a texttiling segment of a conversation which virtually corresponds to the same time stamp. That is, since images are captured every 30 seconds, we could find their corresponding text segments by computing an estimated time per each segment. By using our knowledge about the length of each conversation in terms of seconds and also the number of words spoken in total as well as the number of words spoken in each segment computed by the texttiling algorithm, we can compute the length of each segment in seconds. Furthermore, we identify the image closest to the mean of each segment in terms of seconds. Then we find the LDA [34] topics which each corresponds to a segment in the text, and as explained above each segment is also connected to an image. Thus, in this way event snippets each consisting of an image, a topic, a location and time are computed. We explain this process of keywords extraction in the following paragraphs.

Keyword Extraction: In order to compute topics (i.e., keywords extracted from conversation/meeting transcripts) we adopt two strategies for extracting topics from conversations. The first method is designed for extracting the main themes of a few or one conversation transcript(s) in the temporal order they were discussed. The second method is designed for extracting the main themes of a longer conversation log spread over a week, month or year. We explain the two methods as follows:

- The first method is designed for extracting the main themes of one conversation transcript, computes top co-occurring words from each segment according to the Point-wise Mutual Information (PMI) method [40]. For this purpose by analyzing the words in all sentences of a segment, we compute probability scores for each word, showing its frequency of occurrence in that particular segment. Then, we select the top ten words based on the computed probability scores.

- In the second method, at the same time that a conversation transcript is segmented using the texttiling algorithm, LDA is applied to it and its topics are extracted. LDA is a generative model, currently commonly perceived as state-of-the-art in topic modeling. It discovers topics present in a given text corpus, each topic consisting of a probability distribution over words in the documents. Additionally, we use variational inference as our inference method. The outcome is a number of extracted topics from each conversation. Although, due to the nature of our experiments (explained in the evaluation section of this chapter) which mostly focus on weekly summarizations, we run LDA on each conversation, as mentioned earlier in this subsection, it is viable to run LDA on all conversations of a week, month, or even year for achieving different levels of granularity.

Event Snippet Generation: finally, we connect each set of ten extracted topic words to an image that corresponds to those topic words based on point in time. As shown in Figure 3.3, after having topics and segments extracted from a conversation we assign a topic to each segment. This is done by using LDA to return per each texttiling segment, the topic which has the highest probability in that segment. The result is that the topic with the highest probability in a segment of a conversation is assigned to that segment, and thus could also be linked with an image.

By using the above-mentioned model, all topics extracted from each segment in each conversation are not only chained based on topic similarity but also assigned a sequence label. Moreover, using the sequence-labeled topics, an index file is constructed which maps each topic chain to the corresponding segments in all conversations, thus keeping track of the sequences of spoken topics. We explain the topic chaining in Chapter 4 where we explain the dDTM model in detail.

3.4.2 Motivation Behind the Two Keyword Extraction Methods

Previous studies [146] have shown that LDA topics [34] are effective representations of a conversation. LDA extracts topics based on co-occurrence of words. Therefore we used LDA to extract themes of longer conversation logs.

On the other hand, our other model for extracting themes of one conversation transcript also relies on the idea of capturing top word co-occurrences with the difference that we also consider time (i.e., sequence of events in a conversation) as a factor. This is while LDA does not consider the order of words in a document. Additionally, the study of Tucker et al. [138] showed that excision methods (i.e., methods that extract important words from a conversation) outperform audio speedup methods in terms of human understanding of a conversation. Therefore, we developed this model for extracting the main themes of one conversation transcript by relying on the notion of word co-occurrence as well as excision. For comparing this method versus LDA, we conducted a user study involving ten participants. We presented the participants with the transcription of a lecture as well as a number of keyword sets extracted both based on LDA (with seven topics) and our method (where the number is determined automatically). After the participants read the transcription we asked them to judge which set of words were better representations of the transcription in terms of human understanding. The results of this experiment showed that nine out of ten participants found the output of our method more understandable whereas only one participant found the output of LDA to be a better interpretation of the conversation.

Therefore, the efficacy of the model for extracting the main themes of one conversation transcript was endorsed both by previous work as discussed above (i.e., showing the effectiveness of capturing co-occurrence patterns of words) and one preliminary user study.

3.4.3 Image Processing

The lifelog images used for generating event snippets are captured using a Narrative Clip camera. This device clips on a shirt and captures images at fixed time intervals of 30 seconds. Its battery usually lasts one full day which makes it useful for our experiment. All the images taken are precisely time-stamped which could be later aligned with conversations and location information. The images taken by this device were then processed as explained in the following. The image processing module performs two main tasks. First, it has a face detection algorithm which detects faces in images. Second, assuming that a list of names and a number of images of contacts of the user are given to the system, it utilizes a face recognition algorithm to recognize the faces of people who appear in the images. In this study, we mostly focus on the processing of text rather than other data types, and thus used two already existing techniques for face detection and recognition. However, in future work we would attempt enhancing SILAS by building stronger classifiers for our purpose. In the following, we explain the two modules.

For detecting faces in images we use the HaarCascade classifier [142] available in the opencv library. Furthermore, we perform histogram equalization on grayscale version of the images in order to normalize them. Due to the inevitable occasional rotations of images captured by the Narrative Clip, we run the HaarCascade classifier on all the images multiple times while rotating the images with angles of ‘0’, ‘5’, ‘10’, ‘-5’, ‘-10’. The result is detected faces of the entire dataset

Additionally, using the detected faces we train a high dimensional LBPH (Local Binary Pattern Histograms) classifier [1] for recognizing faces. For this purpose, we labeled the images of people known to the user with their names and used ten images of each user to train the classifier. The outcome is images containing recognized people. The reason behind choosing LBPH was that, although the state-of-the-art in this area is currently Facebook’s algorithm [129], the LBPH is still a very good classifier appearing on most benchmarks including that of Facebook’s paper. We note that since the LBPH classifier computes a confidence score for classification of each image containing a face, whenever, this confidence score is lower than a certain threshold for an image, there is a strong possibility that the face appearing in the image is new and unknown to the system. In such situations the system will ask the user for the name of the person appearing in the image. Over time, the classifier is re-trained until it is given enough labeled data for that particular face to recognize it correctly.

3.4.4 Location Processing

The last module of our system, records the current location (i.e., GPS coordinates) using an available smartphone application. The Narrative Clip itself records location coordinates for each captured image. However, since the GPS coordinates that it records are coded and not usable for us we used a standard smartphone application. The location information is recorded after every repositioning of ten meters. Each pair of recorded latitude and longitude is precisely time-stamped. Thus, this information could be aligned with conversations and images.

3.4.5 Integrated System

As described in the previous subsections, SILAS is capable of summarizing and indexing social interactions data. It can categorize the spoken topics by the locations where they were spoken, or categorizing topics by the people whose faces were recognized in the images when those topics were spoken, etc.

Also since it computes the evolution of topics over time, it can show which people talked about the same topic thread over time. In the future, similarity patterns between various locations could be also utilized [6] to enable sophisticated search queries in a lifelog.

In the next section, we outline an evaluation framework which empirically shows the effectiveness of our system in connecting people to the corresponding topics that were spoken when they were present. Also, we show the effectiveness of our system on enhancing human memory.

3.5 Evaluation

In this section, we present a description of two datasets along with evaluations of SILAS in augmentation of human memory. In Section 3.5.1 we describe our datasets, and then we present an empirical evaluation in the subsequent sections.

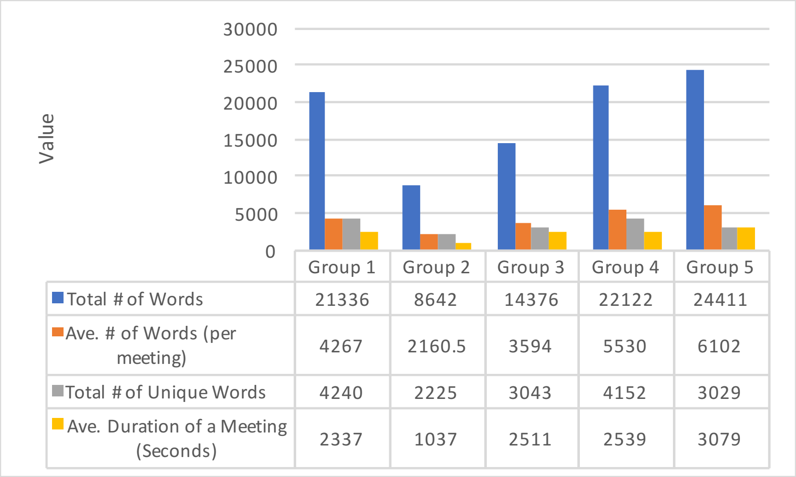
3.5.1 Dataset Description

- **Lifelog Dataset:** This dataset consists of the lifelog of one participant containing three media types: (1) the transcribed audio recordings of all conversations that a participant has had in his daily life, (2) images that were captured at regular time intervals of 30 seconds using the Narrative Clip camera, and (3) the GPS coordinates that a person was located at, when a conversation took place. We describe the statistics of our dataset as follows: The length of the dataset, i.e., the number of days that data of social interactions was collected was 60 days, the number of people with whom our participant has had conversations during the data gathering period is 32. The conversations were recorded with full awareness and consent of all the people involved. The statistics of our text data (i.e., transcriptions of audio recordings) are as follows: the number of conversations in total is 375, the number of hours for the entire conversations is virtually 21 hours, and the number of unique tokens in the dataset is 27,476. Each conversation was manually transcribed using an online professional transcription service for a price of 1 USD per minute. The transcribers occasionally marked words that they were not confident with their transcription. Using the markers, we hand corrected those words. The statistics of our image data is as follows: The number of images captured on average per day is 783, and the overall number of images captured during conversations is 2493. The Narrative Clip would not take photos if its lens is blocked, hence the variability in the number of captured images.

- **Meetings Dataset:** Our meetings dataset consists of two media types: (1) the transcribed audio recordings of all conversations of each participant, (2) images that were captured at regular time intervals of 30 seconds (using the Narrative Clip camera) by each participant's wearable camera. This camera clips on a shirt and captures first-person-view images.

  Our participants consist of five groups where each group is comprised of two fixed individuals with rare participation of a third individual. In total, we recorded the data of nine unique participants in our dataset

  Converting audio to text: We transcribed all audio recordings of the meet-



	Group 1	Group 2	Group 3	Group 4	Group 5
🟦 Total # of Words	21336	8642	14376	22122	24411
🟧 Ave. # of Words (per meeting)	4267	2160.5	3594	5530	6102
⬜ Total # of Unique Words	4240	2225	3043	4152	3029
🟨 Ave. Duration of a Meeting (Seconds)	2337	1037	2511	2539	3079

Figure 3.4. Basic statistics of our dataset

ings using an online transcription service[1] at a cost. The transcription error (due to human error) according to the service is 1%. The transcriptions of the conversations are time-stamped at fixed time intervals of one minute. Later in this section, we explain how we use the time stamps for synchronizing the transcribed text with other signals.

Basic statistics: Figure 3.4 presents some basic statistics of our dataset The report includes per-group statistics, such as total number of words in all four meetings, average number of words per meeting and the number of unique words in all four meetings. Since our dataset is real-world, there are visible differences between statistics and meeting behavior of different groups.

This will enable us to examine the effectiveness of our memory augmentation system in a real-world setting.

Extracting Text Segments: we first extract text segments using the Texttiling [65] segmentation algorithm. This algorithm uses word co-occurrence patterns in sentences to detect changes in the topic of a segment.

Texttiling [65] is “a technique for subdividing texts into multi-paragraph units that represent passages or subtopics”. It utilizes patterns of lexical co-occurrence and distribution as discourse cues for identifying major subtopic shifts. We note that the texttiling algorithm cuts segments in documents only at sentence endings. Therefore, one segment would contain one sen-

[1] http://www.rev.com

tence at least. Our purpose behind using texttiling is to split a conversation into topically coherent segments, such that we would be able to assess the similarity of each segment of a conversation to what a participant recalls about that conversation.

Computing memorability: we recorded four meetings per each group over four weeks. Immediately before the start of each meeting we held an interview with each participant, asking them to describe everything they remembered from their previous meeting. Thus one week after each meeting, we held what we call a recall session where each participant described everything one could recall while being audio recorded. Then, similarly to the meetings, the recordings of the recall sessions were transcribed.

Finally, by computing the Latent Semantic Indexing (LSI) [67] topic similarity (after preprocessing steps such as stop words removal, converting all words to lower case, etc.) on all segments of a meeting we created a topic model of that meeting. The number of topics per each conversation was set to 20 in order to be able to compare the results. We note that since the number of topics is kept the same across all meetings and this is merely for similarity comparison between meetings and recall sessions we only used one number of topics. Subsequently, by querying the model with the corresponding recall sessions, we automatically computed how memorable each segment of a meeting was for an involved participant. This was done by comparing every segment of the meeting with the corresponding recall sessions based on the LSI topic model on the segments. Finally, the similarity between each segment and the corresponding recall sessions were computed based on cosine similarity. Therefore, by computing the semantic similarity between each segment and a segment we produce objective labels of how much a participant remembered or forgot.

Finally, we compute the average sum of all similarity scores for each meeting to compute one similarity score per each participant and per each meeting. Moreover, for each set of four meetings of two participants we use a softmax function to normalize the similarity scores against one another. Softmax function takes a number of input scores and normalizes each of them to a score between 0 and 1, such that the sum of all input score would be 1 in the output. By doing so we compute the similarity scores presented in Tables 3.2 and 3.2. We note that the scores reported in the two tables all represent the average scores of two similar conditions (e.g. two meetings with condition B).

3.5.2 System Output using the Lifelog Dataset

In this section we present some results of an empirical study which are sample output summaries of our system. We show some qualitative results of our system summarizing one week of our participant's daily life tracking the evolution of topics over the week.

For training our high dimensional LBPH classifier we created training data for face recognition of 28 people who appeared in our dataset. Because our participant was automatically capturing images with the Narrative Clip not only when he was involved in conversations with people, but also when he was not involved in any conversations those images could be used for training the face recognition classifier. Table 3.1 presents a sample output of three conversations which were automatically connected to one another by SILAS as similar topics. We explain the figure in order to explain the context. In Table 3.1 the images in the left and right column show the same person, who was a student working with our participant in a team on a project about image processing. The student was co-supervised by our participant and the two people appearing in the image in the middle column. Each event is time stamped as shown in the third row of the table. In the event presented in the middle column, the team is meeting at a cafeteria discussing the tasks of the student and the progress of the project. As illustrated in the table, the topics in all three columns are related to various image processing concepts. In the event presented in the middle column, for example, there was a discussion about future tasks of the student and what features could be used in a classification of images taken indoors and outdoor environments. As could be seen in Table 3.1, the presented topics contain some noise which is justified due to the size of the conversation (i.e., number of sentences) from which LDA topics are generated, however, in our evaluation presented in the next subsections we came to the conclusion that when presented with the topics, a user who has been involved in the conversation, can recall the details of a certain conversation, therefore, showing that topics can be used as effective memory cues for recalling past conversations.

For summarizing one week of the participant, from which three events are presented in Table 3.1, our the dDTM topic model is used which will be presented in the following chapter. For this one week data, dDTM automatically computed a total of twelve different chains of topics that their evolution were tracked over time.

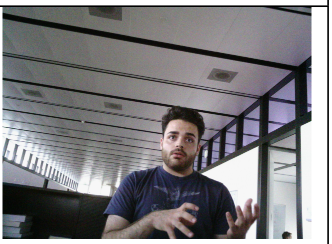
Empirical Results		
		
Topic 1 matcher extractor compares sift features surf assumption region understand equal	Topic 2 images scenario kitchen eating indoors figures sift surf ground remind	Topic 3 svm surf sift tweaked decent files organized images parameters value
13.07. 2015 16:37:54	15.07. 2015 09:50:09	15.07. 2015 15:57:34
Location: A	Location: B	Location: A

Table 3.1. Sample output of SILAS showing three event snippets automatically detected as similar

3.5.3 Episodic Memory Enhancement Test using the Lifelog Dataset

Previous studies show that reviewing a person's daily activities of the past, could improve the person's episodic memory [62, 84, 121]. As mentioned before, the most important motivation behind this study was to build a system with application in health care, and memory enhancement. In this section, we test the effect of reviewing daily social interactions (using our summarization system) on human memory. To the best of our knowledge, all related studies so far, have shown only the effect of replaying raw and unprocessed recordings of a person's life in the form of video, image, or audio, etc. on human memory. However, the drawback of replaying the raw lifelog data to its owner is that he would need the same amount of time as the original experience to review it, and this might prove impossible and inapplicable for many people with busy lives. On the other hand, SILAS summarizes a one hour meeting into several columns similar to the three columns presented in Table 3.1 and if an average person spends a minute for reviewing each column, reviewing the whole meeting would require several minutes. We would like to emphasize that the experiment we are presenting here is a common practice in the field of psychology of memory, and that this experiment was done under the supervision of a qualified clinical psychologist. We use the ABAB case study design, which is a popular method for testing the effect of an external intervention on a single test subject and has proved to be highly accurate in previous work [75]. This method has been used in many different domains and applications. For the sake of completeness, here we briefly describe the method.

We start our experiment with the hypothesis that using SILAS for reviewing daily social interactions, will improve one's episodic memory. The first A in the ABAB represents a baseline test in which the single test subject is tested without any intervention. In our case, this would be assessing our participant's episodic memory of one week without him reviewing that week using SILAS. The participant is asked to write down the memories of his one week without using SILAS after two weeks pass the target week (i.e., the first A). In the next phase of the test which is the first B, we ask our participant to review another week (different than the first week) of his life using SILAS. This phase is designed for treatment measurement. Our participant reviews this week using SILAS for four days after it has passed and then stop reviewing it. Similar to the first A, after two weeks pass the target week (i.e., the first B) our participant is asked to write down his memories of the target week. Subsequently, in the second A phase which is the withdrawal from treatment phase, our participant does not use SILAS to review a third week of his life, similar to the first baseline (the first A). This phase measures the effects of extinction, or the withdrawal of the positive reinforcer, on behavior. In this phase we are determining if the second baseline returns to the original or if the behavioral change we experienced will continue. Analogously to the first A and B, we ask our participant to write down his memories of that week, after two weeks have passed it. And finally the second B is when we re-apply treatment which is the use of SILAS for reviewing daily memories. This phase is to discover the effect of re-applying treatment on the results. Analogous to the first B, our participant reviews the target week (i.e., the second B) using SILAS for four days after it ends, and also writes down his memories of the week after two weeks of the target week.

Our participant reviews the four weeks as described above, each week is reviewed for four days, each day for 45 minutes. These numbers were determined by a qualified clinical psychologist based on an estimation of how much time was needed to thoroughly review the entire week. After two weeks, the effect of the reviewing is assessed by having our participant writing down all the events that occurred in the reviewed week. Furthermore, the three next weeks are also being reviewed and assessed with the same procedure. All four reports written by our participant are submitted to three qualified psychologists who individually assess and compare the four reports according to a questionnaire which measures the amount of information that the participant has been able to recall. The questionnaire examines the number of recalled events for each week, along with the level of detail describing the events.

This experiment would show whether reviewing daily life using SILAS has had an impact on recalling memories and by withdrawing treatment and re-applying it, precisely examines the effect of SILAS on memory improvement.

Our results show that the three psychologists unanimously confirmed that the use of SILAS has significantly improved the episodic memory of our participant and that his memory has been trained to recall more past events with a higher level of detail over time. The four weeks that were reviewed by our participant were four consecutive weeks. The results indicate that our participant had an improvement in his episodic memory and recalling past events. Figure 3.5, exhibits the average scores of the number of recalled past events given by the three psychologists in percentage. In order to show the results in percentage we assumed the highest possible mark in the questionnaire to be 100% and computed the raw scores in percentage for all of the four phases (ABAB). As illustrated in Figure 3.5, the results of this study confirm that the episodic memory of our participant increasingly improved in recalling the events of the four consecutive week. In ABAB design which is specially created for single test subjects, the memory recall

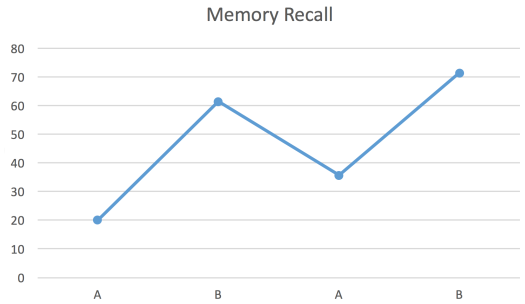


Figure 3.5. Enhancement of Episodic Memory using ABAB Design over Four Weeks

percentage increases during the intervention phases (i.e., the two ‘B’s) which is also true in our case. Although, when withdrawing from using SILAS the percentage of memories recalled should plunge, if it decreases too much so that the percentage of the two ‘A’s would be the same, its an indication that the treatment (i.e., using SILAS) is only temporarily effective and not so effective in a longer term. However, our results prove that the use of SILAS has a more lasting effect over the duration of the test. Hence, we proved that use of SILAS is effective in enhancing one’s episodic memory.

3.5.4 Episodic Memory Enhancement Test in Meetings

Study Design: In this section we present a study design and evaluation for assessing the effect of using SILAS in a workplace meetings scenario. We captured four weekly meetings of five groups where each group consists of two individuals. Each group had a weekly meeting, roughly on the same day each week. One week after each meeting of a group (before their next meeting) each of the participants were asked to recall the details of what was discussed in their previous meeting while being audio recorded. We refer to such session where one recalls the details of a previous conversation as a recall session.

Therefore, for each meeting there is a recall session recorded per each participant. Our experimental setup is based on the ABAB case study design, which is a popular method for testing the effect of an external intervention on test subjects and has proved to be highly accurate in previous work [75]. Each A in ABAB stands for a baseline condition or control (i.e., no intervention) and each B stands for an intervention phase. In the case of our study, an intervention consists in reviewing a previous meeting using a set of event snippets generated by SILAS

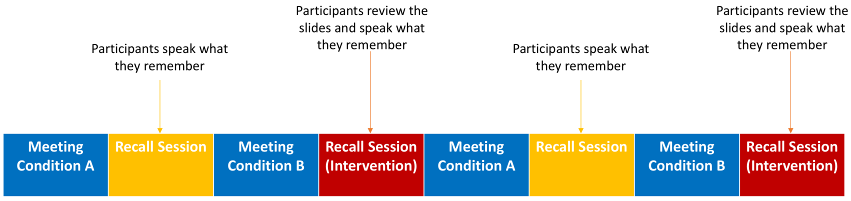

Participants speak what they remember
Participants review the slides and speak what they remember
Participants speak what they remember
Participants review the slides and speak what they remember
Meeting Condition A
Recall Session
Meeting Condition B
Recall Session (Intervention)
Meeting Condition A
Recall Session
Meeting Condition B
Recall Session (Intervention)

Figure 3.6. Our ABAB case study design

for five minutes before attending a recall session. Event snippets are put into a slide deck where each snippet is presented as one slide. Generating the output slides is done automatically by the tool.

Therefore, each participant is exposed to two B conditions (where one first reviews the last meeting using the generated event snippets) and to two A conditions (where the participant is given five minutes time to recall the details of the previous meeting without any memory aid).

We use different permutations of ABAB on different groups of participants to ensure any effect, e.g. learning over time, is controlled. Figure 3.6 illustrates a cycle of our experimental setup when the ABAB permutation is used.

In other words, participants are either subject to the “intervention condition”, (i.e., they review the memory cues shortly before the recall session and their next meeting), or they are in the “control condition”, (where they have to recall their past meeting without any memory augmentation). Figure 3.7 shows one of our participants reviewing a past meeting in a condition B (i.e., intervention) with using the memory cues generated by our memory augmentation tool. We emphasize that in a condition B, a participant is given a maximum of five minutes to review the memory cues which are presented as slides. After they finish the five minutes reviewing they are asked to attend the recall session where they try to recall and speak all the details they remember about the content of their previous meeting.

Such experimental setup leads to the possibility of testing each participant from each group twice with the intervention condition which in effect examines the efficacy of our memory augmentation tool. This experiment aims to answer the research question behind this research which is whether or not it is possible to use conversation logs for effectively augmenting human memory regarding the detailed contents of previous meetings. One of the novelties behind our study is that since meetings are held in a room and the movements of the participants are very limited, the images captured by their wearable cameras barely change. Consequently, the memory augmentation process, will be mostly dependent on the keywords shown to the participants. Additionally, in this study we ask the

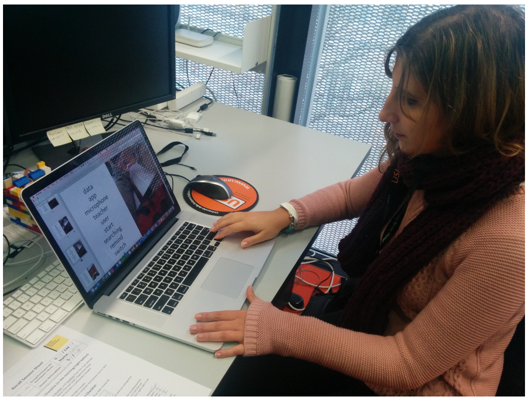


Figure 3.7. One of our participants reviewing the memory cues in a condition B

participants to recall all details of the *contents of previous meetings* and not only remembering general topics.

Experimental Results: As discussed earlier, studies show that reviewing daily activities of the past, could improve one's episodic memory and facilitate recollection of past memories [62, 84, 121]. All of these studies, however, have shown only the effect of replaying raw and unprocessed recordings of a person's life in the form of video, image, or audio, etc. on human memory. This is while, the drawback of replaying the raw recorded data (e.g. a meeting) to its owner is that she would need the same amount of time as the original experience to review it, and this might prove impossible and inapplicable for many people with busy lives. As a result, in this study we gave a maximum of five minutes to each participant to recall a previous meeting. It is noteworthy that in our user study, in many cases, the participants did not use the entire five minutes at their discretion.

We compare the difference in the recall values within test subjects computed objectively using the process explained above. We first start our evaluation by conducting a detailed comparison of recalling a past meeting per each participant between conditions A and B. We compute the recall values for each participant per each condition. Then we compare the recall values between conditions A and B per each participant and carry out an Analysis of Variance (ANOVA) test to examine the significance of our intervention.

Table 3.2 presents the results of this comparison. By looking at the p-values in the table we observe that five participants out of ten are positively impacted by our memory augmentation tool with a confidence of above 90% (as suggested by the ANOVA test). As we noted previously, out of the ten participants examined in this study, we had nine unique participants. In fact, participants specified as 5 and 7 are the same person attending two different sets of meetings. We observe that none of these two attempts by this participant in our user study has been completely successful. We looked into the case of this participant in more detail to research reasons of this failure. We noticed that in the case of group 4 (consisting of participants 7 and 8) the two participants were traveling together on the week in between the two condition Bs and having frequent chats during the trip. These frequent conversations in between the conversations that we recorded, are the likely causes that have influenced the results of our experiment. Although, when recruiting the participants we asked them not to have meetings in between the ones that we recorded, in the case of this group the joint trip was out of our control. This point also reflects on the difficulties of conducting user studies in-the-wild with real-world data.

The impact of the joint trip was evident also by comparing the results of this group against group 3 (participants 5 and 6). To explain this point further, groups 3 and 4 had their meetings every week on the same day. However, according to the p-values the impact of our intervention on participant 5 was more than the impact on participant 7. As participants 5 and 7 are the same person, we found no other causes other than the trip to distinguish the performance of this participant between when he had meetings in group 3 and group 4. Another point that we learned about participant 5 (i.e., or 7) was that he had been very busy during the same period which may explain the poor performance. Furthermore, this participant was the oldest among our participants with an age of above 50.

Another participant who was not significantly impacted by the memory aid tool was participant 9 from group 5. Further analysis of the meetings of this group revealed that participant 9 was a master student meeting a PhD student with whom she was working on her master thesis. In their second meeting which was in condition A, the professor had also joined their meeting. Our results showed that the master student was able to recall this meeting more strongly and better than a condition B meeting. We believe that the participation of the professor who had authority over the student was a factor that influenced the outcome of our experiment. That is also due to the fact that this was the only time the professor attended the meetings.

Overall, we conclude from this experiment that out of the nine unique participants that were recruited in our user study, five of them were positively impacted by our intervention with p-values of less than 0.10. A larger scale study with more participants may observe reduced noise and hence better results.

In a second experiment, we compute the collective recall values per each group. That is, we compute how much each group was helped with respect

Table 3.2. Comparison of recall values under conditions A and B for each participant. The p-values computed by the ANOVA tests presenting the degree of error in significance of our intervention along with the experimental conditions are reported.

	Exp. Cond.	Cond. A	Cond. B	p-value
Particip. 1	BABA	0.2437	0.2578	0.02
Particip. 2	BABA	0.2437	0.2562	0.13
Particip. 3	ABAB	0.2269	0.2730	0.07
Particip. 4	ABAB	0.2261	0.2738	0.00
Particip. 5	AABB	0.2398	0.2601	0.35
Particip. 6	AABB	0.2370	0.2629	0.06
Particip. 7	ABBA	0.2495	0.2504	0.85
Particip. 8	ABBA	0.2443	0.2556	0.22
Particip. 9	BAAB	0.2492	0.2507	0.85
Particip. 10	BAAB	0.2425	0.2574	0.03

to remembering a past meeting in a condition B against a condition A. For this purpose we compute the average of recall values of a pair of participants for each meeting. Then we compare the resulting values based on the two experimental conditions.

Figure 3.8 presents the results of this experiment. We observe that in all groups, the collective recall value of the group in a condition B (i.e., intervention using our memory augmentation) outperforms that of a condition A. Furthermore, the error bars on the plot present the standard error. We observe that in the case of groups 1, 2 and 3 the error bars of the two conditions do not overlap, while in the case of groups 4 and 5 they overlap. Consequently, to make a more in-depth analysis of the results we use a one way ANOVA function for testing the significance of the difference between recall values in conditions A and B.

We present our findings in Table 3.3. The table shows the same values we presented in Figure 3.8 in addition to the p-values computed by a one way ANOVA function indicating the significance of the efficacy of our memory augmentation tool. Moreover, the first column of Table 3.3 presents the experimental condition the was used for each group. As explained earlier, different permutations of A and B were analyzed in our experiments in order to control the effects of learning over time, sequence of conditions, etc. We can observe in Table 3.3 that the p-values of the first three groups are lower or equal to 0.10. This means that for these groups we could conclude that the difference in recall between conditions

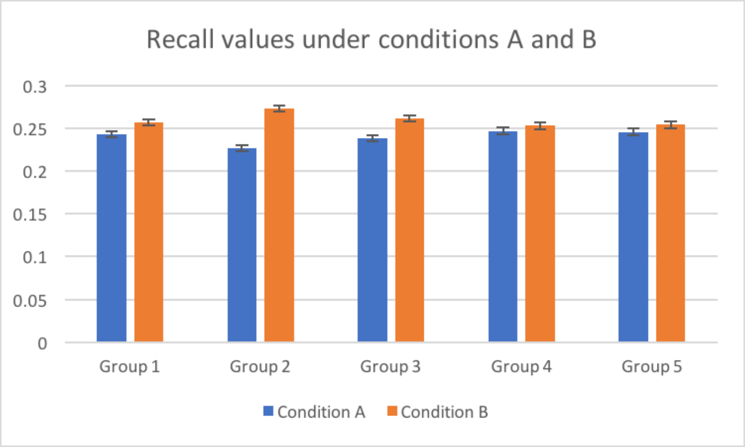


Figure 3.8. Comparison of recall values for different conditions of A and B per each group. The error bars in the figure show the standard error.

Table 3.3. Comparison of recall values under conditions A and B for all groups. The p-values computed by the ANOVA tests presenting the degree of error in significance of our intervention along with the experimental conditions are reported.

	Exp. Condition	Cond. A	Cond. B	p-value
Group 1	BABA	0.2429	0.2570	0.05
Group 2	ABAB	0.2265	0.2734	0.00
Group 3	AABB	0.2385	0.2614	0.09
Group 4	ABBA	0.2469	0.2530	0.15
Group 5	BAAB	0.2458	0.2541	0.22

B and A were significant with 90% confidence. In other words, three groups out of five showed significant positive effect in recollection of memories of a past conversation using our memory augmentation tool.

We note that our memory augmentation tool can be used not only in the domain of meetings, but also any other verbal social interactions, including informal conversations, online textual conversations on various messaging platforms as well as forums and online social media. The same methodology used to extract the salient themes of meetings could be utilized to extract memory cues from other textual communications.

3.6 Conclusions

With the rapid proliferation of wearable lifelogging devices, the interest in using such devices to record personal data and utilize that data for various human-aid applications is rapidly increasing. In this chapter we presented a memory augmentation tool which uses transcriptions of conversation logs and images captured by wearable cameras, as well as GPS coordinates to generate memory cues in order to augment one's memory with respect to a previous conversation/meeting. The memory cues generated by the tool could be used to facilitate recalling past events. We showed through a case study that the majority of our participants found the keywords generated by our approach to extraction of themes of single conversation transcripts more understandable than that of LDA.

As demonstrated in Table 3.1 SILAS also tracks the evolution of a topic over time and chains similar topics together. This topic chaining is based on the dDTM model which will be explained in detail in the next chapter.

We then conducted two user studies to objectively examine the efficacy of our memory augmentation tool. Our experimental results showed that use of SILAS is effective and shows a significant positive effect in recalling a past event on the majority of our participants.

We found out that user studies which are carried out in-the-wild are often difficult to conduct, due to various noises introduced by the behavior of participants. These are challenging factors that are hardly resolvable. A larger number of participants may alleviate the effect of noise and external causes that may have negatively impacted the results of our study, but we succeeded in finding only nine participants to take part in our case study.

We were able to experimentally show via our user study that using conversation logs for memory augmentation is feasible and leads to promising results. The findings of this study could be used to develop future memory augmentation tools.

Finally, SILAS can be extended with other data modalities such as biophysical sensor data, or time series prediction models to predict event snippets which are more important to be shown to a user (e.g. a user might forget or might need to focus on certain content). We have designed models to enable both of these features and will present them in the second part of this thesis where we discuss just-in-time information retrieval.

Chapter 4

Modeling Discrete Dynamic Topics

4.1 Introduction

With the rapid proliferation of lifelogging devices as well as online social media platforms that record user generated content in the form of temporal data streams, the interest in analyzing these streams is on the rise. Conversation lifelogs or social media textual posts can be analyzed with respect to their topics [34] or sentiments [19, 22] and the evolution of topics or sentiments can be computed over time. In the previous chapter we presented SILAS, a memory augmentation tool that summarizes one's daily conversations into event snippets. In this chapter we present a model that not only can extract the main themes of textual documents, but also can track the evolution of these main themes over time given a sequential dataset

In recent years methods for managing, exploring, and indexing large datasets of digitized text documents have become increasingly important. Topic modeling is one of these methods, and it has been used in many different applications, such as summarization and trend analysis [33] as well as information retrieval [148]. We define topic models as in Definition 1.

One basic topic model is LDA [34] (introduced in Chapter 2) that has set the basis for many other topic modeling methods since its introduction. LDA assumes that all documents are exchangeable in the entire collection, i.e., the probability of documents in the dataset is invariant with respect to permutation. Thus, the primary application of LDA was shown in exploring non-sequential datasets. However, many datasets, such as news streams, blog and microblog posts, spread over a time-line and the containing documents have timestamps [91, 92].

Based on LDA, Blei and Lafferty introduced DTM, which was initially used for exploring a sequential dataset containing all scientific articles published in the journal *Science* [33]. Given a sequential dataset sliced up based on discrete timestamps, DTM computes the topics of each time slice and chains each (supposedly) same topic over all the time slices. This model is based on the strong assumption that each topic is present over all time slices. Although this certainly holds for homogeneous datasets, such as the articles published in *Science* journal which shows yearly advancements in similar topics, it is not necessarily a valid assumption in case of less structured and highly dynamic datasets, such as streams of news, blog articles, tweets, or transcripts of lifelog conversations.

In this chapter we introduce the dDTM approach which, analogously to DTM, is based on LDA. However, dDTM relaxes the assumption that each topic must be present over all time slices of a sequential dataset. The motivation behind this model is that we empirically found out that some topic chains computed by the traditional DTM do not correlate with the highly dynamic nature of some real-world datasets consisting of tweets, blogs, and news. Hence, the computed topics are not always meaningful for making interpretations about the data (in Section 4.2.1 we will explain this point in more technical terms). In such datasets, where topics change at a high pace, some topics may appear in one time slice and disappear in the next time slices before appearing again.

In the previous chapter we conducted an experiment on a stream of real-life natural conversations of an individual recorded over a span of two months. We showed that "the topics of daily natural conversations of an average person are intermittent." In other words, certain topics disappear for some time before reappearing again.

Motivated by the above-mentioned issues, we propose a new model which overcomes the limitations of DTM. We evaluate our model using two datasets.

The first one is the publicly available `Signal Media` dataset containing over one million news articles collected during one month [42]. The second one is made of tweets posted by the British Twitter Channel `NFCWorld` over the last three years. This channel posts tweets regarding emerging technologies and we crawled its tweets using the Twitter API.

The most important contributions of the research presented in this chapter are:

- We present a new algorithm for topic modeling which overcomes the limitation of the original DTM algorithm. That is, our model does not necessarily require a topic to be chained to another topic in the next time slice, but it presents some flexibility in computing evolution of topics over time.

- We compare our dDTM model against DTM in terms of likelihood on held-out data. This evaluation method compares the quality of topic chains computed by each model. Moreover, we present qualitative results that demonstrate how our method allows the exploration of data streams (i.e., large sequential datasets) of documents to identify emerging topics.

- We evaluated the proposed dDTM approach on a large stream of news and blog articles as well as our collection of tweets on emerging technologies by the `NFCWorld` channel.

The remainder of the chapter is organized as follows. Section 4.2 presents necessary background information about dynamic topic modeling. In Section 4.3 we present our novel dDTM method. Section 4.4 evaluates our method compared with the original DTM in terms of representation of the computed topical chains on held-out data. Finally, Section 4.5 concludes this chapter and presents some insights into future work.

4.2 Background

In this section, we present the background on topic modeling by describing DTM [33] which is essential for explaining our model. We previously defined LDA in Chapter 2.

4.2.1 Dynamic Topic Model

DTM [33] assumes that a given sequential corpus of documents is divided into different time slices by using the document timestamps. It models topics of each time slice starting from the first one using LDA. Then, it chains together the natural parameters of a topic at each time slice, which evolves linearly, using a linear Kalman filter [73] over consecutive time series. Thus, the parameters of each topic, $\beta_{t,k}$, are chained together in a state space model that evolves with a Gaussian noise. Subsequently, DTM draws each topic β such that:

$$\beta_{t,k}|\beta_{t-1,k} \sim \mathcal{N}(\beta_{t-1,k}, \sigma^2 I) \tag{4.1}$$

where $\mathcal{N}$ is a logistic normal distribution. In this way DTM connects the same topics over time. We note that, the σ parameter shown in Equation 4.1 allows for variation in a topic over two subsequent time slices. By assigning small values to σ the model ensures that one topic would not evolve to a different topic over two subsequent time slices.

Likewise, there is a similar evolution process for the α parameter, as α impacts the per-document topic proportions θ that is drawn from a Dirichlet distribution.

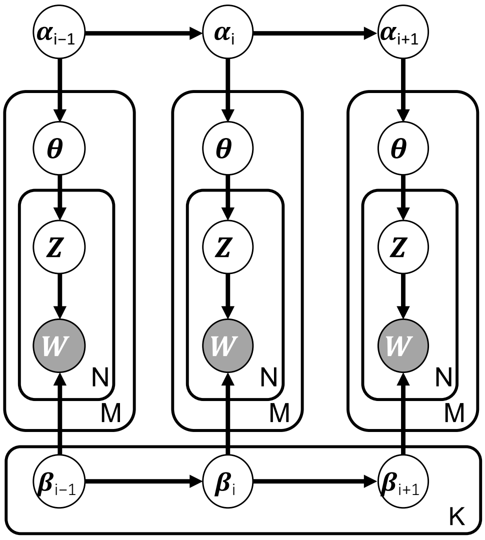


Figure 4.1. Graphical model of DTM.

$$\alpha_t|\alpha_{t-1} \sim \mathcal{N}(\alpha_{t-1}, \delta^2 I) \tag{4.2}$$

The graphical model of DTM is illustrated in Figure 4.1.
In spite of being a powerful model for statistical interpretation of a sequential corpus, DTM comes with two limitations:

- In DTM each topic at time slice t should be connected to a topic at time slice $t-1$. As discussed in the introduction, the design of this algorithm is based on the assumptions made about the dataset made of articles from the journal *Science*. However, in the case of real-world discussions (e.g., on micro-blogging websites) the focus on some topics changes suddenly and new topic trends may emerge. In particular, the majority of posts we can observe in one time slice would cover a set of topics which might be completely different from the previous one. Analogously, in the case of articles published by news channels, it occurs that the focus of the news reports suddenly changes as novel situations arise (e.g., a news about a terrorist attack may disappear from the media until another one takes place or the suspects are arrested and put on trial). Moreover, it is reasonable to assume that in such data streams, different topics may emerge, disappear and after some time re-appear.

- If there are new topics, i.e., did not exist at the first time slice of the dataset, DTM would not immediately show those topics once they emerge. This is because it assumes small changes in a topic over time. As shown in Equation 4.1, a topic evolves linearly with Gaussian noise, while the mean of the Gaussian is the topic in the previous time slice and by assigning a small value to σ it allows a small variation in the topic over two consecutive time slices.

The effect of such modeling is that if a new topic emerges in the data the model will not immediately respond to it. If the topic persists over a number of time slices DTM will gradually take it into account in the evolution process. This characteristic causes DTM to be unable to immediately detect an emerging topic. However, in the next section we propose a novel topic model which overcomes the limitations of DTM, hence it suites the above-mentioned real-world applications better. Our model, differently from DTM, uses a non-linear evolution process for the topics, and also it does not rely on the approximations of the evolution of the topic proportions over time.

4.3 Discrete Dynamic Topic Model

Similar to DTM, our model is also based on LDA [34]. Given a sequential dataset (i.e., stream of documents) divided into different time slices, as shown in Figure 4.2, LDA model is applied for computing topics in each time slice. In our model we use LDA for extracting and chaining topics over different time slices together in order to compute the topic evolutions. The reason behind using LDA is that, compared with other topic-modeling algorithms, it has shown strong results for detecting topics in texts [34]. Additionally, LDA is computationally scalable on large datasets.

Differences with DTM: In our model we chain together the multinomial distributions over observed words (i.e., topics) to estimate the latent topical chains over time. Differently from DTM [33], we do not estimate the α parameter which influences the topic proportions over subsequent time slices. The reason is that in the case of streaming data, the data over two subsequent time slices might not always be similar enough, hence an estimation of topic proportions based on the first time slice might not be correct.

Indeed, in dDTM we relax the assumption that a topic will always be present in all time slices. Thus, it is not reasonable to adjust topic proportions of one time slice based on estimations from the previous time slice. We note that in a setting where a topic is not present at the time slice t_0, but it is present at t_1, the topic proportions of t_0 are not a reliable approximation for the topic proportions of t_1. Therefore, an estimation based on the current time slice t would not be reliable for the next time slice $t + 1$. Our goal is to relax the assumption that all topics need to be present over all time slices, so we do not estimate the topic proportions over time, but we rather keep the α parameter fixed. In other words, since we deal with highly dynamic data, such as stream of news and tweets, the assumption made by DTM for the articles of the journal *Science* whose documents show gradual advancements of the same subjects discussed over time would not hold. This means that the estimations made based on tweets of the last time slice might not be correct for the news or tweets of the current one.

Another difference with respect to the original work on DTM is that we do not restrict the number of documents per time slice to a fixed number.

Further differences of our model compared with DTM include use of a Markovian state space model which will be elaborated in the following.

dDTM Model: Our model, illustrated in Figure 4.1, estimates the topical chains based on the multinomial distributions over words (i.e., topics) in a non-linear fashion as opposed to DTM where topics evolve linearly.

For this purpose, from each LDA generative process per time slice two matrices of per-document topic proportions and topic vectors. The former matrix is in the size of $documents*topics$ and the latter is in the size of $topics*vocabulary$. Furthermore, to infer topic chains over different time slices we use a Hidden Markov Model (HMM)[106], which similarly to the Kalman filter, implements an Expectation Maximization (EM) algorithm, named Baum-Welch [45].

However, differently from our method, DTM uses a Kalman filter, which makes the assumption that the evolution of a topic in the state space model is linear, hence it is suitable for continuous-time settings. Whereas, in the case of HMM used by our dDTM there is no such assumption, thus it is suitable for discrete-state settings.

The purpose of dDTM is to discover the latent structure of chains between the same topics over time. The non-linearity of our model enables relaxation of the assumption that every topic should be present in all time slices. Therefore, our dDTM is based on the following generative process:

1. For each topic $k, 1 \leqslant k \leqslant K$:

   1.1. Draw $\beta_{0,k} \sim \mathcal{N}(\mu, \sigma)$.

2. For each document d_t at time t:

2.1. For each topic k and m identified topics from previous time slices:
From the Hidden Markov Model draw
$\beta_{t,k}|\beta_{t-m,1..k} \sim \mathcal{N}(\beta_{t-1}, \sigma)$.

2.2. Draw $\theta_t \sim \text{Dir}(\alpha)$,

2.3. For each word:
Draw $z_{t,n} \sim \text{Mult}(\theta_t)$,
Draw $w_{t,n} \sim \text{Mult}(\pi(\beta_{t,z_t,n}))$.

Note that π is a function which maps the multinomial natural parameters to the mean parameters:

$$\pi(\beta_{t,k})_w = \frac{\exp(\beta_{t,k,w})}{\sum_w \exp(\beta_{t,k,w})}.$$

Figure 4.2 show the graphical model representation of dDTM.

The vectors of the distributions over words (i.e., topics) of each time slice are given as input to the Baum-Welch algorithm. These topic vectors represent topics mapped onto the simplex (i.e., a vector that contains the entire vocabulary). Thus, each topic vector is in the size of the entire vocabulary, which is extracted from all documents present in every single time slice. Each entry in the vector represents the probability of a word to appear in the topic.

Furthermore, the probabilities computed by the Baum-Welch algorithm are based on a Gaussian distribution, hence the model is computed with Gaussian emission probabilities, and the evolution of each topic is with a Gaussian noise. This will result in having all similar topics throughout various time slices chained to one another.

As mentioned earlier our model uses an HMM and is based on the Markov assumption. The Baum-Welch algorithm computes the HMM model probabilities using the Forward-Backward algorithm [27, 28], which is responsible for belief propagation in the underlying Bayesian network of an HMM, and it consists of computing the network probabilities in a forward and backward pass. For more explanation about this algorithm we refer to [124].

In order to determine the optimal number of output states (i.e., the number of topic chains) of HMM, dDTM always computes a trade-off between the Bayesian Information Criterion (BIC) [119] for a given configuration against the number of states. Hence, it automatically connects the similar topics by maintaining an optimal number of states (i.e., topic chains). Given a fixed number of topics, K, dDTM is able to find for each time slice a set of topics whose size is at least K.

We used variational inference based on the HMM inference for computing dynamic topics. The main idea behind variational-inference methods is to assume a simple family of distributions over the latent variables, indexed by free variational parameters. Furthermore, they find the member of the distribution family which is closest to the true posterior in terms of Kullback-Leibler divergence [145].

After the model parameters are learned by the Baum-Welch algorithm, we applyed Viterbi decoding [143] to find the most likely sequence of hidden states. The Viterbi algorithm is a backtracking algorithm used for computing the most probable state sequence among all possible paths.

4.3.1 Further Details on dDTM

So far, we explained all major components of our model dDTM. In the this section we elaborate further on the use of HMM and BIC.

HMM: It is a generative probabilistic model in which a sequence of observable outputs, is generated by a sequence of hidden states. The hidden states are internal and thus unobserved. The transitions between states are based on a Markov chain. This means that, given a state space, the probability distribution of the next state depends merely on the current state and not on the sequence of all states that precede it. Each transition from one state to another is weighed using either an initialization probability matrix (which could be assigned randomly) and a transition probability matrix. An HMM could be used to statistically model a Markov process with unobserved states.

dDTM assigns a state per each topic chain. A transition between two topics in the same time slice is restricted using model transition probabilities.

Among the various problems that HMM can solve, we are interested in *how to adjust the model parameters $\lambda = (A, B, \Pi)$ to maximize P(O $\mid$ λ)*, where A is the state transition probability distribution, B is the observation symbol probability distribution, Π is the initial state distribution, and O is the output observation sequence that HMM generates. To solve this problem the Baum-Welch algorithm is used. This algorithm implements an Expectation-Maximization algorithm to find the maximum likelihood estimate of the parameters of an HMM, given a set of observed feature vectors.

Furthermore, based on the state transition probability, the model moves to the next state and a new observation will be selected till all T observations in the sequence are generated. Given the observation sequence, the same process can be used to model an HMM.

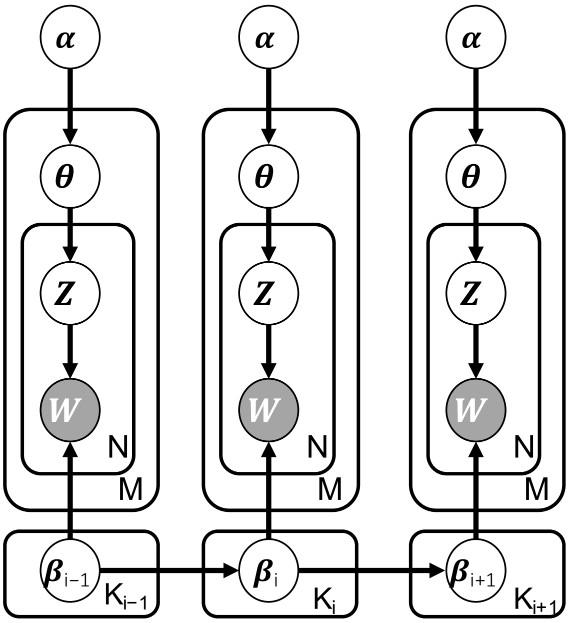


Figure 4.2. Graphical model of dDTM

Determining the optimal number of topic chains: While the combination of Baum-Welch and Viterbi algorithms as explained in the previous subsection could automatically discover the latent structure of the evolution of topics in a temporal dataset, they still would require the number of dynamic topics to be given manually. In order to circumvent this limitation, dDTM uses BIC to discover the best number of topic chains automatically.

BIC [119] is a measure used for selecting the best model among a finite set of models. Although adding more hidden states to an HMM would lead to a higher likelihood and thus a better fit to the data, after a certain point this could result in overfitting. Thus, BIC enhances a penalty for each added parameter or, in the case of our problem, an added topic chain. Therefore, it finds the best trade-off between the number of states and the model fitting the data. BIC is formally defined as:

$$BIC = -2 \cdot ln(L) + k \cdot ln(n) \tag{4.3}$$

where n is the number of data points (i.e., inputs), k is the number of parameters to be estimated and L is the maximized value of likelihood of the model. For using BIC we initialize the HMM with a number of topic chains equivalent to the number of topics n (to be derived from each time slice) given by the user. Then, we repeat re-initializing the HMM with $n+1$ topic chains until we reach an upper bound of $n+20$ (which is set). We empirically found out from a number of datasets that an upper bound of $n+20$ is a suitable choice, while this could be even a bigger number. For each run we measure the BIC value, and we select the model with the lowest BIC value. In this way dDTM allows the computation of dynamic topics over time.

4.3.2 Discussion

In the case of news, blog articles, and tweets, there is less control over the quantity of the incoming data or the pace of changes from one trending topic to another. DTM [33], which is used for tracking evolution of topics over time, fails to capture new topics and their discrete evolution, hence it is not a suitable option for detection of new emerging topics in such datasets.

We empirically found out that DTM has two main limitations: (1) if a topic cannot be observed in the data at the time slice zero, it would be tracked over time only with a lag. This is due to the fact that DTM evolves linearly with a Gaussian with a mean which is set based on the model parameters of the previous time slice; (2) each topic is assumed to be chained to similar topics of all subsequent time slices, while our model relaxes this assumption and allows topics to disappear and appear again over time.

To the best of our knowledge, there has been no previous research work addressing these limitations. Our model overcomes DTM's limitations and as we will show in Section 4.4, it is superior to the original DTM when handling streaming data such as news, blog articles, or tweets.

4.4 Evaluation

In this section we first describe the datasets used for our experiments along with the evaluation methodology, then we present the results.

4.4.1 Dataset Description

We used two datasets for evaluating our model. One is the `Signal Media` dataset and the other one is a collection of tweets from the `NFCWorld` Twitter channel downloaded using Twitter API. These two datasets are different in size, length of documents, and span different time windows. We chose them in order to show the performance of dDTM in different real-world situations. Indeed, as we will see, our model can be applied to data streams that are of different size, duration of time, and which contain long documents (e.g., news and blogs) as well as short documents (e.g., tweets).

Signal Media Dataset: This is a publicly available dataset which contains over one million timestamped news articles collected from September 1 till September 30, 2015.

The dataset is made of news and blog articles from different sources which include major news channels, such as Reuters, and local sources of news and blogs. The dataset contains 265,512 blog articles and 734,488 news articles. The average length of an article from this dataset is 405 words. Analyzing the data, we found that it also contains articles published on dates other than the above-mentioned time interval, but since their count is small compared with the rest of the dataset and can not be effectively used for time-based analysis, we decided to discard them. The total number of articles was reduced to 985,867.

For our experiments, all the news articles were preprocessed by removing stopwords, URLs, tokens not starting with alphabet letters, punctuation marks, and words which occur less than five times.

NFCWorld Dataset: This dataset contains all the tweets that were published by the `NFCWorld` Twitter channel over three years (from April 15, 2013 to April 12, 2016). According to their official Twitter web page, the tweets published by this channel report on emerging technologies.We collected the tweets using the Twitter API[1]. The number of tweets in this dataset is 3,374. For our experiments, tweets were preprocessed by removing stopwords, URLs, hashtag signs, tokens not starting with alphabet letters, punctuation marks, and words occurring less than three times.

4.4.2 Evaluation Methodology

In order to evaluate our model quantitatively, we examine the log likelihood of a model produced by dDTM on held-out data. We train our model on the data of the first five time slices and then evaluate the model based on the log likelihood of the model fit on data of each of the next time slices iteratively. For instance, we first train a model based on the first five time slices and then compute the log likelihood of the model fit given the data from the 6th time slice. Then, we fit the model with data of the first six time slices to compute the log likelihood given the

[1] https://dev.twitter.com/rest/public

Model	`Signal Media`	`NFCWorld`
DTM	276,473,612.34	431,603.59
dDTM	206,390,120.21	379,685.63

Table 4.1. Average negative log likelihood of each model predicting next year's topics on the two datasets (the lower, the better).

data from the 7th time slice, and we continue this process iteratively till the last time slice. This evaluation method is standard in the domain of topic modeling, and it was also used in the original publication on DTM [33]. Therefore, we can compare the predictive power of our dDTM model against DTM which is our baseline. Additionally, we compute the precision and recall of the topic chains produced by dDTM in order to examine the quality of the dynamic topics.

We also present further qualitative results showing the performance of dDTM in modeling topics and identifying topical trends and changes immediately as they take place.

4.4.3 Experimental Results

In this section, we describe different type of experiments we conducted for the evaluation of dDTM.

First Experiment: We applied dDTM and DTM to the `Signal Media` and `NFCWorld` datasets in order to extract dynamic topics. Given the size of the `Signal Media` dataset, we set the number of topics to be extracted from each time slice to 30 and the number of time slices to 30, while for the `NFCWorld` dataset, given its size, we set these numbers to five topics and twelve time slices. We set the number of topics only based on what we found feasible to represent in this study, however, as in LDA or DTM the number of topics could be defined to any desired value. We set an upper bound for the number of topic chains to be computed by dDTM to 20. This means, that for example in the case of `NFCWorld` dataset, dDTM will find the model with the best likelihood from 30 up to 50 topic chains.

We compare dDTM against DTM based on the average negative log likelihood. As we can see from Table 4.1, our proposed model performs better than DTM. Note that we report the average negative log likelihood, therefore a lower value is better.

Results obtained on the two datasets suggest that our model achieves a higher likelihood on the next time slices' topics.

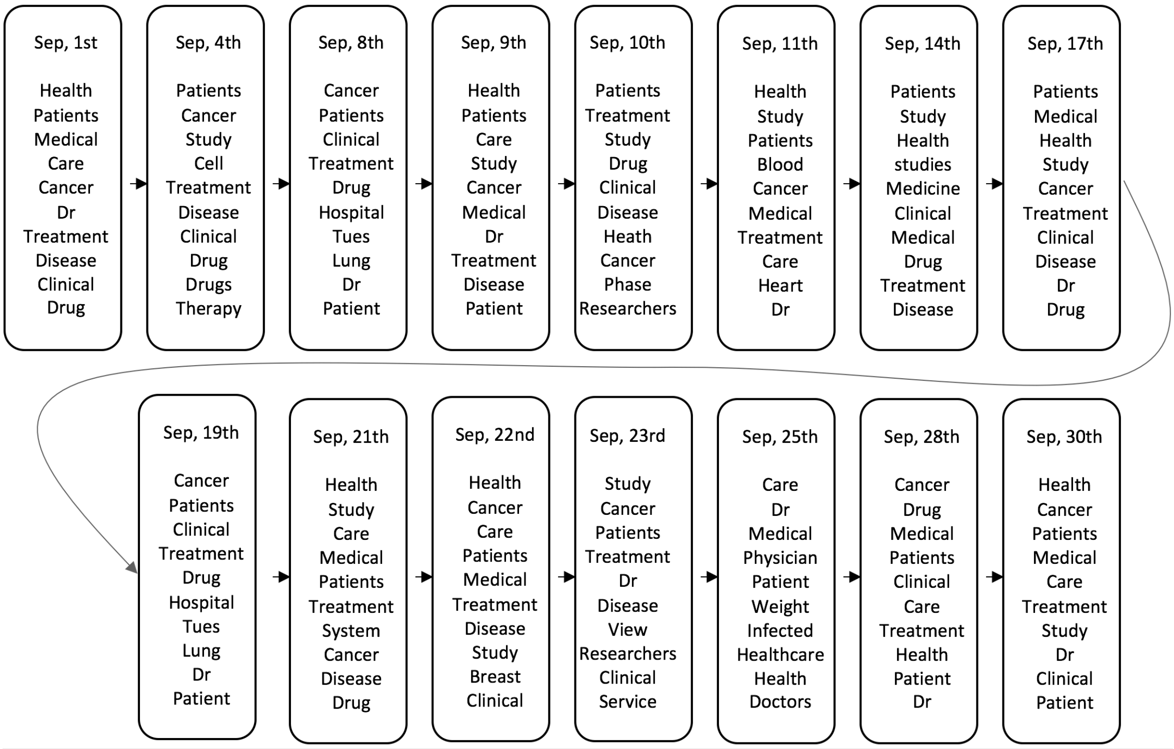


Figure 4.3. Continuation of a Topic on Health Care from the `Signal Media` dataset

This experiment confirmed that although DTM is very effective in extracting topics from collections which change slowly over time (e.g., articles from the journal *Science*), dDTM is superior in terms of negative log likelihood on streaming data, such as news articles, blogs, and tweets, where topics change with a much higher pace.

Second Experiment: In order to prove the effectiveness of dDTM in computing topic chains, we conduct an experiment to compute the precision and recall of the topic chains. This experiment shows the precision of the topic chains computed by dDTM and endorses the correctness of the skips made on some topics. For this purpose we use the following equations:

$$Precision = \frac{tp}{tp + fp} \tag{4.4}$$

$$Recall = \frac{tp}{tp + fn} \tag{4.5}$$

where tp, fp, and fn are the number of true positives, false positives, and false

Dataset	Precision	Recall
Signal Media	100%	100%
NFCW	89.74%	100%

Table 4.2. Performance of dDTM in computing coherent dynamic topics.

negatives, respectively.

Since manually labeling 900 topics of Signal Media dataset and 60 topics of NFCWorld dataset for computing precision and recall is prohibitive, we used statistical sampling for extracting a representative sub-sample of topics from the original datasets. By using the z-test we learned that for a confidence level of 95% and an error margin of +/- 9.24% (i.e., from 85.76% to 100% of our population) we will need 100 samples from the Signal Media dataset and 39 samples for the NFCWorld dataset Thus, we labeled the sub-samples manually and computed precision and recall of the topic chains returned by dDTM. The assigned labels were either 'correct fit' or 'unfit' for a given topic chain and time slice. The precision examines whether a sampled topic is a fit for the chain it appeared in, while the recall examines whether there were other topics in the same time slice of the sampled topic that were missed although they were a fit for that chain. The results of this experiment, presented in Table 4.2, endorse the effectiveness of dDTM in computing coherent and accurate dynamic topics. Experimental results showed that dDTM performs better in the case of the bigger Signal Media dataset with longer documents than with the smaller dataset of at most 140 characters long tweets. Such result was expected, since modeling topics from small datasets of short documents always brings noise due also to the lack of context provided by the limited textual windows. By looking at the recall measures on the two datasets, we could conclude that the discrete evolution of topics and the skips made by dDTM over some time slices are justified. Moreover, the very high precision measures indicate highly coherent topic chains.

Third Experiment: Figure 4.3 presents an example of a dynamic topic computed using dDTM over the Signal Media dataset The figure shows a discrete evolution of the topic over time. We can self-interpret the topic as *health care and treatment*. It is present over 15 time slices out of the 30 days (i.e., 30 possible time slices). The date on which each topic appears is also indicated above each topic.

We could observe that although our model does not evolve linearly over time and the topics are not present in all time slices, the computed dynamic topics

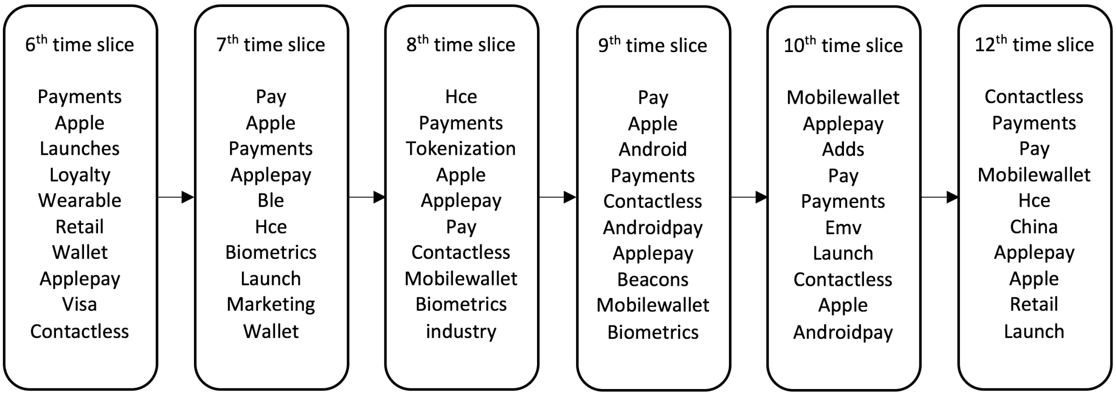


Figure 4.4. A dynamic topic about Apple pay service extracted from the NFCWorld dataset

appear to be coherent.

We repeated the qualitative analysis with the NFCWorld dataset. The advantage of using the NFCWorld dataset is that, although small, it is spread over a longer period of time, thus it is more suitable for observing topical trends that change over time.

Now using the NFCWorld dataset, we demonstrate an example of a dynamic topic which correlates with a real-world phenomenon. Figure 4.4 illustrates the dynamic topic. In our analysis, we could observe that this topic, returned by dDTM, is related to the *Apple pay service*. In Figure 4.5 we show how dDTM could immediately identify this emerging topic which was not present in the first few time slices. For this topic, we visualized the behavior of the term “Applepay” over different time slices in terms of its probability of being representative of the topic. Note that each of the twelve time slices indicated on the x axis represent a duration of three months starting from April 15, 2013. As we can observe from Figures 4.4 and 4.5, this topic was not present over the first five time slices until October 2014. By looking at the Wikipedia page about Apple pay, we could verify that this service was launched by Apple in the United States in October 2014 which corresponds with the 6th time slice on the x axis. Subsequent to the launch of the service there was a peak in the probability of the corresponding term. This example clearly shows that when the service did not exist, the corresponding topic was not extracted by dDTM. However, as soon as the service was launched, dDTM could immediately identify it. We could notice a second peak on the 12th time slice which probably corresponds to the launch of the Apple pay in China and according to the Wikipedia page it took place in February 2016. In summary, we showed an example of how dDTM can immediately identify and track dynamic

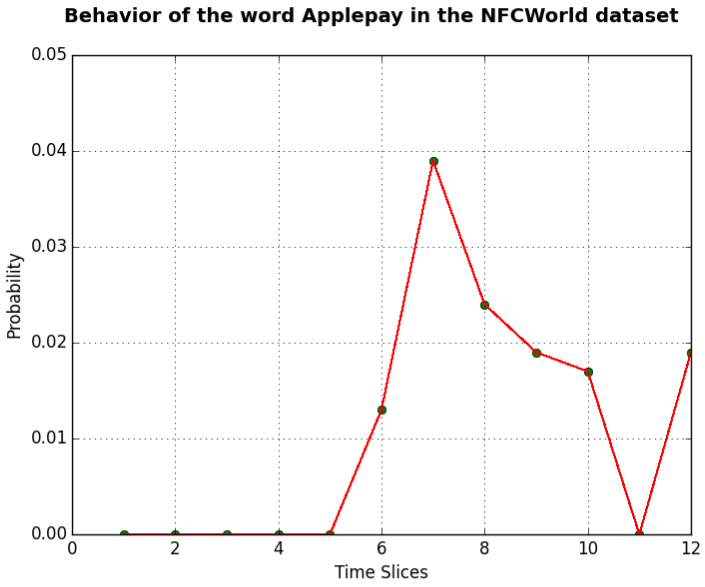


Figure 4.5. The behavior of the term “Applepay” in the topic Apple pay service. Each point in the time axis correponds with a three months duration starting from April 15, 2013 extracted by dDTM. Time slices 0, 6, 12 correspond to Apr ’13, Oct ’14, and Apr ’16 respectively.

topics over discrete time slices. If a market analyst, or a business owner, or yet a politician, had access to such insights, he/she could take advantage of these insights to adjust current and future strategies accordingly.

4.5 Conclusions

In this chapter, we proposed a novel model named dDTM for computing discrete dynamic topics over time. We tested our method on two different datasets, namely, the `Signal Media` and the `NFCWorld` datasets. dDTM is based on LDA and HMM as well as the assumption of independence of topics over time. This assumption applies to datasets made of tweets or daily news, where topics change from one time slice to another at a very high pace. Experimental results on such data demonstrated that our method is superior to DTM. Additionally, through qualitative examples we showed that our method can immediately detect emerging topics, while DTM always responds to changes in topics with a delay and is not able to model discrete topics over time.

There are several interesting directions for future work. We would like to apply dDTM for analyzing trending topics and compare our method against other methodologies developed in this domain. A comparison of different methods for computing HMM state probabilities such as Gaussian mixture models could be another interesting analysis. Furthermore, the time slice length allocation is another challenge for daily conversations which could be an extension of this study. Finally, we plan to conduct a more in-depth analysis of dDTM which would include analysis of the topic chains' quality with respect to human judgment.

Chapter 5

CATS: Customizable Abstractive Topical Summarization

5.1 Introduction

In the two previous chapters we presented models that can extract topics and track their evolution over time, as a means for presenting topical summaries of one's conversations or even other textual datasets over time. These models were extracting words that together represented a semantic concept. We showed that using such summaries can benefit users recall their past. However, due to their brevity the capacity of these summaries for conveying detailed context maybe limited. In the meanwhile personal assistants such as Apple Siri or the Google Assistant are increasingly using natural language as means of communication with humans which does not have the limitations of topic summaries discussed above. Therefore, in this chapter we present a summarization model that can generate textual summaries in the form of natural language. Our proposed model is the first summarization system that brings customization into language generation. In this chapter, we will elaborate on the details of this model, however, we first introduce automatic summarization and its applications.

Automatic document summarization is defined as producing a shorter, yet semantically highly related, version of a source document. Solutions to this task are typically classified into two categories: extractive summarization and abstractive summarization. We explained the differences of these two approaches to summarization in Section 2.2.2.

The majority of research on text summarization thus far has been focused on extractive summarization[19, 98, 128], due its simplicity compared to abstractive methods.

In the field of information retrieval, search engines are increasingly presenting summaries, mash-ups and digests of relevant documents in the form of natural language answers to user queries. Automatic summarization lends itself for key use cases in mobile search and scenarios involving communication with search engines via voice. Previous research shows that merely reading out the textual output of a search engine result page is an insufficient interaction paradigm [43, 113]. Furthermore, the underlying components of a spoken conversational search system (where communication between user and system is mediated verbally through voice) will need to operate differently from a traditional IR system [43, 111, 132]. A recent user study [134] on conversational search [9] has observed the importance of document summarization when presenting results of users' spoken search queries. The ideal voice-based assistant would summarize the key points of particular relevance for a certain searcher. This chapter presents a novel abstractive summarization framework as a first step towards this vision.

Aside from the previously discussed importance of document summarization in the context of modern search engines, its applications go further. In the news domain, summarization is extensively used to concisely describe the gist of news articles and news events [120, 133]. In workplace environments where meetings are held frequently, automatic summarization functionality, together with automatic transcription, can support the minute-taking process [122]. In such scenarios, the use of a summarization system may increase efficiency in the workplace environment. In the medical domain, doctors could benefit from document summarization in order to quickly catch up with the medical history of a newly assigned patient by only focusing on the short summaries provided by a summarization system [52].

In this chapter, we introduce CATS, a novel abstractive summarization sequence-to-sequence model, which is not only capable of summarizing text documents with an improved performance as compared with the state of the art, but also allows to selectively focus on a range of desired topics of interest when generating summaries. Our experiments corroborate that our model can selectively add or remove certain topics from the summary. Furthermore, our experimental results on two publicly available datasets indicate that the proposed neural sequence-to-sequence model exhibits a high performance in terms of ROUGE scores and can effectively outperform state-of-the-art baselines.

The main contributions of our research presented in this chapter are:

- We introduce a novel neural sequence-to-sequence model based on an encoder-decoder architecture that outperforms the state-of-the-art baselines in the task of abstractive summarization on two benchmark datasets.

- We show for the first time how the attention mechanism [12] may be used for simultaneously learning important topics as well as recognizing those parts of the encoder output that are vital to be focused on.

- We show a method for automatically summarizing meeting transcripts using a neural architecture, despite the very small size of our meetings dataset.

5.2 Proposed Model: CATS

5.2.1 Model Overview

Our abstractive summarization scheme CATS is a neural sequence-to-sequence model based on the attention encoder-decoder architecture [97], which is also one of our baselines. Additionally, we incorporate the concept of pointer networks [141] into our model, which enables copying words from the encoder output while also being able to generate words from a fixed vocabulary. Furthermore, we introduce a novel attention mechanism controlled by an unsupervised topic model. This ameliorates attention by way of focusing not only on those words which it learns as important to produce a summary in the standard attention mechanism, but also by learning the topically important words in a certain context. We refer to this mechanism as topical attention. Over the encoder-decoder training steps, the model parameters adapt in a way to learn the topics of each document. During testing, when the model decoder generates summaries of test documents, it therefore no longer requires the input information from the topic model, as it appears to have learned a generalized pattern of the word weights under each topic.

We depict our model in Figure 5.1. In the following we describe the various components of our model.

5.2.2 Encoder & Decoder

The tokens of a document (i.e., extracted by a document tokenizer) are given one-by-one as input to the encoder layer. Our encoder is a single-layer Bi-directional Long Short Term Memory (BiLSTM) network [56]. The network outputs a se-

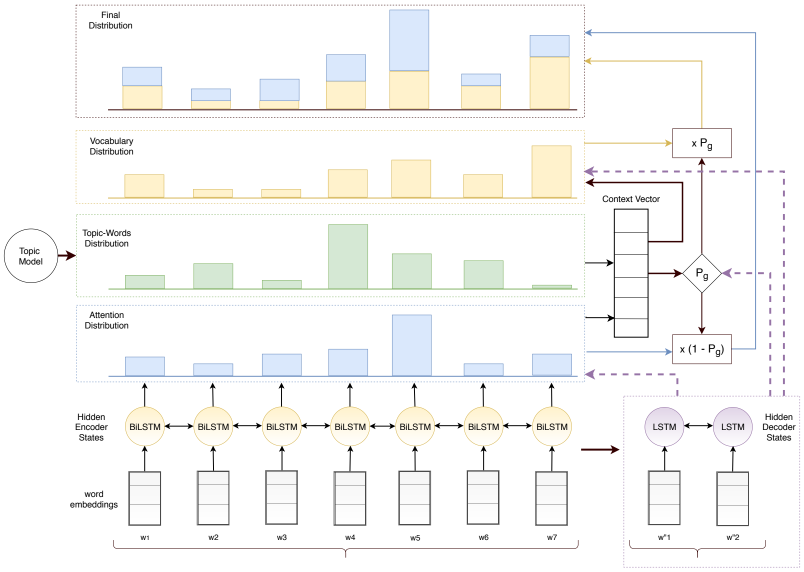


Figure 5.1. The architecture of our proposed model

quence of encoder hidden states h_i, each state being a concatenation of forward and backward hidden states, as in [12].

At each decoding time step t, the decoder receives as input x_t the word embedding of the previous word (while training, this is the previous word of the reference summary and at test time it is the previous word output by the decoder) and computes a decoder state s_t. Our decoder is a single-layer Long Short Term Memory (LSTM) network [57].

5.2.3 Topical Attention

We propose the topical attention distribution a^t to be calculated as a combination of the usual attention weights as in [12] (depicted in Figure 2.2) and a "topical word vector" derived from a topic model. We use LDA [34] as the topic model of choice. Besides the experimentally shown robust performance [34], an important reason for selecting LDA over other topic models is that words under this model are always assigned probabilities between 0 and 1 and the sum of the probability scores of all words in each topic is 1. This facilitates the fusion of these scores with attention weights, which are then fed to a softmax function without the need for additional normalization steps.

In order to compute the topical attention weights, after training an LDA model using the training data, we map the target summary corresponding to each document to its LDA space. This gives us the strength of each topic in each each target summary. Furthermore, since for each topic we also have the probability scores of each word in a fixed vocabulary $\mathcal{V}$, for a given document d we could calculate a *topical word vector* τ^d of dimension $|\mathcal{V}|$ considering all the words in that document, such that:

$$\tau^d = \sum_i P(\text{topic}_i|d) \cdot \tilde{\mathbf{w}}_i \tag{5.1}$$

where $P(\text{topic}_i|d)$ is the probability of each LDA topic being present in the target summary, and $\tilde{\mathbf{w}}_i$ is the $|\mathcal{V}|$-dimensional vector of probabilities $\tilde{w}_j = P(\text{word}_j|\text{topic}_i)$ of all words in vocabulary $\mathcal{V}$ under topic$_i$.

Then, for an input sequence of length K, we compute the final attention vector $a^t \in \mathbb{R}^K$ at decoding step t as:

$$e^t_k = v^T \tanh(W_h h_k + W_s s_t + b_{\text{attn}}) \tag{5.2}$$

$$a^t = f(e^t, \tau^d) \tag{5.3}$$

where $e^t \in \mathbb{R}^K$ is a precursor attention vector, $h_k \in \mathbb{R}^n$ represents the k-th encoder hidden state and $s_t \in \mathbb{R}^l$ the decoder state at decoding step t, while $v \in \mathbb{R}^m$, $W_h \in \mathbb{R}^{m\times n}$, $W_s \in \mathbb{R}^{m\times l}$, $b_{\text{attn}} \in \mathbb{R}^m$ are learnable parameters. Function f combines the topical word vector with the precursor attention vector, and multiple options could be considered, including feed-forward neural networks. In practice, we found that a well-performing choice of f yields the following distribution over the input sequence:

$$a^t = \frac{\text{softmax}(e^t) + \text{softmax}(\tilde{\tau}^d)}{2} \tag{5.4}$$

where $\tilde{\tau}^d \in \mathbb{R}^K$ denotes the "reduced" topical word vector which is formed by selecting the K components of $\tau^d \in \mathbb{R}^{|\mathcal{V}|}$ corresponding to the K words of the input sequence.

The attention distribution can be viewed as a probability distribution over the words from the source document, which tells the decoder where to look to produce the next word. Subsequently, the attention distribution is used to produce a weighted sum of the encoder hidden states, known as the context vector $h^*_t \in \mathbb{R}^n$, as follows:

$$h^*_t = \sum_k a^t_k \cdot h_k \tag{5.5}$$

The context vector, which can be seen as a fixed-sized representation of what has been read by the encoder at this step, is concatenated with the decoder state

s_t and the result is linearly transformed and passed through a softmax function to produce the final output distribution $P_{\mathcal{V}}(w)$ over all words w in vocabulary $\mathcal{V}$:

$$P_{\mathcal{V}} = \text{softmax}(V[s_t, h_t^*] + b) \tag{5.6}$$

where $V \in \mathbb{R}^{|\mathcal{V}| \times (n+l)}$ and $b \in \mathbb{R}^{|\mathcal{V}|}$ are learnable parameters.

5.2.4 Pointer Generator

The idea behind the pointer generator is to circumvent the limitations of pure abstraction when it comes to factual content such as names, dates of events, statistics and other content that requires copying from the source document to produce a correct summary. The basic encoder-decoder architecture presented in Figure 2.2 often makes mistakes with people's names or other factual content while generating a summary. As a response, pointer networks [141] where introduced in the machine translation domain. In this thesis we utilize the concept of pointer generators in order to give our model the flexibility of choosing between generating a word from a fixed vocabulary or copying it directly from source when needed.

We define p_g as a generation probability such that $p_g \in [0, 1]$. We calculate p_g for time step t from the context vector h_t^*, the decoder state s_t and the decoder input x_t as:

$$p_g = \sigma(w_{h^*}^T h_t^* + w_s^T s_t + w_x^T x_t + b_{pt}) \tag{5.7}$$

where vectors w_{h^*}, w_s, w_x, and scalar value b_{pt} are learnable parameters and σ is a sigmoid function.

Subsequently, p_g is used to linearly interpolate between copying a word from the source (specifically, to copy from the source document we sample over the input words using the attention distribution) and generating it from the fixed vocabulary using $P_{\mathcal{V}}$.

For each document, we define the union of the fixed vocabulary $\mathcal{V}$ and all words appearing in the source document as the "extended vocabulary". Using the linear interpolation described above, the probability distribution over the extended vocabulary is:

$$P(w) = p_g P_{\mathcal{V}}(w) + (1 - p_g) \sum_{\forall i : w_i = w} a_i^t \tag{5.8}$$

In Equation 5.8, we note that if a word w would be an OOV word, then $P_{\mathcal{V}}(w)$ would be equal to zero. Analogously, if w does not appear in the source document, then $\sum_{\forall i:w_i=w} a_i^t$ would be equal to zero. In expectation, the most likely words under this new distribution are the ones that both receive a high likelihood under the output distribution of the decoder, as well as much attention by the attention module. Words with a high likelihood under the initial output distribution, which however receive little to no attention, will be generated with a reduced probability, while words receiving much attention, even if they receive a low likelihood by the decoder or do not even exist in the vocabulary $\mathcal{V}$, will be generated with an increased probability.

Therefore, by being able to switch between OOV words and the words from the vocabulary, the pointer generator model mitigates the problem of factual errors or the lack of sufficient vocabulary in the output summary.

5.2.5 Coverage Mechanism

As mentioned in Section 2.2.2, the coverage mechanism [135] is a method for keeping track of the level of attention given to each word at all time steps. In other words, by summing the attention at all previous steps, the model keeps track of how much coverage each encoding has already received.

This mechanism alleviates the repetition problem, which is a very common issue in recurrent neural networks with attention. It originated from the machine translation domain [135], when trying to address the problem of translating a word multiple times while not translating other words at all. In text summarization, the notion of coverage has been used with a small modification [120, 149]. Xu et al. [149] introduce this idea for the first time in the context of image captioning. Unlike previous work, which used a GRU to update the coverage vector at each step [135], they [149] summed up the attention distributions to obtain the coverage vector. They named this approach “distraction”, as it prevents over-attending certain words.

We follow the same approach and define the *coverage vector* $c^t \in \mathbb{R}^K$ simply as the sum of attention vectors at all previous decoding steps:

$$c^t = \sum_{i=0}^{t-1} a^i \tag{5.9}$$

First, the coverage vector is taken into account when calculating the attention vector by adding an extra term and modifying (5.2) as follows:

$$e_k^t = v^T \tanh(W_h h_k + W_s s_t + c_k^t \cdot w_c + b_{\text{attn}}) \tag{5.10}$$

where $w_c \in \mathbb{R}^m$ is a learnable parameter vector of the same length as v.

Second, following [120], we use the coverage vector to introduce an additional loss term, which is added to the original negative log-likelihood loss after being weighted by hyperparameter λ, to produce the following total loss at decoding step t :

$$\mathscr{L}_t = -\log P(w_t) + \lambda \sum_{i=0}^{k} \min(a_i^t, c_i^t) \tag{5.11}$$

This additional loss term encourages the attention module to redistribute attention weights by placing low weights to input words which have already received much attention throughout previous decoding steps. The overall loss for the entire output sequence of length T is:

$$\mathscr{L} = \frac{1}{T} \sum_{t=0}^{T} \mathscr{L}_t \tag{5.12}$$

5.2.6 Decoding

In order to generate the output summaries, at each step, the decoder outputs a probability distribution over the target vocabulary. To get the output word at this step, there are several possibilities, such as sampling from the distribution. However, the most popular method is referred to as *beam search*. The principal idea behind beam search is to choose the top N most likely target words and then tentatively feed each into the next decoder input. Thus, at each time-step t, the decoder gets N different possible inputs and for each of those computes the output distribution (i.e., distribution of the following word). It only considers the top N most likely target words for each of these N different inputs, and among these N^2 possibilities, keeps only the top N and rejects the rest. This process continues so as to ensure that each target word gets a fair chance at generating the summary.

During evaluation of the model using the test data, contrary to training, we do not need to provide the model with any topical information from our trained LDA topic model. Still, we found that the performance benefit of topical attention persists. We believe that the reason may be that during training, the model parameters learn to best take advantage of the provided topical attention distribution, implicitly learning patterns of topic-words weights. These patterns affect predictions even in the test phase.

5.3 Approach for Meeting Summarization

5.3.1 Challenges of Meeting Summarization

Most summarization research thus far has focused on news documents for reasons of data availability. However, in addition to the small size of the existing meeting datasets which make them unusable for most neural models, there are other issues that make meeting summarization additionally challenging: (1) Most news articles are first-person narratives about a single event. Meetings, on the other hand, have a very different structure involving a dialogue between two or more parties. (2) Meetings are composed of spoken utterances between people, whereas their summaries and minutes are usually formulated from a third-person point of view by the human scribe. Therefore, meeting summarization also requires change of structure from dialogue to a third-person narrative summarizing events. (3) Meetings can touch on multiple topics and are not restricted in terms of topical coherence. (4) Meeting transcripts include broken sentences, colloquial expressions, false starts and flawed grammar, all of which virtually never occur in carefully curated news articles. As an example, here is an excerpt from a meeting in one of the meeting datasets used in this chapter which contains most of these flaws:
"mm-hmm . so sh . i 'm a bit confused about uh what 's the difference between the functional design and conceptual design ? uh i is it just uh more detail , uh as i understand it ? right . how how it will be done . so whe where do we identify the components of our uh product ? "

These issues are a common challenge of meeting transcripts and are noticeable in every meeting in the meeting datasets used in this thesis.

5.3.2 A Transfer-Learning Method for Meeting Summarization

Transfer-learning is referred to machine learning research that take advantage of datasets in a certain domain to solve problems in a different domain, often due to limitations such as lack of sizable datasets.

In this section, we introduce a transfer-learning approach. We train all neural sequence-to-sequence models initially on a large news dataset presented in Section 5.4.1. Our transfer-learning approach is based on fine-tuning and adapting model parameters to the new task of meeting summarization.

As a result, after we pre-train the models on the large-scale news dataset, we fine-tune them as follows: We feed the models with the meeting training dataset described in Section 5.4.2. We use a small learning rate to tune all parameters from their original settings to minimize the loss on the new task. Additionally, we change the encoder sizes for all neural models from 200 to 400 and the decoder sizes from 50 to 100. Moreover, we increase the minimum number of tokens generated from 35 to 65 to account for the greater length of meeting transcripts and corresponding summaries.

Fine-tuning adapts the models' parameters to make them more discriminative for the new task, and the low learning rate is an indirect mechanism to preserve some of the representational structure learned in the news summarization task. Moreover, we expose the models to the meeting training data for 50 epochs on the meeting training set with a batch size of 16.

We tune all neural models presented in this chapter in the same way.

5.4 Evaluation

In this section we introduce our experimental setting, including details of our datasets, baseline models, and evaluation metrics. Finally, we present the experimental results of this work.

5.4.1 Datasets

The CNN/DailyMail dataset

We use the CNN/DailyMail dataset [66, 97], which contains news articles from the *CNN* and *Daily Mail* websites. This dataset is large, and due to its size as well as the length of its documents, has been used as a benchmark collection in many papers focused on text summarization. However, there exists two different versions of it, which can make direct comparison between the numbers reported in different papers misleading. The difference between the two versions is that one is anonymized (i.e., named entities are replaced with unique but anonymous identifiers), while the other one contains the original text of the news articles. The experiments reported in this chapter are based on the non-anonymized version of the dataset, containing 287,226 pairs of training articles and reference summaries, 13,368 validation pairs, and 11,490 test pairs. On average, each document in the dataset contains 781 tokens paired with multi-sentence summaries (56 tokens spread over 3.75 sentences).

Similar to [97, 120], we use a range of pre-processing scripts to prepare the data. This includes the use of the *Stanford CoreNLP* tokenizer to break down documents into tokens. For greater transparency and reproducibility of our results, we make all pre-processing scripts available together with our code base.

Table 5.1. Statistics of our meeting datasets.

	minutes	ave. #tokens per doc.	ave. #tokens per summary	minimum #tokens	median #tokens	maximum #tokens	#meetings
AMI	4868	5843	283	892	5998	11552	142
ICSI	3513	13080	449	2785	12605	22573	61
ADSC	NA	446	118	152	482	1383	45

5.4.2 Meetings Dataset

For our empirical investigation of meeting transcripts summarization, we compile the available meeting datasets that have been used in previous work on meeting summarization.

For this purpose, we gathered data from the well-known AMI dataset[1] as well as the ICSI dataset[2] which are the only exisiting datasets of real-world meetings. AMI contains two categories of meetings involving between two to four participants. The first collection consists of freestyle meetings where the participants can decide on the topics of discussions, and targeted ones about designing technology products (e.g., a remote control). The ICSI dataset on the other hand, contains weekly group meetings of academic groups of three to ten participants. Both AMI and ICSI are face-to-face meetings that were initially audio recorded and then later transcribed. We randomly divide the AMI and ICSI datasets in a 50-50 split to construct a training set as well as a test set. As a result, we end up with 101 real-world meetings as our test set and the remaining ones as the training set.

In order to increase the size of our training set we also add the Argumentative Dialogue Summary Corpus (ADSC) dataset[3] to our training set. The ADSC is composed of controversial online conversations on topics of societal and political relevance such as gun control, gay marriage, the death penalty and abortion. Table 5.1 presents detailed statistics on all three datasets.

Finally, we also use the CNN/DailyMail corpus (presented in Section 5.4.1) for pre-training all baselines and our proposed models. That is due to the small size of the currently available meeting transcripts datasets which make training of neural-based models virtually impossible.

5.4.3 Baseline Models

We empirically compare CATS with several abstractive baselines as follows:

- *Attention-based encoder-decoder* [97].

[1] http://groups.inf.ed.ac.uk/ami/download/

[2] http://groups.inf.ed.ac.uk/ami/icsi/download/

[3] https://nlds.soe.ucsc.edu/node/30

- *PGN and PGN+Coverage* [120].
- *RL with Intra-Attention* [101].
- *BottomUpSum* [53].
- *InformationSelection* [87].
- *ML+RL ROUGE+Novel, with LM* [79].
- *UnifiedAbsExt* [68].
- *RNN-EXT + ABS + RL + Rerank* [37].

5.4.4 Evaluation Metrics

We follow the accepted practice [51, 98, 120] of evaluating our proposed model against the baseline methods in terms of $F_1\,ROUGE\,1$, $F_1\,ROUGE\,2$, and $F_1\,ROUGE\,L$ scores which are standard metrics for evaluating text summarization systems. We use the official Perl-based implementation of ROUGE [89], following common practice.

$F_1\,ROUGE\,1$ is defined as the number of uni-grams (i.e., tokens) that overlap between the summaries generated by a system and the reference summaries. Likewise, $F_1\,ROUGE\,2$ is referred to the number of bi-grams that overlap between the summaries generated by a system and the reference summaries. Finally, $F_1\,ROUGE\,L$ is defined as the number of longest common sub-sequences (i.e., number of longest co-occurring n-grams in a sequence).

5.4.5 Experimental Results

We specify our model parameters as follows: the hidden state dimension of RNNs is set to 256, the embedding dimension of the word embeddings is set to 128, and the mini-batch size is set to 16. Furthermore, the maximum number of encoder steps is set to 400 and the maximum number of decoder steps is set to 100. In decoding mode (i.e., generating summaries on the test data) the beam search size is 4 and the minimum decoder size which determines the minimum length of a generated summary is set to 35. Finally, the size of the vocabulary that the models use is set to 50,000 tokens.

To train a topic model we run LDA over the training data. LDA returns K lists of keywords representing the latent topics discussed in the collection. Since the actual number of underlying topics (K) is an unknown variable in the LDA model, it is important to estimate it. For this purpose, similar to the method proposed in [18, 58], we went through a model selection process. It involves keeping the LDA parameters (commonly known as α and η) fixed, while assigning several values to K and running the LDA model for each value. We picked the model

Table 5.2. Results of a comparison between our proposed models against the baselines in terms of F_1 ROUGE metrics on the CNN/Dailymail dataset. Statistical significance test was done with a confidence of 95%. '*' means that results are based on the anonymized version of the dataset and not strictly comparable to our results.

Models	ROUGE 1 (%)	ROUGE 2 (%)	ROUGE L (%)
CATS (Ours)	38.01	16.35	34.87
CATS+coverage (Ours)	**41.73**	**18.64**	38.17
LEAD-3 Baseline	40.34	17.70	36.57
Attn. Enc-Dec [97]	35.46	13.30	32.65
PGN [120]	36.44	15.66	33.42
PGN+coverage [120]	39.53	17.28	36.38
RL with Intra-Attention [101] '*'	41.16	15.75	**39.08**
BottomUpSum [53]	41.22	**18.68**	38.34
InformationSelection [87]	41.54	18.18	36.47
ML+RL ROUGE+Novel, with LM [79]	40.19	17.38	37.52
UnifiedAbsExt [68]	40.68	17.97	37.13

that minimizes $\log P(W|K)$, where W contains all the words in the vocabulary. This process is repeated until we have an optimal number of topics. The training of each LDA model takes nearly a day, so we could only repeat it for a limited number of K values. In particular, we trained the LDA model with values K ranging from 50 up to 500 with an increment of 50, and the optimal value on the CNN/Dailymail dataset was found to be 100.

The experiments on all neural models reported in this chapter were conducted using a Titan V GPU with 18GB of RAM per node.

Based on the setup described above, we now present our experiments for evaluating our model:

Experiment comparing all models in terms of ROUGE

We first compare our proposed models against all baselines in terms of the F_1 ROUGE metrics presented in Section 5.4.4. The results of this comparison are given in Table 5.2.

As we observe in Table 5.2, our model with coverage outperforms all other models in terms of ROUGE 1. In order to verify the significance of the difference we conduct a statistical significance test based on the bootstrap re-sampling technique using the official ROUGE package [89]. In the case of ROUGE 2 we achieve state-of-the-art performance in a tie with the ’BottomUpSum’ approach of [53]. In the case of ROUGE L, [101] reports the highest performance; however, this is due to their model loss function optimizing directly on the evaluation metric ROUGE L instead of the summarization loss. In fact, [68] reports an experiment that shows summaries generated by the [101] method achieve poorest readability scores as compared with a number of models including PGN and their own UnifiedAbsExt model, a finding which we also confirmed by comparing them with the output of our model (see Section 5.4.5). We note that we did not include the method of [36] in our comparison, due to the fact that unlike most papers that use preprocessing scripts of [120] for the non-anonymized version of the dataset, they use different scripts. The effect of this difference on their LEAD-3 baseline remains unclear as they do not report it. Thus, their results may not be necessarily comparable with ours.

Human Evaluation of Summaries

We conduct a human evaluation in order to assess the quality of summaries produced by CATS+coverage in comparison with that of PGN+coverage [120] and summaries of RL with Intra-Attention [101] provided by them, in terms of informativeness and readability of 50 randomly chosen summaries by the three models. By comparing the output produced by the three models, the three human assessors[4] assigned scores ranging from 1 to 5 to each summary, while blinded to the identity of the models. The average overall scores of each model are shown in Table 5.3.

Table 5.3. Human evaluation comparing quality of summaries on a 1-5 scale using three evaluator.

	Readability	Informativeness
CATS	**4.1**	**3.9**
PGN	3.5	3.3
RL+Intra-Attention	2.6	2.9

We observe that the summaries generated by our model are judged to be more readable and more informative.

[4] None of the assessors are affiliated with this paper.

Human Evaluation of Customizing Summaries

In this section, we report a human evaluation of CATS's capability to include only certain topics in a summary and exclude others. As mentioned earlier, CATS is the first neural abstractive summarization model that allows its users to selectively include or exclude latent topics from their output summaries. In order to demonstrate this feature, we remove a few topics from the output of the topic model, fine-tune the trained summarization model for a few additional training steps and analyze the effect. Our expectation is that the focus of certain output summaries which should usually contain those topics will change, while naturally the ROUGE values will decrease. For this experiment, we chose two topics and removed them from the summaries one at a time. The first topic is related to *health-care* and its top five keywords are "dr", "medical", "patients", "health", and "care". The second topic is related to *police arrests and charges* with its top five words being "charges", "court", "arrested", "allegedly", and "jailed". We randomly selected a total of 50 test documents that originally contained either of the above-mentioned topics. In order to do so we used the LDA model described in the beginning of Section 5.4.5. Using the LDA rankings of topics of source documents, we randomly chose 50 that contained either-mentioned topics and those topics were not their sole or primary focus but in the second rank. Three human judges evaluated whether the summaries generated by CATS with restricted topics showed exclusion or reduction of those topics or there was no major difference. They were instructed to look for existence of the top 20 words of each topic in particular, except for cases that one of these words is a part of a name (e.g. American Health Center). For each document, we take the majority vote of the human assessors as the final decision. The results of this experiment show that in 44 documents the topics were excluded, in four documents the topics were reduced and in two documents the majority vote showed no major difference.

Table 5.4 shows an example summary produced by CATS that was restricted not to include the *health-care* topic, next to a summary produced by CATS with no topic restriction as well as the corresponding reference summary. We observe that the focus of the summary is altered such that it focuses on the crime-related aspects rather than health-care in order to avoid using words such as "hospital", "patients" and "medicine".

Analysis of Repetition in Output Summaries

In this experiment we analyze the quality of the output summaries produced by our models and those produced by PGN and PGN+coverage in terms of repeti-

Table 5.4. Comparison of a CATS generated summary next to a summary restricted with health-care topic and the human-written reference summary.

CATS topic-restricted	***CATS***	***Reference***
victorino chua , 49 , denies murdering tracey arden , 44 , arnold lancaster , 71 and derek weaver , 83 , and deliberately poisoning 18 others between 2011 and 2012 . chua has pleaded not guilty to 36 charges in all , including three alleged murders , one count of grievous bodily harm with intent , 23 counts of attempted grievous bodily harm with intent , eight counts of attempting to cause a poison to be administered and one count of administering a poison .	victorino chua , 49 , has given evidence for the first time and denied he tampered with saline bags and ampoules at stepping hill hospital in stockport . a nurse today told a jury he did not murder three hospital patients and poison almost 20 more at stepping hill hospital in stockport in order to kill and injure people he was caring for . chua denies murdering patients tracey arden , 44 , arnold lancaster , 71 and derek weaver , 83 , and deliberately poisoning 18 others between 2011 and 2012 .	victorino chua , 49 , denies murdering patients at stockport hospital in 2011 . filipino nurse also accused of poisoning 18 more at stepping hill hospital . denies injecting insulin and other poisons into bags of medicine on ward .

tion of text. A common issue with attention-based encoder-decoder architectures is the tendency to repeat an already generated sequence. In text summarization this results in summaries containing repeated sentences or phrases. As described in Section 2.2.2, the coverage mechanism is used to reduce this undesirable effect.

Here we compare our two models, CATS and CATS+coverage, to PGN and PGN+coverage in terms of n-grams repetition with n ranging from 1 to 6. For this purpose we train all four models with exact same parameters whenever applicable. The upshot of this experiment is reported in Figure 5.2. The scores reported in the figure are normalized average repetition scores over all output summary documents in the test set of the CNN/Dailymail dataset. We compute the scores by calculating the average of per-document n-gram repetition score $S_{\text{rep,doc}}$ over all test output documents, which is defined as $S_{\text{rep,doc}} = \frac{\#\text{duplicate n-grams}}{\#\text{all n-grams}}$.

We observe that our models demonstrate lower repetition of text in their output summaries compared with both PGN and PGN+coverage, which is confirmed by manual inspection of the output. This trend is consistent on all the tested n-grams.

We believe that the reason behind this phenomenon is that our model tends to focus not only on the few words in the input sequence which are assigned high attention weights, but also on other words which are topically connected with these words in a certain context. Firstly, this acts as an attention diversification and redistribution mechanism (an effect similar to coverage). Secondly, these topically connected words receive a higher generation probability (through Equations 5.6 and 5.8) and the model is more inclined to paraphrase the input.

The result of this experiment indicates that our topical attention mechanism may be a viable solution to the repetition issue in sequence generation based on encoder-decoder architectures.

Qualitative comparison of output summaries

As a third experiment, we qualitatively compare the performance of our model versus that of the most competitive baseline, PGN, while not using the coverage mechanism in either of the models. Instead of cherry-picking examples, we juxtapose the best and the worst output summaries of our model (in terms of ROUGE) with the corresponding summaries generated by the PGN baseline. The example showing the best output summary of our model in terms of ROUGE is presented in Table 5.5. As we can observe, our model is capable of producing a summary fully explaining the allergy; however, the summary does not cover any statements about the person's weight. Comparing this summary against that of

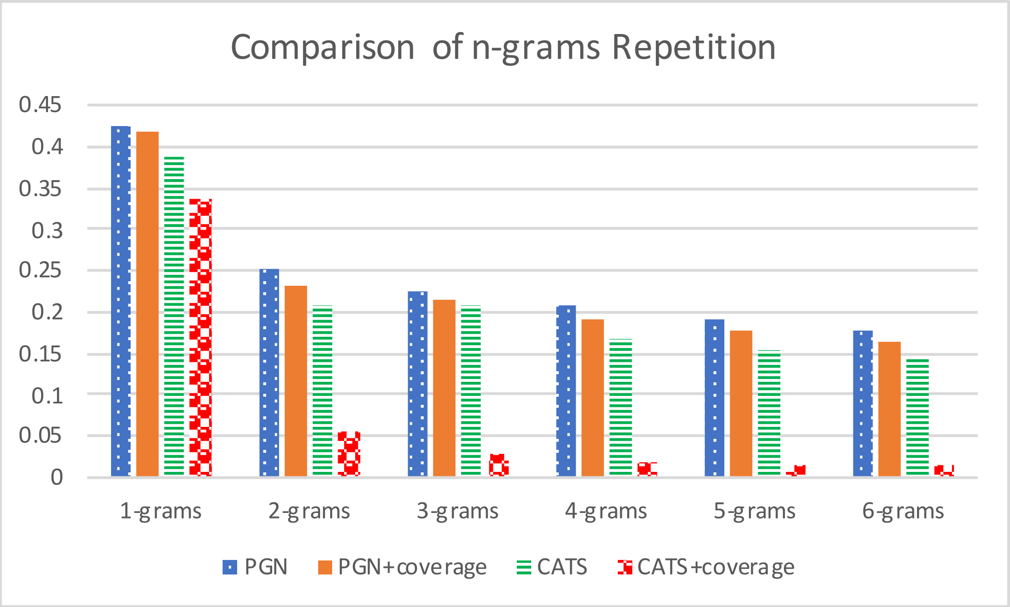


Figure 5.2. Experiment comparing the degree of n-grams repetition in our models versus that of the PGN and PGN+coverage baselines on the CNN/Dailymail test set. Lower numbers show less repetition in the generated summaries.

Table 5.5. Comparison of the best generated summary of our system (i.e., in terms of $F_1\,ROUGE$) and the corresponding summary produced by PGN, next to the human-written reference summary.

PGN	***CATS***	***Reference***
kerrie armitage , from leeds , suffers from the ultra-rare condition aquagenic urticaria .she claims a kiss from her husband can trigger a painful flare-up , due to contact with his saliva .she claims a kiss from her husband can trigger a painful flare-up , due to contact	kerrie armitage , from leeds , suffers from the ultra-rare condition aquagenic urticaria .she was diagnosed two years ago after her skin erupted in agonising blisters .she claims a kiss from her husband peter -lrb- right -rrb- can trigger a painful flare-up .	kerrie armitage , 28 , from leeds , suffers from an allergy to water .external exposure to water - rather than drinking liquid - causes a reaction .condition also means sweating or crying can trigger a painful flare-up .also suffers from exercise-induced anaphylaxis , so has put on weight .

Table 5.6. Comparison of the worst generated summary of CATS (in terms of $F_1 ROUGE$) and the corresponding summary produced by PGN, next to the human-written reference summary.

PGN	***CATS***	***Reference***
walmart 's staunch criticism of a religious freedom law would boost pay for about 500,000 workers well above the federal minimum wage .former minnesota gov. tim pawlenty called on the gop to " be the party of sam 's club "	walmart 's staunch criticism of a religious freedom law in arkansas came after the company said in february it would boost pay for about 500,000 workers well above the federal minimum wage .the company is emerging as a bellwether for shifting public opinion on hot-button political issues .	while republican gov. asa hutchinson was weighing an arkansas religious freedom bill , walmart voiced its opposition .walmart and other high-profile businesses are showing their support for gay and lesbian rights .their stance puts them in conflict with socially conservative republicans , traditionally seen as allies .

the PGN, we observe that our model has clearly generated a better summary, as it includes an additional informative sentence, while the PGN-produced summary contains a repetition and ends with an incomplete sentence[5].

Furthermore, in Table 5.6 we show the worst output summary produced by our model (in terms of ROUGE) next to its corresponding summary from the PGN model. We can see that both our model and PGN go off-topic in the generated summaries. However, PGN also makes a semantic error by perceiving the criticism of a religious freedom law as the cause of a higher pay.

Thus, both in the cases of its best and worst performance in terms of ROUGE, CATS was capable of generating better summaries than the baseline, a trend which is consistent through all examples we have examined.

Experimental Results on Meeting Summarization: We begin by comparing all models in terms of the $F_1 ROUGE$ metrics on meeting summarization. Table 5.7 illustrates the results of this experiment. As a point of reference, we also include the performance of our model without transfer-learning as well as PGN without transfer-learning (named in the table as "CATS No-TL" and "PGN No-TL") in order to demonstrate the importance of the proposed fine-tuning step.

As we can see in the table our model with the transfer-learning significantly outperforms all other models in terms of ROUGE 1 and ROUGE L while it matches PGN and PGN with coverage in terms of ROUGE 2. Our statistical significance test is based on bootstrap re-sampling using the official ROUGE package [89] and confirms that the observed improvement over the baselines in terms of ROUGE 1 and ROUGE L is significant with a confidence of 95%.

The most important finding of this experiment is the comparison of our model and the PGN model against their equivalent versions with no transfer-learning applied. The considerable improvement in performance corroborates that our transfer-learning approach is very effective in building a meeting abstractive summarization system.

We also observe that the best extractive baseline is the Luhn method which is superior or equal to other baselines in terms of ROUGE 1 scores while it falls behind in the other metrics.

As discussed in Section 5.3, challenges of meeting summarization include dialogues need to be summarized from a third-person point of view as well as the use of conversational expression such as "oh", "uhum", etc. Manual inspection of a 20% random sample of produced test summaries did not show even a single such utterance while all produced statements use the desired third-person

[5] The phrase "-lrb right -rrb", where '-lrb' stands for 'left round bracket' and '-rrb' means 'right round bracket', is due to the fact that the husband mentioned in the article, named Peter, appears on the right side on a photograph

Table 5.7. $F_1 ROUGE$ scores on AMI/ICSI test sets.

	ROUGE 1	ROUGE 2	ROUGE L
CATS No-TL	12.13	1.54	11.15
CATS	**30.85**	**8.89**	**28.50**
Attn. Enc-Dec	24.73	5.69	22.09
PGN No-TL	11.37	1.48	10.37
PGN	27.41	8.58	25.88
PGN+coverage	28.55	8.61	26.24
Luhn [90]	28.40	3.94	23.69
TextRank [93]	3.36	0.46	2.89
LexRank [49]	23.46	3.05	20.05
LSA [126]	20.00	2.54	16.37

narrative.

5.5 Conclusions

With the rapid proliferation of natural language interactions with search engines, abstractive summarization is of key importance. In this chapter we present CATS, an abstractive summarization model that makes use of latent topic information in a source document, and is thereby capable of controlling the topics appearing in an output summary of the source document. This can enable search engines to customize generated texts based on user profiles or the context of a query, in order to present content tailored to the user's information needs.

Our experimental results show that our proposed models outperform state-of-the-art pointer-generator baselines in terms of standard evaluation metrics for summarization (i.e ROUGE) on an important benchmark dataset, while producing summaries which are qualitatively better (i.e., contain less repetition and are more readable). In addition to the considerably superior performance, our models also provide the flexibility of allowing a user to control the topical focus of the summaries they generate.

CATS can serve as a foundation for future work in the domain of automatic summarization. Based on the results of our experiments in this chapter, we believe the future work on summarization systems to be exciting, in that a generated summary could be customized to users' needs. We envision three ways of controlling the focus of output summaries using our models: First, as demonstrated in the experiment in Section 5.4.5, certain topics could be disabled in the output of the topic model and be consequently discarded from output summaries. Second, a reference document could be provided to the topic model, its topics could be extracted and subsequently direct the focus of generated summaries. This is useful when a user wants to see summaries/updates primarily or only regarding issues discussed in an existing reference document. Third, content extracted from user profiles (e.g. history of web pages they have read) could be provided to the topic model, their salient themes extracted by the model and then taken into account whenever presenting users with information via voice-activated search or showing summaries in a search result page.

All three directions mentioned above are interesting future work of this research. Furthermore, we plan to extend this work by modifying the proposed models such that hyperparameters of the topic model are learned in the training of the sequence-to-sequence model. We would like to find out whether such training can improve the performance of our models.

Part II

Just-In-Time IR for Personal Assistance

Chapter 6

Just-In-Time IR: Systems that Know When you Forget

6.1 Introduction

In the first part of this thesis we focused on summarization methods for presenting summaries of one's conversations and lifelog event snippets to a user. In the second part of this thesis we present a family of models for just-in-time IR, in order to predict and address users' information needs.

Recently a new class of personal assistants that are capable of addressing users' information needs in a just-in-time IR setting proactively is emerging.

Users' information needs may include timely notifications about a certain context such as location [3, 4, 10], important reminders regarding social interactions with other people, search queries that a person is likely to find useful, other events, etc. Personal assistants can assist people by recommending the right information at just the right time and help them in accomplishing tasks. Because of the ubiquitous nature of mobile personal assistants, they have a broad range of potential capabilities. One of these potential capabilities is to carry out sophisticated tasks for supporting failing memories. Such support of human memory has been thus far limited, merely to setting reminders and calendar events.

In this chapter, we present our work on developing a cutting-edge personal assistant for supporting failing memories in every day social interactions. Specifically, we envision a personal assistant that can anticipate the parts of a past conversation that you are likely to forget, and remind you about them. Our experimental results on a real-world dataset of meetings reveals evidence that developing such systems is feasible and can produce promising results.

Human memory is a critically important cognitive ability that we constantly rely on for carrying out various tasks in our daily lives [63]. However, sometimes, due to the volume and intensity of information that we are exposed to on a typical day, or due to our lack of adequate attention, or yet due to aging, this critical cognitive ability fails to recall important events in our past. In a workplace environment, failing to recall important work related events can result in frustration and disappointment. Discussions may be repeated and the work cycle may be prolonged. This can result in a waste of time, energy and resources, in addition to bringing tension and misunderstandings to the work place.

Aiding human memory [74] for later recall of workplace meetings has been tackled by summarizing, indexing [70], and generating memory cues [71] of digital records of meetings. An interesting research question in this domain is whether it is possible to build a system that can foresee which parts of a conversation you are likely to forget within a fixed time interval, hence setting proactive reminders to assist you with them. Such augmentation of human memory can effectively serve as a solution in preventing failure to recall past events. In this chapter we present experiments that reveal an effective solution to this question.

In recent years, the different fields of lifelogging[60] and personal assistants have emerged in parallel which, if combined, may initiate an effort to build personal assistants that can analyze more data modalities and result in much more powerful personal assistants than exist today. The advent of various wearable data capture devices (e.g. wearable video/audio recorder or biophysical sensors) has also created new opportunities to utilize the collected data for various human-aid applications including the support of failing memories. The availability of smartwatches that are increasingly embedding biometrics sensors and are synchronized with smartphones is also creating the opportunity of developing personal assistants that can carry out augmentation of human memory.

The new paradigm of personal assistants such as Google Now, Microsoft Cortana or Apple's Siri seek to build on the notion of JITIR by 'offering proactive experiences that aim to recommend *the right information at just the right time* and help you get things done, even before you ask' [127]. These personal assistants are increasingly being built into other platforms. As an example, Google Now is built into Google home, and Microsoft Cortana is available on Windows desktop and thus we believe that they would play a significant role in the future.

In this chapter, we focus on another aspect of JITIR for personal assistance by tracking one's conversations with the purpose of predicting the parts of a conversation that one may forget. To the best of our knowledge, this is the first work that aims at augmenting human memory by predicting the parts of a conversation that one is likely to forget We use a real-world dataset of weekly meetings of six groups of people. In this dataset, we recorded real workplace conversations of each group (where a group consisted of two individuals) over the span of an entire month. In addition to recording the audio/video of a meeting, we record biophysical sensor data such as Electro-Dermal Activity (EDA) as well as first-person-view images captured automatically by each individual's wearable camera. By analyzing the recorded data, our proposed system extracts personalized insights for each user. Having access to such insights, a potential user of our system can be helped with remembering the things one is likely to forget, increasing self-awareness, being able to plan the future better, and more.

Thus, the main contributions of our research presented in this chapter are:

- We propose the idea of augmenting human memory by predicting the segments of a conversation that one is most likely to forget.
- We present a model capable of carrying out prediction of parts of a conversation that a person is likely to forget or remember.

6.2 Background

In Chapter 2, we presented the *forgetting curve* by Ebbinghaus. We stated that according to this curve, forgetting is an exponential function of time and a human forgets on average about 77% of the details of what he has learned (for the first time) after just six days. This motivated our goal in augmenting human memory in this chapter to assist one in recalling details of one's past events that one is likely to forget. Moreover, our study is motivated by a memory augmentation tool that we have already developed and deployed in the context of a project[1] for aiding people's memories in their workplace meetings [13, 31]. This system takes as input transcriptions of audio recordings of one's conversations and images taken automatically by one's wearable camera at fixed time intervals. Both media types are time synchronized. The tool then processes the data by extracting topics of the transcribed conversations, detecting and recognizing faces in the images, and connecting the topics with their corresponding images. We refer to a topic connected with its corresponding image as an *event snippet*. Figure 6.1 shows an example of an event snippet (i.e., an image paired with a topic). The rationale behind using topics is that they can effectively summarize long conversations and at the same time backtrack from a topic to its sentences of origin. This gives the user the opportunity to review the exact sentences from which a topic was extracted. Through pressing a button the system will display the next or

[1] http://recall-fet eu/

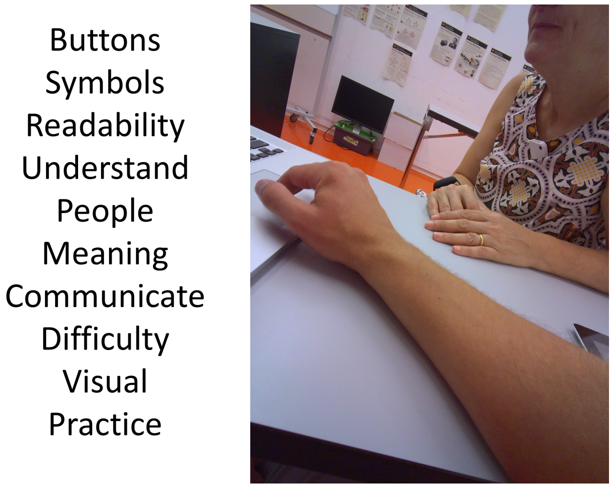


Figure 6.1. Sample output of our memory augmentation tool

previous topic. The system can also produce PDF document outputs of a set of event snippets representing a meeting, which could be easily reviewed to recall the context of the meeting in detail. We showed the efficacy of this system in improving users recall of past social interactions through a user study in Chapter 3.

Knowing which parts of a conversation are non-memorable /memorable can help further developing such system which can show event snippets as reminders, personalized for the memory of each user and customized for the specific parts of a conversation (i.e., content). This scenario motivates the work presented in this chapter. Furthermore, the benefit of this research work is endorsed by other studies [76, 85, 121] which showed for people from different age groups (i.e., ranging from young to old) that replaying the recordings of their lives have significant effect in helping them better recall and remember past events.

6.3 Research Goal

We hypothesize that the content of a conversation and the biophysical sensor recordings such as EDA have an impact on the degree that a person remembers each part of her conversations. If we would be able to show that there are correlations between such signals and memorability, this would mean that it would be possible to design models capable of anticipating the parts of a conversation that one is prone to forget. If so, the upshot could be an advancement in human memory health care. Therefore, by conducting a feasibility study, in this chapter,

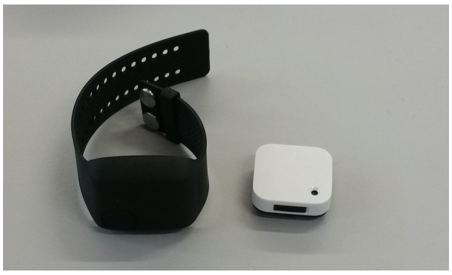


Figure 6.2. Empatica E4 biophysical sensor (left) and Narrative Clip Wearable Camera (right)

we experimentally show the potential of this new research direction. To achieve this goal in a ubiquitous setting, we only use easy-to-use wearable devices that one could practically use in every day life.

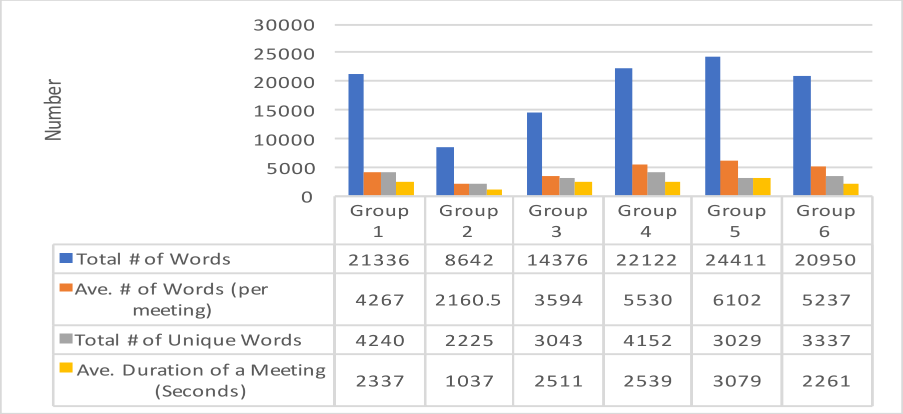
Research in the field of psychology of memory [139] shows that negative experiences are more memorable for people than positive ones. Can this finding be useful for our study? To answer this question we analyze the variations in sentiment expressed in different segments of a conversation. For processing the data, we examine the effect of changes in the sentiments expressed in a recorded conversation via sentiment analysis of the transcribed conversations. Moreover, we utilize the EDA biometric signal which records the skin responses to any affective cause. We therefore test the effect of these two signals for the detection of forgettable/memorable segments of a conversation.

6.4 Dataset

Our dataset consists of recordings of workplace meetings of six groups of people. Each group consisted of two members. For each group, the audio of four consecutive meetings over four weeks in addition to their biophysical sensor readings were recorded. Our dataset is real-world and captured in the wild, meaning that the involved participants were asked to simply have their usual meetings with no regulations imposed from our side. The audio was recorded with an audio recorder and the biometrics were recorded using the Empatica E4 wristband sensor[2]. Figure 6.2 shows the Empatica E4 wristband next to a Narrative Clip wearable camera that is used to automatically capture images (e.g. the one shown in Figure 6.1).

Statistics of the Dataset Overall, there were twelve participants recruited for

[2] https://www.empatica.com/e4-wristband



	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6
🟦 Total # of Words	21336	8642	14376	22122	24411	20950
🟧 Ave. # of Words (per meeting)	4267	2160.5	3594	5530	6102	5237
⬜ Total # of Unique Words	4240	2225	3043	4152	3029	3337
🟨 Ave. Duration of a Meeting (Seconds)	2337	1037	2511	2539	3079	2261

Figure 6.3. Statistics of our Dataset

this study. For recruiting participants, we looked for groups of two people who usually had a weekly work-related meeting. No restrictions were imposed on the meetings from our side. Thus, we were able to collect a real-world dataset of workplace meetings.

Converting audio to text: Subsequently, we transcribed all audio recordings of the meetings using an online transcription service[3] at a cost. The transcription error according to the service is 1%. The transcriptions of the conversations are time-stamped at fixed time intervals of one minute. Later in this section, we explain how we use the time stamps for synchronizing the transcribed text with other signals.

Basic statistics: Some important statistics of our transcribed audio recordings of meetings are reported in Figure 6.3. The report includes per-group statistics, such as total number of words in all four meetings, average number of words per meeting and the number of unique words in all four meetings. In general, Figure 6.3 shows the variability in the statistics of meetings and the behavior of the different groups.

Overall, we recorded 917.6 minutes of audio, where the average duration of a meeting was 38.23 minutes. The average number of days in between every two consecutive meetings for all the six groups when rounded down to full days was 8.

Extracting Text Segments: we first extract text segments using the popular Texttiling [65] segmentation algorithm. This algorithm uses word co-occurrence patterns in sentences to detect changes in the topic of a segment.

Texttiling [65] is ‘a technique for subdividing texts into multi-paragraph units

[3] http://www.rev.com

that represent passages or subtopics'. It utilizes patterns of lexical co-occurrence and distribution as discourse cues for identifying major subtopic shifts. We note that the texttiling algorithm cuts segments in documents only at sentence endings. Therefore, one segment would contain one sentence at least. Our purpose behind using texttiling is to split a conversation into topically coherent segments, such that if one segment correlates with a change in biophysical responses of one of the people involved in a conversation, we would be able to infer that the topic discussed in that segment has triggered the change. This characteristic of the texttiling algorithm makes it useful for our goal.

Furthermore, in order to synchronize the biophysical sensor readings with the textual transcription of a conversation, we use the following steps:

1. We use the per-minute time stamps of the transcribed audio and the number of words spoken within every one minute time lapse to determine the speed of the conversation.

2. By using the ending points of each segment of the conversation (determined by the texttiling algorithm) along with the number of words within each segment, we compute a close estimation of the length of each segment in seconds.

To achieve this we locate the next per-minute time stamp directly after a segment end. Then since we know the time lapse until the per-minute time stamp, we only need to compute the time lapse until the end of the texttile segment and add it to the previously known duration of time. Additionally, we know the number of words spoken until the next per-minute time stamp and the number of words spoken until the end of the texttile segment. Therefore, we compute the duration in time for these words by:

$$DTime = \frac{\#w_{ts} \times 60}{\#w_n} \tag{6.1}$$

where $DTime$ is the difference in time of the texttile segment with the next per-minute time stamp, $\#w_{ts}$ is the number of words in between the last per-minute time stamp and the next texttile segment ending, and $\#w_n$ is the number of words in between the previous per-minute time stamp and the following per-minute time stamp. The same method for synchronizing text and other signals holds if the beginning of a texttile segment is not at the start of a per-minute time stamp, but its somewhere in between.

The result of this step is multimodal signals synchronized by time.

Labeling the Text Segments: we recorded four meetings per each group over four weeks. Immediately before the start of each meeting we held an interview with each participant, asking them to describe everything they remembered from their previous meeting. Thus one week after each meeting, we held what we call a recall session where each participant described everything one could recall while being audio recorded. Then, similarly to the meetings, the recordings of the recall sessions were transcribed.

Finally, by computing the LSI [67] topic similarity (after preprocessing steps such as stop words removal, converting all words to lower case, etc.) on all segments of a meeting we created a topic model of that meeting. Subsequently, by querying the model with the two corresponding recall sessions, we automatically computed objective labels on how memorable each segment of a meeting was for an involved participant. This was done by comparing every segment of the meeting with the corresponding recall sessions based on the LSI topic model on the segments. Finally, the similarity between each segment and the corresponding recall sessions are computed based on cosine similarity. Therefore, by computing the semantic similarity between each segment and a segment we produce objective labels of how much a participant remembered or forgot.

Meetings differ from one another in terms of topics, depth, breadth, etc. Therefore, the range of similarity scores that are computed based on LSI differ from one meeting to another. Hence, for each meeting we normalize the scores by subtracting from the computed similarity score the average similarity score of all segments within that meeting. Finally, we determine a decision threshold for discriminating memorable from non-memorable. For this purpose, we examined the dataset of meetings to determine the decision threshold empirically. In order to determine this threshold, we manually looked into the dataset. We found out that a threshold value of 0.10 is optimal for distinguishing memorable segments from the others. The overall number of segments in our dataset is 1008, from which 616 segments are labeled as memorable and 392 labeled as forgotten. We note that the number of segments per meeting vary, because as explained earlier the segments were computed using the texttiling algorithm and each meeting may vary in the number of segments which are topically coherent.

6.5 Preliminary Analysis

In this section we present a preliminary analysis of our dataset through a case study to verify the applicability of our research goal. We will first, however, briefly describe the sentiment analysis algorithm that we implemented along with a brief explanation of EDA signals.

6.5.1 Sentiment Analysis Classifier

For analyzing sentiment, we implemented an unsupervised sentiment classifier similar to the one used in [19, 20]. The reason that we used this classifier over the commonly used SentiStrength classifier [131] was that it significantly outperformed the SentiStrength classifier [131] on three different datasets. Therefore, by using this classifier we opted for a sentiment analyzer superior to the SentiStrength.

Here we briefly explain the sentiment classifier that we implemented. Each conversation segment is split into smaller text snippets based on the punctuation marks '.', '!' and '?'. We therefore define a text snippet as a number of words that occur in between two punctuation marks.

The algorithm first replaces slangs with their equivalences using a slang dictionary to identify sentiment words more accurately. To build this slang dictionary, we manually collected slang phrases to include in the dictionary by using as many online resources that we could find, and furthermore adding the slang dictionary of SentiStrength [131] to our collected dictionary.

Then in a second step, we used a modified sentiment lexicon [22] to tag all sentiment-bearing words in each conversation with their corresponding sentiment scores. We further tagged all intensifier words (e.g. absolutely) and diminishers (e.g. might) with their corresponding scores. Additionally, we tagged negation words. Finally, if a word did not belong to any of the mentioned categories, it was tagged with the score '0'.

After having all the words in a document tagged either by their score or type, now we should handle occurrence of intensifiers, diminishers, and negations. First, the algorithm intensifies the strength of a sentiment-bearing word that appears after an intensifier word, by the score of that intensifier word. Analogously, in the case of diminishers, we weaken the strength of a sentiment-bearing word that appears after a diminisher word by the strength of that diminisher. Finally, for handling negations, we flip the polarity of the score of a sentiment-bearing word that appears after a negation. Furthermore, we weaken the flipped sentiment score by 1. That is, if the flipped score is positive, we subtract 1 from it and if it's negative we add 1 to it. Note that, while performing the above mentioned computations, in all cases we ignore the '0' tagged words that appear in between one of the above mentioned valence shifters and a sentiment-bearing words in a single text snippet.

In order to compute a sentiment score for a text snippet, we aggregate the words' scores. We define the decision threshold for classifying text snippets as '0'. That is, if the overall sentiment score of a document is less than or equal to '0' it is classified as negative, otherwise it is classified as positive.

As mentioned earlier, this classifier outperformed SentiStrength in terms of classification accuracy and F_1 measure and area under the curve of a receiver operating characteristic curve on three different datasets[20].

Finally, for computing the overall sentiment score of each segment, we average the sentiment score of all text snippets present in that segment.

6.5.2 EDA Signal

EDA sensor reading is a measure of electrical skin response excreted by the eccrine glands, which is connected to the sympathetic nervous system [44]. It is most commonly associated with changes in sympathetic arousal. Anger, happiness, interest, excitement as well as disgust can all cause changes in EDA. It is important to note that EDA is not correlated with valence, so both positive and negative affect can alter sympathetic arousal [44]. Thus, changes in EDA are most commonly associated with changes in sympathetic arousal. However, factors that cause increased sweat production, such as humidity, room temperature, and physical exertion, can also cause increased EDA amplitude. However, in the case of our meetings dataset, since the participants stay in the same room throughout a meeting, such effects are controlled.

In the case of our meetings dataset, each meeting involving two persons who are both engaged in the conversation, it is reasonable to believe that most alteration in the EDA signal is caused by the content of the meeting. If there would have been more than two people in a meeting, there would have been the possibility that a person would have been left out of the conversation or would have not followed the conversation, hence our assumption in that case would not have been realistic. An EDA biophysical recording will contain an individual's response to the content of a meeting and the behavior of the other individual involved in the meeting.

6.5.3 Case Study

Now that we explained our sentiment analyzer and the basics of EDA, we can proceed with analyzing the data. We first start our analysis with a case study on the dataset. The goal of this case study is to examine whether there is any correlation between the sentiment expressed in the transcribed conversation and memorability, between EDA signal and memorability or even between EDA and the sentiment expressed in the text. The result of this analysis not only can help us in designing a model for predicting the memorable/ non-memorable moments of a conversation, but also can be used as a helpful finding for other researchers in the field.

Since we are dealing with time series data (a sequence of topically coherent segments extracted from a meeting along with their corresponding time-synchronized EDA signal) we use the Cross Correlation Function (CCF) which in essence is the convolution between two signals. The reason behind using CCF is that it analyzes the correlation between two signals at different lags (i.e., displacements of the signals). In our case each lag is a segment of a conversation. Such correlation analysis, for example, can reveal not only a possible correlation between sentiment of a segment and its corresponding memorability score, but also can show if there is a correlation between the sentiment of the previous or next segments and the memorability of the current segment. Thus, since CCF is a useful method for assessing the effects of different parts of two signals on one another we utilize it in this experiment.

The results of our case study on randomly selected participants and meetings are presented in Figures 6.4, 6.5, and 6.6. We randomly selected five participants from five different meetings of our dataset for this case study. Figure 6.4 shows that the there is a strong correlation between memorability and negative sentiment. Also, it shows that the segments that are preceded by segments with negative sentiment are more memorable. A segment of a meeting expressing negative sentiment, for example, could be regarding difficulties and problems in a project, or in extreme cases arguing and strong disagreements between the involved participants. This finding also mirrors the results of psychology research such as [139] which show that negative experiences are more memorable for people than positive ones. The fact that we could support the findings of a psychology lab study with our study on a in-the-wild dataset using information retrieval techniques is an interesting finding.

We note that in Figures 6.4, 6.5, and 6.6 the height of the dashed lines (in both positive and negative directions) indicate the significance threshold and all bars that cross these lines being statistically significant.

Figure 6.5 reveals that memorability correlates with a rising EDA signal. However, we also observe that a memorable segment is preceded by a falling EDA signal. Thus, we conclude that shortly after a local minimum of an EDA signal we can find a significant reduction in forgetting, and thus an increase in memorability.

Furthermore, Figure 6.6 shows that an increase in sentiment intensity, regardless of the polarity, triggers the rise of the EDA signal. We note that since

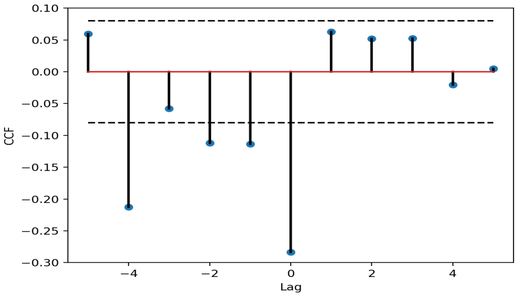


Figure 6.4. Cross Correlation between Sentiment and Memorability at different lags (each lag is a segment of a meeting)

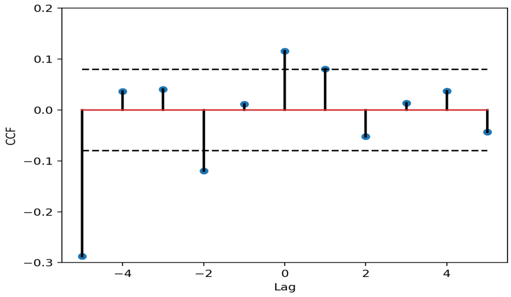


Figure 6.5. Cross Correlation between EDA and Memorability at different lags (each lag is a segment of a meeting)

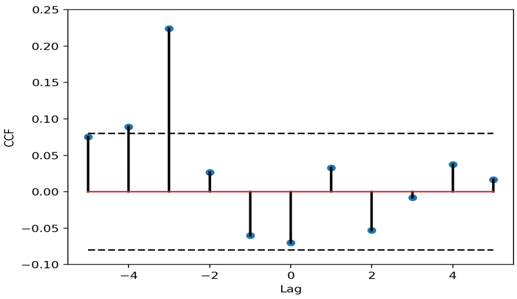


Figure 6.6. Cross correlation between sentiment and EDA at different lags (each lag is a segment of a meeting)

EDA is sensitive to both positive and negative emotions, we only computed the absolute sentiment score in Figure 6.6 and did not take into account the positiveness or negativeness of the sentiment scores. Hence, we can conclude that sentiment intensity correlates with an increase in the EDA signal.

6.6 Methodology for Predicting Non-Memorable Segments

Now that we have a basic understanding of the problem of predicting non-mameorable/ memorable moments in a conversation, we strive for designing a model that can accurately foresee parts of a conversation that a person is likely to remember or forget. One of the best methods when dealing with time series data is HMM. In [104] we can find a number of time-series prediction tasks where HMMs hold the state-of-the-art performance, motivating our use of HMM for this task. We use an HMM in an unsupervised setting.

HMM is a generative probabilistic model in which a sequence of observable outputs is generated by a sequence of hidden states. In our work, we use HMM in a specific way to unpack the latent hidden structure in the meetings to predict the segments of a conversation that one forgets versus the ones that one remembers. In the following we explain the HMM architecture that we adopted based on the notation of (Rabiner, 1990) [106].

Among the various problems that HMM can solve, we aim at: *given the observation sequence of $O = O_1, O_2, \ldots O_T$, and a model $\lambda = (A, B, \Pi)$, how to select a corresponding state sequence $Q = q_1, q_2, \ldots q_T$ which is optimal in the sense that it best explains the observations,* where A is the state transition probability distribution, B is the observation symbol probability distribution and Π is the initial state distribution. To solve this problem the model λ is trained using the Baum-Welch algorithm given the input data. This algorithm implements an Expectation-Maximization algorithm to find the maximum likelihood estimate of the parameters of an HMM, given a set of observed feature vectors.

In the case of the problem we are addressing in this thesis, our model architecture has seven states shown in Figure 6.7. We explain the observable outputs and the architecture in the next few paragraphs. Training the model starts by choosing an initial state q_1 according to the state distribution Π which assigns equal probability to all states. Furthermore, based on the state transition probability, the model moves to the next state and a new observation will be selected till all T observations in the sequence are generated. Given the observation sequence, the same explained process can be repeated to model an HMM. Subsequently, the Viterbi algorithm is used for decoding the most likely sequence of the transition states, hence predicting the forget/remember segments.

The architecture of the HMM that we deploy has seven states, and two output emission outputs. One of the two output emissions represents memorable segments while the other represents the forgotten segments. We use the Baum-Welch algorithm with 20 training cycles for learning the backward and forward probabilities. Additionally, the propagation of probability scores follows a Gaussian function. After the model is trained, we use the Viterbi algorithm to decode the best solution in the HMM network and find the best path. Viterbi is a backtracking algorithm based on dynamic programming which defines $\delta_j(t)$, i.e., the single best path of length t which accounts for the observations and ends in state S_j, such that $\delta_j(t) = \max_{q_1 q_2 \ldots q_{t-1}} P(q_1 q_2 \ldots q_t = j, O_1 O_2 \ldots O_t | \lambda)$.

Figure 6.7 presents the architecture of our model. For determining the non-memorability/ memorability of segment S_t, the model considers the features present in the sliding window of the three segments before it, the segment itself and the three segments after it. Each transition between states depicted in the figure, bears a probability score designating the strength of the transition. Furthermore, each of the states are connected to two output emission nodes, where one represents a non-memorable segment and the other represents a memorable segment. The result is an output prediction label for memorability of the segment S_t.

The features that we extract from the meeting contents and the corresponding biophysical signals are the moving average of the EDA signal per each meeting and participant as well as the corresponding sentiment signal expressed in the conversation. Furthermore, we compute the Canonical Correlation Analysis (CCA) of both signals. The use of CCA is motivated by the existing correlation

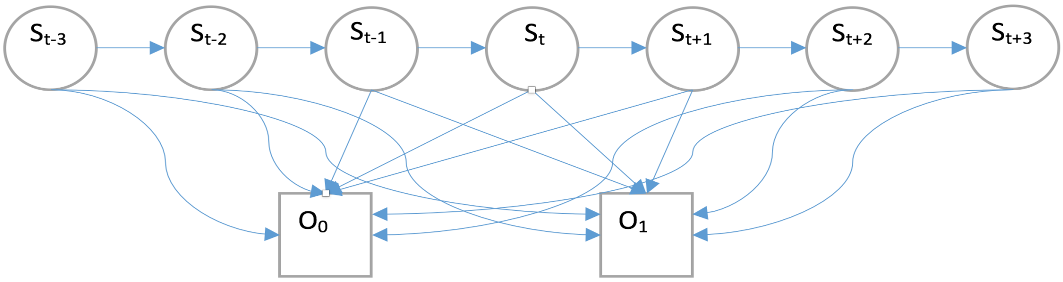


Figure 6.7. The architecture of the HMM proposed for memorability prediction. S stands for an HMM state.

between sentiment and EDA shown in a case study in Figure 6.6 as well as the correlation between each of these signals with memorability. CCA captures the co-variance between two signals and hence a suitable representation of the physical response of each participant in relation to an expressed sentiment within the conversation and vice versa.

Subsequently, for every segment of a conversation we keep a window of the CCA values corresponding to three segments before and after the target segment, and the CCA value corresponding to the target segment itself. For segments, such as the first segment of a meeting, that there is no CCA values for its previous segments, we use padding with identical values to the closest neighbor where such computation is viable. Therefore, for each segment of a conversation we feed the HMM a feature vector of length seven consisting of the described CCA values.

Finally, we compare the output of the HMM against the ground truth and evaluate the model.

The advantages of our approach are: First, it is unsupervised and there is no need for huge datasets to train it. Second, the use of CCA features, which capture co-variance between two signals is very critical to our approach due to the fact that CCA converts the data into a single signal which is normalized across different participants and meetings. This approach, generalizes our model to different participants and meetings.

6.7 Evaluation

For evaluating our proposed model, we compute its precision, recall and F_1 measure by using our dataset. We compare the performance of our model against two baselines:

Table 6.1. A comparison between our memorability prediction model against a random baseline

	F1 (%)	Prec. (%)	Recall (%)
Our Model	68.53	55.66	86.45
EDA Mean Slope (baseline)	59.81	50.45	73.43
Random (baseline)	55.08	59.97	50.97

Random baseline: the random baseline is a model which predicts the labels of each conversation segment by assigning it a randomly generated label. For computing the random baseline, we find the average performance of five runs of a random predictor. Then we compare the results against the performance of our proposed prediction model. By including the random baseline in our experiments, we can show that the prediction of non-memorable segments of a conversation using our model is possible and the improvement over the random baseline is statistically significant.

Mean slope of EDA signal: as explained in Subsections 6.5.2 and 6.5.3, a rise in the EDA signal correlates with sympathetic arousal caused by an emotion such as anger, happiness, excitement, stress, interest, disgust, etc [64]. Thus, capturing rises and falls of an EDA signal can be a beneficial representative of the signal. In order to use an EDA signal we time-synchronize it with the corresponding transcription of a corresponding conversation, following the same procedure described in Section 6.4, and divide it in segments. Each segment of the EDA signal corresponds with its respective transcription segment. Then we compute the first derivative of each EDA segment between every two consecutive points in the signal (i.e., with a sampling rate of five seconds) and subsequently computing the average of all derivatives. This method was used in [64] to process an EDA signal for detecting emotions. For every two consecutive points in the EDA signal if the first point is less than the next point, the signal has a rising trend and the first derivative is positive. Thus, we consider a positive mean slope as a positive class and thus memorable. On the contrary, if the average slope would be negative or zero the segment is considered non-memorable.

Table 6.1 presents the results of performance comparison of our model against the two baselines. We observe that the performance of our model is significantly higher than both the derivative of EDA and the random baseline. In order to confirm the statistical significance we used the two tailed paired t-test with the p-value of less than 0.05.

Thus, we experimentally demonstrated that this novel direction of research

Table 6.2. A detailed comparison between different features to train the HMM presented in Figure 6.7

	F_1 measure (%)
HMM + CCA	68.53
HMM + EDA	62.77
HMM + Sentiment	65.24

has the potential of turning into an important area of research that could truly help people throughout their everyday lives. People with good memories, or even people with mild memory deficiencies can benefit from such technology and live healthier lives with respect to their memory. Companies could use such technologies to improve the performance of their employees hence decreasing the work cycle. Such technology could be also used in the smart room scenario [144] to set personalized reminders for people to remind them about previous meetings.

In a second experiment, we strive for testing the effect of sentiment and EDA signals independently in order to measure the contribution of each of these signals to the non-memorable/ memorable segments prediction problem. To achieve this, we compare the performance of our HMM model trained with CCA features, against the same HMM once trained only with sentiment features and another time only with EDA features. We follow the same strategy of feeding the model with features of three segments before and after a target segment and the target segment itself. We compare the three models trained with three different features to examine the effectiveness of each signal. Table 6.2, presents the results of this experiment. The model trained using the CCA features outperforms both other models. Moreover, the sentiment features outperform the EDA features. This indicates that sentiment is a very strong feature in representing the content of a conversation. However, we observe that the EDA affective signal is still an important feature in combination with sentiment.

6.8 Conclusions

With the rapid proliferation of smartphones and wearable devices with powerful hardware, the possibility of developing various human aid applications that run on these devices is more than ever. One of the most important directions for human aid is assisting human memory by reminding one about the parts of a conversation that one is likely to forget In this chapter, we first presented a case study which revealed what are the significant patterns for detection of non-memorability/ memorability of conversations, in terms of sentiment and EDA signals. These new findings, can open a path to powerful memory augmentation and personal assistance tools of the future which may significantly improve the lives of many people by helping them reminisce their memories. Furthermore, based on our findings we introduced a novel method customized for each person for predicting the segments of a conversation that is likely to be forgotten. This could be helpful to focus the user's attention on those segments by showing her reminders. Our experimental results showed the efficacy of our proposed model.

This novel study can serve as a foundation for future work in the domain of memory augmentation. There are a number of directions for future work. Other signals such as heart rate or blood volume pressure can be investigated. Furthermore, an exploratory study of our dataset such as the effect of one attendee forgetting/ remembering a segment of meeting on the other attendee is another direction for future work.

Chapter 7

Just-In-Time IR: Predicting Topics of a Future Meeting

7.1 Introduction

In Chapter 6 we presented a method for predicting segments of a conversation that one is likely to forget/ remember, based on biophysical sensor data and the sentiment expressed in a conversation. In this chapter, we present another method useful for augmentation of human memory in meetings that given the history of one's previous meetings, predicts those previously discussed topics that are likely to be revisited in a future meeting.

Memory augmentation is the process of providing human memory with information that facilitates and complements the recall of an event in a person's past. Recently, there has been a lot of attention on processing the content of meetings for later reuse, such as reviewing a meeting for supporting failing memories, keeping in mind key issues, verification, etc. That is due to the fact that meetings are essential for sharing knowledge in organizations. In this chapter, we propose four novel time-series methods for predicting the topics that one should review in preparation for a next meeting. The predicted/recommended topics can be reviewed by a user as a memory augmentation process to facilitate recall of key points of a previous meeting. With the growing number of meetings at an organization that one may attend weekly and with the growing number of topics discussed, forgetting past meetings becomes inevitable, hence recommending certain topics to the user in order to prepare the user for a future meeting is beneficial and important. Our experimental results on real-world data, demonstrate that our methods significantly outperform a state-of-the-art HMM baseline. This indicates the efficacy of our proposed methods for modeling semantics in temporal data.

As described in the previous chapter human memory is a critically important cognitive ability that may sometimes fail due to the volume and intensity of information that we are exposed to on a typical working day, or due to our lack of adequate attention, or due to aging. In a workplace environment, failing to recall important work-related events can cause problems reducing productivity and efficiency. In a study by Jaimes et al. [71], common issues that people forget regarding a past meeting were studied. Among several investigated issues, inability of participants to recall a significant amount of a previous meeting's content (i.e., dialogue) after a week time is noteworthy.

Augmentation of human memory in a workplace environment has been proposed as a solution in preventing failure to recall past events [38, 71]. Jaimes et al. [71], proposed a framework for memory cue based retrieval of meeting content. Moreover, Lamming et al. [50] studied memory problems in a workplace environment and designed a system referred to as "memory prosthesis". Automatically recorded data of user activities could be later retrieved by their system for remembering things, specially those that the user did not think were needed to be remembered at the time. They explain further that context-sensitive reminders can be shown to the user for reminicing past work-related memories. From a somewhat different perspective of personal lifelog management, [38] also proposed a memory augmentation tool with multimodal memory cues. They explain that "being reminded of information in a work situation (e.g. previous meetings with an individual)" could be an application of their system. Their work among other similar works, rely on the advent of various wearable data capture devices (e.g. wearable video/audio recorder or biophysical sensors) which has recently created great opportunities to utilize the collected data for various human-aid applications such as support of failing memories and memory augmentation.

As discussed in 2.1.2, using summarization tools for memory augmentation, as opposed to searching keywords has been proposed [13]. In such settings a summary of the meeting can reminisce the faint memories of that meeting. At the same time, [96] shows that extractive summaries provide a more efficient way of navigating meeting content than simply reading through the transcript and using audio-video records, or navigating by keyword search. As a result of the two rationales, other work on processing meeting content [146] have focused on extracting summaries from a meeting transcript including extraction of LDA [34] topics for producing summaries. In support of meeting summaries, [69] explains that "users absorb information in summaries more quickly than in full text, despite some loss of accuracy". They study meeting summarization by focusing on user needs such as highlighting the most important decisions made in a meeting. Motivated by these studies we aim at predicting the continuation of LDA topics of previous meetings in a future meeting to address yet another users need: preparation for the next meeting.

As discussed previously, the Remembrance Agent [108] is also an interesting framework for aiding human memory in a proactive fashion.

Addressing users' near-future information needs, has been also studied in the context of personal assistants such as Google Now, Microsoft Cortana or Apple's Siri. These systems offer proactive experiences [123] that aim to recommend *"the right information at just the right time"* [127].

In this chapter, we focus on proactive augmentation of human memory [15, 16] with respect to the content of previous meetings in a workplace environment, by reminding a user of relevant topics of previous meetings that will be important in the next meeting. We use two real-world datasets of weekly meetings of ten groups of people. In these datasets, we recorded real workplace meetings of each group (where a group consisted of two individuals) over a span of an entire month (in the case of the first dataset) and over six weeks (in the case of the second dataset). We compare various methods for predicting the topics of a conversation that will be continued from the previous meetings, which should therefore be reviewed by the people involved in that meeting to prepare next meeting. To the best of our knowledge, this is the first work that aims at augmenting human memory by predicting the continuation of intermittent topics in consecutive meetings.

The main contributions of our research presented in this chapter are:

- we propose the idea of augmenting human memory by predicting the topics of one's next meeting.
- to address the above problem, we develop four new methods for performing this task. These methods are all based on a probabilistic word embedding model that tracks each word in all its different contexts (i.e., co-occurrence with other words) over consecutive meetings.

7.2 Methods

In the previous section we discussed that our goal in this study is *predicting the topics that one should review in preparation of one's next meeting* given her previous meetings. Note that we did not state our goal as predicting the topic of one's next meeting due to the fact that in most situations future events are unpredictable. For instance, in our dataset of conversations it occurred between two individuals that before their last meeting one of them had been sick. Therefore, for the first time out of all their conversations he talked about his sickness and a new topic was introduced in their conversation. Predicting such future events is virtually impossible. Hence, our goal is only to predict which topics will continue from previous meetings, not predicting the unpredictable.

In this section, we present four new different methods as well as the baseline method for predicting the topics to be reviewed in preparation of one's next meeting. In our research of methods that can effectively predict the continuing topics over time, we devised various methodologies. One of the methods we designed assigns average weights to all words across all consecutive meetings, and disregards the passage of time. We designed two other methods, one which increases the weights of most recent words, and another which, increases the weights of more established (i.e., persistent) words. Additionally, we developed an evolutionary method that combines both recency and establishment effects, while estimating weights for each of these effects over time. We elaborate on the details and rationale behind each of these time-series methods in the following subsections.

Finally, we use HMM [106] as the state-of-the-art baseline for unsupervised multi-variate time-series prediction, and compare its performance against our proposed methods. HMMs have shown strong results in predicting multivariate time-series in different applications [150] and are commonly considered as the state-of-the-art in a number of domains. Pietrzykowskiand et al. [104] enlists a number of papers that describe domains where HMMs hold the state-of-the-art performance. We elaborate on the HMM baseline approach in the last subsection of this section.

7.2.1 The CorrAv Effect

We model each word from each meeting using a Gaussian Mixture Model (GMM) with K components, where each component c_i represents a context word for the target word being modeled. We define a context word as a word which co-occurs with the target word in one or more sentences of the meeting. Thus, we obtain the following equation density function f_w:

$$\begin{aligned} f_w(\vec{x}) &= \sum_{i=1}^{K} p_{w,i} \mathcal{N}(\vec{\mu_{w,i}}, \sigma w, i) \\ &= \sum_{i=1}^{K} \frac{p_{w,i}}{\sqrt{2\pi |\sigma_{w,i}|}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu_{w,i}})^T \sum_{w,i}^{-1}(\vec{x}-\vec{\mu_{w,i}})} \end{aligned} \tag{7.1}$$

where $p_{w,i}$ is the probability of a component modeling a certain context with $\sum_{i=1}^{K} p_{w,i} = 1$, $\mu_{w,i}$ is the probability of the position of the i_{th} component and σ_w models the uncertainty of the context. We learn the parameters of $GMM_{i,n}$ for word i from meeting n to compute probability of each context of each word in each meeting. This will lead to computing $p_{w,i,n}$ which is the probability of word w in its i_{th} context from the n_{th} meeting.

Finally, we formally define the CorrAv effect as follows: given the topics of your last n consecutive meetings, we would like to predict which topic continues in the $(n+1)_{th}$ meeting. Lets say we have a vocabulary v of all the words occurring in the first n meetings. We construct a word vector containing the average probability of presence of all words in each of their contexts in v. Finally, we use the following equation to compute the aggregate probability of word w in a certain context under the CorrAv effect:

$$P_{w,c} = \sum_{n=1}^{N} \sum\nolimits_{w_i \in v} \frac{BM25_{w,n} * P(w_{i,c,n})}{(n)} \tag{7.2}$$

where n is the meeting sequence number, $P(w_{i,c,n})$ is the probability of word w under component c (i.e., context c), derived from the n_{th} meeting. The resulting constructed word vector is an average representation of probability of all words present in all n meetings. Finally, $BM25_{w,n}$ is the weight of the word w in meeting n computed using the probabilistic algorithm BM25 [112] in each of the previous meetings.

Using Equation 7.2, we learn a reference vector for each word in each context across all previous meetings. As it will be explained later in Section 7.3.1, our dataset contains LDA topics extracted from the first n meetings whose continuation in the $(n+1)_{th}$ meeting should be predicted. For each given topic whose continuation should be predicted, we construct a new word vector derived from the reference vector. That is due to the fact that in the reference vector we have computed each word in all its contexts. However, words have different meanings or contexts, and an LDA topic model puts together words that often reflect the same context. Therefore, we have to select relevant contexts for each topic whose continuation is being predicted. We search through each given LDA topic and identify context words of a target word that have the highest probability in the LDA topic. Subsequently, from the reference vector we add the probability of each word in its identified topic to the new vector that we construct. We name this new vector the $Vector_t$ where t refers to the LDA topic whose continuation should be predicted, meaning that for each topic t we construct a unique vector of contextual words. We emphasize again that $Vector_t$ contains all the words in v with the difference that if a word w is present in topic t we use the context probability of the word which best suites the context of t, and otherwise we assign it the highest context probability.

The last step is to compare a $Vector_t$ with a topic t to make the prediction. Other previous work [94] have used element-wise dot product of word vector distributions as an energy function which would show how similar two vectors are and could be used for predictive tasks. However, we take a different approach by computing the correlation between two vectors as a measure of similarity. This is inspired by the underlying belief in topic models and word embeddings that words have certain meanings in certain contexts. Thus, looking at the words co-variance can better capture a context similarity or difference.

As a result, we compute the Pearson correlation of $Vector_t$ with the topic t to make the prediction. The result of this step, is a ranked list of all of the t topics from the first n meetings based on this correlation value. According to the CorrAv effect, the topics highly correlating with the average probability of all words are those that are most likely to continue in the $(n+1)_{th}$ meeting, hence the naming. We use the Pearson correlation metric defined as $P_{X,Y} = \frac{COV(X,Y)}{\sigma_X \sigma_Y}$, where $P_{X,Y}$ is the Pearson correlation of two populations X and Y, COV is the co-variance and σ is the standard deviation. Pearson correlation is a measure which captures the linear dependence between two populations X and Y, and returns a correlation score ranging from '-1' to '1'. In particular, it returns '1' if the two populations are identical, '0' if the two populations have no correlation, '-1' if the two populations are uncorrelated. In our use case, the Pearson correlation can capture the dependence between a topic t and a learned $Vector_t$.

Finally, we require a threshold for accepting a topic as a continued topic which needs to be reviewed. To determine an effective threshold we use n-fold cross validation. That is, we iteratively leave out each set of four meetings (for a single group) and compute the threshold which minimizes the Mean Squared Error (MSE) of prediction on the remaining folds. Then, using the computed threshold, we evaluate the left-out fold.

7.2.2 The Recency Effect

The recency effect assigns higher weights to the contextual words of the most recent previous meetings. Then it identifies those topics which are most correlated with the word vectors computed based on the recency effect.

We follow the same problem definition described in Section 7.2.1. All steps for computing GMMs for each word are the same as explained in Section 7.2.1.

The only difference of this model with CorrAv is that the computed probability scores are higher for words spoken in the most recent meetings. Therefore, after computation of GMMs instead of using equation 7.2, we compute the $P_{w,c}$ according to the following equation:

$$P_{w,c} = \sum_{n=1}^{N} \sum\nolimits_{w_i \in v} \frac{BM25_{w,n} * P(w_{i,c,n}) * e^{(n-1+\lambda)}}{(n)} \tag{7.3}$$

where n is the meeting sequence number, $P(w_{i,c,n})$ is a the probability of word w under component c (i.e., context c), derived from the n_{th} meeting. $BM25_{w,n}$ is the weight of the word w in meeting n computed using the probabilistic word ranking algorithm BM25 in each of the previous meetings. The $e^{n-1+\lambda}$ is the rate with which recent words from recent meetings are assigned higher weight. As explained earlier in Section 2.1, psychology research [47] has shown that forgetting is an exponential function of time. Thus, we compute the variable λ according to the forgetting rate of 77% shown by the psychology study. As a result of this computation λ is set to 1.125. Further exploration of this variable remains a future work and in this work we only rely on the findings of psychology research.

Using equation 7.3 we model each word in its various contexts in a temporal fashion, meaning that contextual co-occurrences of words are weighted differently in time.

Subsequently, all steps explained for the CorrAV method after Equation 7.2 were also applied in the case of recency modeling.

7.2.3 The Establishment Effect

The establishment effect, identifies topics containing words that have been frequently used or are in other words established. This method assigns a higher weight to words that have persisted from previous meetings. Therefore the establishment effect is the opposite to the recency effect, as it assigns higher weights to more persistent words.

Again, we refer to the problem definition stated in Section 7.2.1. We compute the GMMs for each word in the same way as explained in Section 7.2.1. The only difference of this model with CorrAv is that the computed probability scores are weighted higher for words which have been persistent over time. Therefore, after computation of GMMs instead of using equation 7.2, we compute the $P_{w,c}$ according to the following equation:

$$P_{w,c} = \sum_{n=1}^{N} \sum\nolimits_{w_i \in v} \frac{BM25_{w,n} * P(w_{i,c,n}) * e^{-(n-1+\lambda)}}{(n)} \tag{7.4}$$

where n is the meeting sequence number, $P(w_{i,c,n})$ is a the probability of word w under component c (i.e., context c), derived from the n_{th} meeting and $BM25_{w,n}$ is the weight of the word w in meeting n computed using the probabilistic word ranking algorithm BM25 in each of the previous meetings. The $e^{-(n-1+\lambda)}$ is the rate with which established words are assigned higher weight. As explained earlier in Section 2.1, psychology research[47] has shown that forgetting is an exponential function of time. Therefore, the rationale in the establishment effect is that if a word persisted in meetings of two individuals over time, we could weight this remembered word higher by the exponential forgetting factor. Similar to the recency model we compute the variable λ according to the forgetting rate of 77% shown by the psychology study. As a result of this computation λ is set to 1.125.

Analogously to the CorrAv and recency effects, all steps explained for the CorrAv method after equation 7.2 are also applied in the case of the establishment method.

7.2.4 K2RE

This method is a hybrid that combines the recency and establishment effects. It dynamically estimates the weights of each of these effects for each meeting and corrects itself over time. We refer to this method as *Kalman combination of Recency and Establishment (K2RE)*. It utilizes the Kalman filter [73] to estimate the Recency and Establishment weights over time. By measuring the meeting behavior of a group in terms of establishment and recency, K2RE adapts itself to data of each group. In the following, we first present a general overview of K2RE and then elaborate on its details. Figure 7.1 illustrates the components of the K2RE method. This method integrates scores from the recency and the establishment effects, described in Sections 7.2.2 and 7.2.3, using a linear interpolation.

The linear interpolation for a meeting at time slice t is defined as:

$$K2RE_t = w_{E,t} * Score_{establishment} + w_{R,t} * Score_{recency} \tag{7.5}$$

where $Score_{establishment}$ and $Score_{recency}$ are computed by the establishment and the recency effects, respectively. Furthermore, $w_{E,t}$ and $w_{R,t}$ are establishment weights and recency weights computed by the Kalman filter at time t, such that:

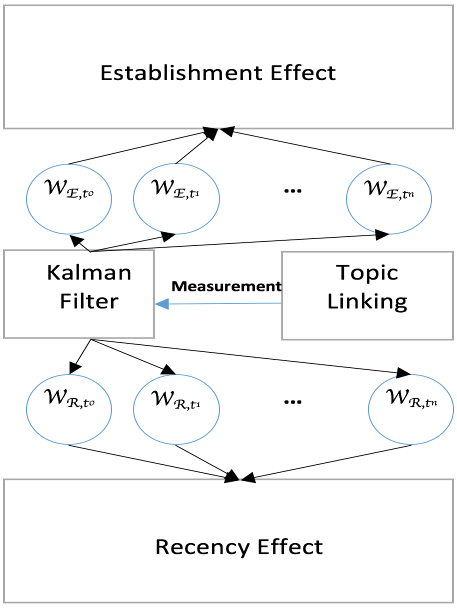


Figure 7.1. K2RE Method's Architecture

$$w_{E,t} + w_{R,t} = 1 \tag{7.6}$$

This means that at each time slice t each of the two effects will be given a weight, either equally or if one effect is assigned a higher weight, the other will receive a lower weight.

The Kalman filter is always initialized by assigning equal probability of 0.5 to both $w_{E,t}$ and $w_{R,t}$.

$$f(n) = \begin{cases} X^{G} = A^{G} f_{t-1} + \varepsilon_t{}^{G} & t = 2, \ldots, T \\ z_t = H^{G} f_t + w_t{}^{G} & t = 1, \ldots, T \end{cases}$$

where f_t is the system state at time t, A^{G} denotes the transition of the dynamic system from $t-1$ to t, H^{G} describes how to map state f_t to to an observation (i.e., measurement) z_t and both $\varepsilon_t{}^{G}$ and $w_t{}^{G}$ are mutually independent Gaussian noise variables with co-variances R_t and Q_t respectively. The superscript G in the system of equations explains that we compute these equations per each group. The dynamic system, then evolves over time and updates itself proportional to the Kalman gain.

Now that we explained the Kalman filter module and how it computes the weights for recency and establishment effects, we explain the topic linking module and describe how the measurement process explained in the Kalman filter equations works.

In Chapter 4, we introduced a topic model for tracking the evolution of intermittent topics over time [23]. In other words, this model tracks the evolution of topics that may occur discretely over time, such that a topic does not need to be necessarily present over all time slices. In the topic linking module shown in Figure 7.1 we use a component of the mentioned topic model. This component links together similar topics over time. As explained, the linking of a topic may be discrete or continuous under this model (i.e., a topic may be present over all time slices or it may skip some time slices). We briefly explain the model in the following. The general idea is to form a Gaussian random walk in a Markovian state space model. The Markov assumption enforces probabilities of a hidden state at time t to be computed merely dependent on the previous time slice and not all the previous states. We utilized this assumption to compute topic chains that capture the evolution of a topic discretely over time. If two topics over two different time slices are similar according to the following criterion, they will be linked.

$$\beta_{t,k}|\beta_{t-m,1..k} \sim \mathcal{N}(\beta_{t-1}, \sigma^2 I) \tag{7.7}$$

where $\beta_{t,k}$ is topic k at time slice t, $m \in (1,..,n)$ with n being the number of previous time slices (meetings) and σ is the maximum variance allowed from the mean of a topic in the previous time slice. By assigning a very small value to σ, the model links two topics that are highly similar. Furthermore, the Baum-Welch [28] algorithm learns the forward and backward probabilities of the transitions among the topics. We used this model as a component of K2RE for linking the topics that are similar across different meetings. The model takes as input the topics from the first n meetings whose continuation in the $(n+1)_{th}$ meeting is to be computed.

After linking similar topics over every two consecutive meetings, the topic linking module computes the recency rate of the topics for meeting n by computing the number of topics from the meeting $n-1$ that have been present in the meeting n divided by the total number of topics in the same meeting. This measurement is given as the observation matrix to the Kalman filter for each meeting which we explained above. Subsequently, the Kalman filter computes the evolution of recency and establishment weights using the Kalman filter equations presented above. Furthermore, using Equation 7.6, similarly to the recency and establishment methods, a K2RE reference vector is generated.

Moreover, analogously to the other three previous methods, all steps for computing word vectors and computation of Pearson correlation as an energy function are performed.

7.2.5 Hidden Markov Model Baseline

HMMs [106] have been extensively used for modeling multivariate time series and predicting next states.Therefore, in this thesis we also use HMM to predict the continuing topics as a baseline for our benchmark.

The architecture we use does not allow transitions between topics in the same time slice but it enforces connections between topics over consecutive time slices. This is set using the transition matrix. Additionally, the HMM we implement uses a Gaussian kernel. Using the Baum-Welch algorithm the model is trained. For determining the number of HMM output states, similarly to [150], we use BIC to find the optimal number of output states given the data of each set of four meetings. Finally, after training the HMM model with the topics of the first *n* meetings we measure the likelihood of each of the topics that we want to predict its continuation under the trained model. The result is a likelihood score per topic. We normalize the likelihood scores by dividing each of them by the maximum likelihood score. Finally, similarly to the previous models described in this section we compute the optimal classification threshold using n-fold cross validation.

7.3 Evaluation

7.3.1 Datasets Description

First Dataset

Our first dataset consists of recordings of workplace meetings of ten groups of people. Each group consisted of two members. For each group, the audio of four consecutive meetings over four weeks were recorded. Our dataset is real-world and captured in the wild, meaning that the involved participants were asked to simply have their usual meetings with no regulations imposed. Out of the ten groups, data of three groups were used as a development set, to design our models. The remaining seven groups were used as a test dataset In the following, we report the statistics of the test dataset

Participants: overall, there were 14 unique participants recruited for this study. For recruiting participants, we looked for groups of two people who usually had a weekly work-related meeting. No restrictions were imposed on the meetings from our side. Thus, we were able to collect a real-world dataset of

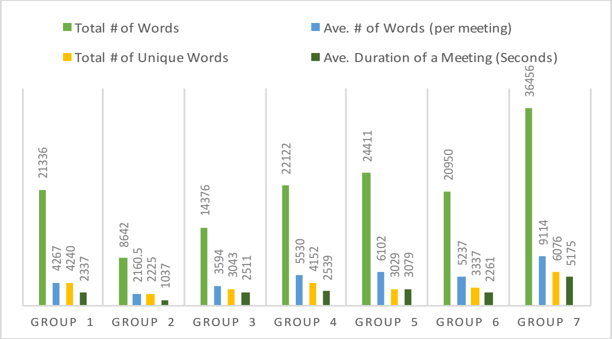


Figure 7.2. Statistics of the First Dataset

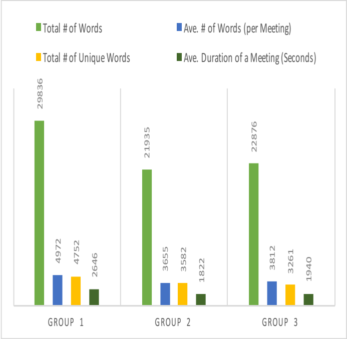


Figure 7.3. Statistics of the Second Dataset

workplace meetings. Due to the nature of this dataset (involving participants and being real-world) capturing it took a long time. All participants signed consent forms to be recorded.

Converting audio to text: subsequently, we transcribed all audio recordings of the meetings using an online professional transcription service[1] at a cost. The transcription error according to the website is 1%.

Basic statistics: Some important statistics of our dataset are presented in Figure 7.2. We note that by comparing the number of unique words in all four meetings combined and average number of words per each meeting, we observe how focused the topics of the four consecutive meetings are. As an example, by looking at statistics of group 7 we observe that the number of unique words used in all four meetings is almost half of the average number of words per-meeting. In general, Figure 7.2 shows the variability in the statistics of meetings and the behavior of the different groups.

The average number of days in between every two consecutive meetings for all the seven groups when rounded down to full days was 8.

Topic extraction: as explained earlier, we extract LDA topics from the texts. For this purpose, we treat every full sentence in the text as a document and extract LDA topics from all of them. Since the number of topics (K) discussed in two different meetings might vary, it is important to estimate the number of topics per each meeting. For this purpose, similar to the method proposed in [58], we went through the model selection process explained in Chapter 5. This consists in keeping the LDA Dirichlet parameters (commonly known as α and η) fixed, and assigning several values to K and computing an LDA model each time.

[1] www.rev.com

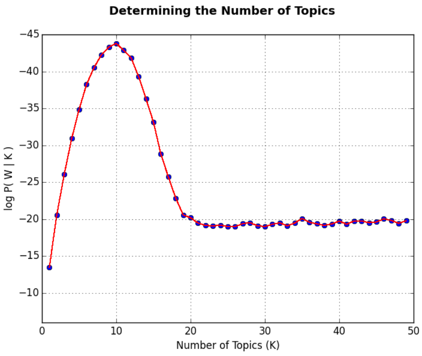


Figure 7.4. Model selection results for a randomly selected meeting from our dataset The plot shows the log-likelihood of the data under different number of topics, K.

Subsequently we picked the model that satisfied:

$$\underset{K}{\operatorname{argmin}}\, logP(W|K)$$

In Figure 7.4 we show an example of model selection to find the optimal number of topics for a randomly selected meeting from our dataset. As we can see in the plot, for this meeting the optimal number of topics is 10. We repeated this process for each meeting to find the optimal number of topics for that meeting. Figure 6.1 shows a real sample topic about "interface design" from our dataset

Labeling the extracted topics: In order to obtain ground-truth labels for the topics of the first three meetings that continue in the 4th, we asked a human assessor to label the topics of the first three meetings for every set of four meetings. The goal was to label the topics of the first three meetings from each set by examining whether a topic was also discussed in the fourth meeting or not. Therefore, by looking at the topics of the last meeting, the human annotator determines if any of the topics of the first three meetings continued in the last meeting. Hence, there are two possible labels to assign to each topic, i.e., 'continued' and 'not continued'. This is the ground truth for evaluation.

The assessor is given instructions on how to label the topics. These instructions include putting more emphasis on the top 30 words in each topic to take a decision. That is due to the fact that at the end, the users of our system usually look at the top words of each topic to understand it. On the other hand, using a k-nearest-neighbors implementation, information on the top five neighbors of each of the topics from the first three meetings in the last meeting is provided to the human assessor to simplify the annotation task. This is while the assessor is asked to take a final decision on the label, based on his own understanding. The assessor assigned to each set of the meetings is familiar with domain knowledge necessary to label it.

Statistics of labeled dataset: subsequently, our goal is to correctly predict the topics that have continued over time. After the topics of all first 3 meetings were labeled based on whether or not they are continued in the $4th$ meeting, the resulting number of labeled topics to be predicted was 205. Our goal is to correctly predict the assigned labels. The dataset of the labeled topics is unbalanced with 60% of the topics continued (positive class) and 40% that did not continue (negative class). Due to the bigger size and variation of this dataset, in most of the experiments that we conducted in the evaluation section this dataset was used.

Second Dataset

Our second dataset is recorded and prepared with the same process as the first dataset Thus, we skip the redundant explanations about the preparation process and merely highlight the differences. The two main differences of this dataset compared with the first dataset is the smaller size which is three groups, and the higher number of meetings recorded for each group which is 6. Hence, we use this dataset to analyze the effect of more number of consecutive meetings on the prediction models. Figure 7.3 presents some basic statistics of this dataset. We used this dataset in Subsection 7.3.3 where we tested the effect of the number of meetings over time. The average length of a meeting in this dataset is 35.6.

We prepared this dataset by labeling the topics of the first three meetings which will continue in the fourth meeting, then labeling the topics of the first four meetings which continue in the fifth meeting and finally labeling all the topics of the first five meetings which will continue in the sixth meeting. The number of labeled topics from the first three meetings whose continuity in the fourth meeting was labeled is 85 with 48 being continued and 37 not continued. Furthermore, the number of topics from the first four meetings was 112 where 65 constituted the continued class and 47 the discontinued class. Finally, the number of topics extracted from the five first meetings was 139 of which 70 were continued and 69 discontinued.

7.3.2 Evaluation Metrics

We performed a rigorous testing of all the four methods presented in Section 7.2 and compared them against the baseline method. For evaluation, we used standard information retrieval evaluation metrics, namely, precision, recall, F_1 measure and Mean Average Precision (MAP).

Our choice of evaluation metrics is influenced by two factors. First, we used metrics that are commonly used in the information retrieval community for prediction tasks. Second, since we are dealing with an unbalanced dataset, it was important to use metrics such as MAP or F_1 which work well even on unbalanced datasets.

Additionally, for the seven sets of meetings in our test data we always performed a 7-fold cross validation by iteratively leaving one set out and evaluating it using the threshold value learned from the remaining folds.

7.3.3 Experimental Results

All presented experiments, except the experiment on *number of meetings over time* in Subsection 7.3.3, are using the first dataset In our first experiment, we compute precision and recall values of all the proposed methods and compare them with the HMM baseline. Table 7.1 shows precision values at different levels of recall and for all decision thresholds. The values are obtained from per-group interpolated precision-recall curves. The table shows that the K2RE method outperforms other methods as well as the HMM baseline in terms of precision at all levels of recall. We also observe that on low levels of recall the second best performing method is the establishment effect. Additionally, we observe that on higher levels of recall the recency effect performs as the second best method.

In the second experiment we computed MAP as well as the F_1 measure and reported it in Figure 7.5. The F_1 measures are computed on the dataset using 7-fold cross validation for each of the seven groups. We can observe through this experiment that K2RE significantly outperforms the HMM baseline. We confirmed the significance in performance difference using a two-tailed paired t-test with the p-value of less than 0.05. Furthermore, the performance of the CorrAv, establishment, and recency effects were confirmed to be statistically significant in terms of the F_1 measure against the HMM baseline. We also present a comparison between all the methods with respect to F_1 measure and MAP in Table 7.2.

Effect of Parameters

In this subsection we analyze the effect of some parameters that influence our presented models.

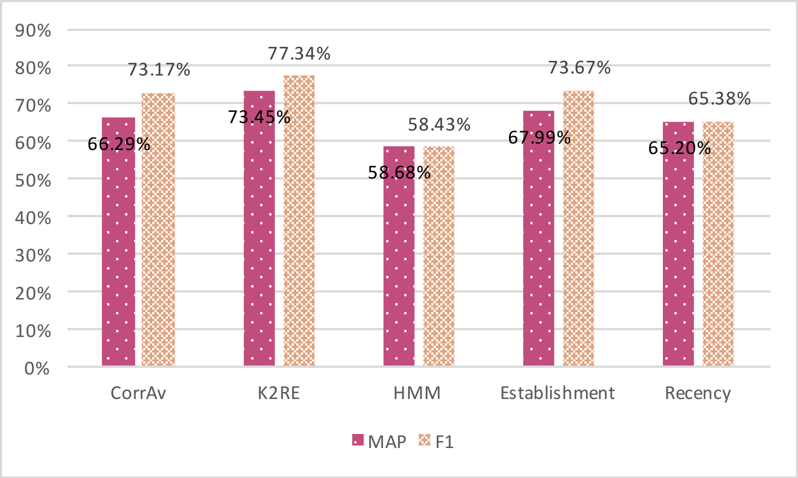


Figure 7.5. Comparison of our novel methods against the HMM baseline in terms of MAP and F_1 measure on the first dataset

Table 7.1. Precision (in percent) of all methods at different Recall levels for all decision thresholds, derived from per-group interpolated precision-recall curves on the first dataset

Recall (%)	Prec. CorrAv	Prec. Estab.	Prec. Recen.	Prec. K2RE	Prec. HMM
10	79.16	84.04	78.15	**87.25**	68.53
20	74.12	81.18	75.77	**85.46**	66.62
30	71.88	72.02	73.98	**80.10**	66.62
40	69.66	70.41	73.98	**79.03**	66.62
50	69.49	70.41	71.55	**77.28**	66.62
60	68.78	70.41	70.08	**77.15**	66.62
70	67.04	69.46	70.08	**76.63**	66.28
80	66.60	68.18	69.83	**74.15**	65.81
90	66.60	66.87	68.30	**73.15**	65.81
100	66.40	66.04	66.48	**70.25**	64.28

Table 7.2. A comparison of our proposed methods against the HMM baseline on the first dataset

	CorrAv	Establish.	Recency	K2RE	HMM
F_1 Measure (%)	73.17	73.67	65.38	**77.34**	58.43
MAP (%)	66.29	67.99	65.20	**73.45**	58.68

Data Sample Sizes

Second, we show the effect of different sizes of training data on MAP, since MAP is more indicative of the changes in models for all decision thresholds. Figure 7.6 shows the results of this experiment. We computed the MAP measure for two, four, six and seven groups respectively. The result of the same metric for all seven groups which is the full size of the first dataset was also presented in Figure 7.5 which we also include here for easy comparison. We can observe from the graph that the performance of our *K2RE* method is stable and superior to other methods for different numbers of groups being tested. Moreover, we generally see that almost all methods show the same behavior across the different data sample sizes. The only exception is the MAP value of the recency effect for two groups, that for the first time shows a higher performance than the establishment method. In our analysis of the data, we observed that one of the two groups is having mostly brainstorming meetings where they discuss different ideas for a project and they change ideas at a high pace. Although, we observe through this experiment, as well as the one presented in Figure 7.5, that for a larger number of groups on average, the establishment effect is a stronger model for predicting the continuation of topics, we also observe that there are cases where recency is a better model. This also proves the need for developing the K2RE method which can at all times adapt itself to the meeting behavior of a group being analyzed. The evolutionary K2RE method is clearly a stronger model than the others and as the number of groups being analyzed increases, its efficacy compared with the other models is more visible. Furthermore, another observation from Figure 7.6 is that the MAP value of the K2RE method starts to converge as the size of the dataset reaches four groups and higher.

Number of meetings over Time

Third, we analyze the effect of sequentially adding meetings to predict the continuation of topics from previous meetings to the next meeting over time. To achieve this goal we used *the second dataset* which consists of three groups with six meetings recorded for each group. In this experiment, we use the first three meetings of each group to train the models. As in the case of the previous experiments we aim to predict those topics of the first three meetings that will continue in the fourth. However, we then continue with predicting those topics of the first four meetings that will continue in the fifth meeting and so on. We measure the performance of the models in terms of MAP. As a result of this experiment we present Figure 7.7 with results of prediction on the fourth, the fifth and the sixth meetings for all three groups. As we see in the figure, the K2RE method shows better performance compared with all other methods over time. The performance of the establishment effect in terms of MAP drops over time,

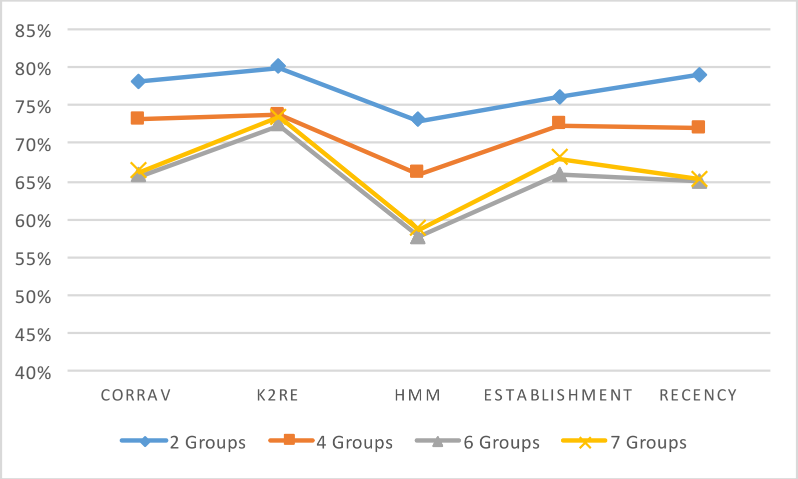


Figure 7.6. A Comparison of All Methods using the MAP Measure, on different Data Sample Sizes from the first dataset: 2 Groups, 4 Groups, 6 Groups and 7 Groups.

while the CorrAv method stays more robust in this sequential analysis. This is a consequence of how this method is conceived. This experiment confirms that the dynamic K2RE method, is robust against a higher number of meetings over time.

7.4 Discussion

By investigating the results of our evaluation presented in the previous section, we observe that:
The K2RE method clearly outperforms other methods as well as the HMM baseline in all experiments that we conducted, and is therefore the superior model in our benchmark.
The establishment effect is the second best method and it outperforms the recency effect.
The recency effect is clearly a weaker method also compared with the CorrAv method. However, we observed that in the case of some groups the recency effect was better than CorrAv and establishment effects in modeling the continuation of topics over time. By observing the establishment and recency effects perform very differently on data from different groups, and also by observing that the CorrAv method (which correlates with the average of the two aforementioned methods) always performs reasonably well, we believed that, if we design

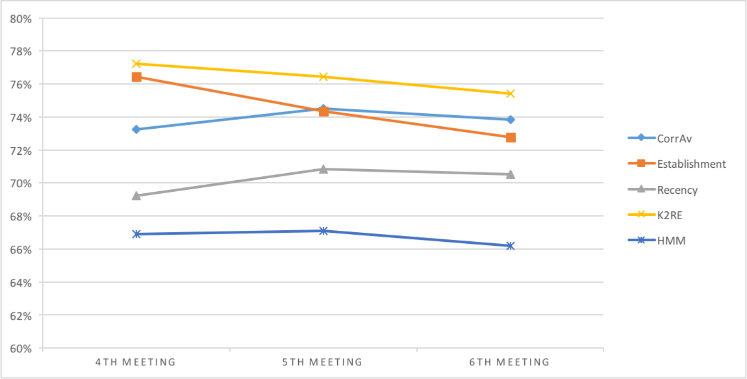


Figure 7.7. A Comparison of All Methods in terms of MAP, by sequentially adding meetings. This Experiment was performed on the second dataset

a model that could learn to perform a weighted average and adapt itself to the input data, this model is likely to outperform all other proposed models. Our results confirm that this belief and intuition was correct.
The K2RE method benefits from the fact that it distinguishes two different effects which are also opposite to one another, and weighs each effect differently over time, while the HMM baseline tries to find a generic solution without considering any different effects that maybe present in the data.

Our goal in designing the proposed models was to develop intuitive time-series algorithms for modeling user behavior, specifically regarding conversations and meetings. The broader vision and strategy that we tried to incorporate into the K2RE method, was that people often have the tendency to repeat the same behavior, but we also sometimes have the willingness to explore a different one. Hence, modeling this contrast between establishment and recency or, in other terms, repetition and exploration, was an important strategy that not only proved to outperform all other tested methods, but may also be useful in other domains and with other types of user data. The concept behind K2RE may be also used in personal assistants, to track user behavior over time and provide the user with the right information just in time the user might need them.

Finally, we would like to add that in Chapter 3 we have achieved results indicating that on average an individual who reviewed her past meetings using our memory augmentation tool, SILAS, could recall significantly more details of a meeting compared with when she did not use the tool.

Hence, we believe that further development of such technologies is effective in assisting people to recall their past important meetings.

7.5 Conclusions

In this chapter, we introduced the problem of predicting topics to be reviewed in preparation of one's next meeting to augment one's memory as a just-in-time IR approach. For this purpose, we proposed four different novel methods and compared them against an HMM baseline which has been extensively used in the literature for multivariate time series prediction tasks and is the state-of-the-art in a number of domains [104]. We showed through extensive experimentation that the dynamic K2RE method, that combines recency and establishment effects, significantly outperformed all other methods as well as the HMM baseline. The developed methods could be implemented as a part of a proactive memory augmentation system that aids people in their every day lives. The benchmark presented in this chapter could be a foundation for future studies and further development of such technologies.

There are a few interesting directions for future work. One interesting future work would be adapting the K2RE method to other user intent and context tracking domains and compare it with other methods in those domains. For example, analyzing datasets which current personal assistants such as Microsoft Cortana or Google Now gather (that track user behavior to anticipate their information needs) could be a possibility.

Chapter 8

Just-In-Time IR for Predicting Topics in Scholarly Papers

8.1 Introduction

Predicting topics that continue in a future time-slice of a sequential dataset can have applications in many differnt domains. Thus far in Chapter 7, we presented different models for predicting topics that continue in future meetings. We showed that the K2RE method outperformed all other tested models, maybe due to its evolutionary nature.

In this chapter we go beyond our meeting datasets and present a scalable version of the K2RE model and show it capabilities on a larger and temporally longer dataset of scholarly papers.

Scientific papers are a vehicle for advancing science and technological development. Discovering topics from scientific papers and analyzing their evolution over time is beneficial for making important decisions by governments, research organizations, funding agencies, and even researchers. As an example, research-funding organizations can adjust their granting policies based on insights produced by predictive models in order to favor topics that are trending and get increasing attention rather than those that are losing momentum and interest.

In this chapter, we use a publicly available dataset of papers from the Neural Information Processing (NIPS) conference which were published over 29 years. We propose a novel evolutionary method capable of predicting the topics of past time slices that would continue future time slices.

We compare the predictive performance of our method against two dynamic-topic-modeling baselines in terms of near-future prediction of continuing topics. The first baseline is the DTM by Blei et al. [33] which tracks the evolution of topics over time. DTM assumes that all topics are present in all the time slices of a sequential corpus of text. The second baseline is the dDTM [23] which modifies DTM by relaxing the assumption that a topic should be present in all the time slices. Thus, dDTM tracks the evolution of intermittent topics over time, hence the word "discrete" in the name. Both DTM and dDTM were extensively explained in Chapter 4 and for further details on these models we refer to Chapter 4. Due to the larger size of the NeurIPS papers dataset, in this chapter we can use these two models as baselines. It is noteworthy to mention that whether the topics extracted from the NIPS papers tend to be continuous or discrete, we use both DTM and dDTM as baselines to rigorously test our proposed model in the different possible scenarios. We aim to address the following research questions: How does DTM perform against dDTM? How does our model preform in comparison with the two baselines? Due to the large size of our dataset in this Thus, the contributions of our research presented in this chapter are as follows:

- We present a scalable version of the K2RE model (presented in Chapter 7) for predicting the continuing topics over time.
- We conduct an analysis of the dataset of NIPS papers to show different features of our novel method, and compare it with DTM and dDTM baselines.

8.2 Our Model

In this section, we present a sacalable version of K2RE for predicting topics that will continue in the future. Similar to the K2RE for effectively predicting the continuing topics over time, we devise various strategies which consider the *recency* and the *establishment* effects. The former is captured by increasing the weight of the most recent topics, whereas the latter assigns higher weights to the more established topics. Based on these two effects, we developed a dynamic method that combines recency and establishment measurements.

8.2.1 The SK2RE Method

We call our methodology *Scalable Kalman combination of Recency and Establishment (SK2RE)*. It combines the two effects of recency and establishment using a Kalman filter. In the following we first explain the two effects and then further elaborate on other details of our model.

Recency. The recency effect ranks the topics by assigning higher weights to most recent topics. Then, as an energy function it computes the correlation of each topic (whose continuation is to be predicted) with the vector of newly computed weights. We formally define the recency effect as follows: given the topics of the last n consecutive time slices of a sequential dataset, we would like to predict which topics continue in the $(n+1)_{th}$ time slice. Let V be the vocabulary of all words occurring in the first n time slices. We construct a word vector containing probability scores corresponding to each word in V. The assigned probability scores are higher for the words appearing in the most recent topics. Thus, first we compute LDA topics (this is explained in Section 8.3.1) from the first n time slices and then we compute the average probability of each word present in all topics according to the recency effect using the following equation:

$$P_{ref,Rec} = \sum_{n=1}^{n}\sum_{t=1}^{t}\sum\nolimits_{w_i \in t} \frac{P(w_i) * 2^n}{(n * t)} \tag{8.1}$$

where n is the sequence number of the time slice, t is the number of topics derived from each time slice, and w_i is a word from that topic. The 2^n is the rate with which higher weights are assigned to recent topics. The resulting word vector is an average representation of the probability of all the words present in all the n time slices.

Therefore, this effect assigns higher weight to a word which has occurred in the most recent time slice of a sequential corpus. We refer to the word vector where the probability of each word is computed with Equation 8.1 as the recency-reference vector.

Establishment. As for the establishment effect, given a vocabulary V made of all the words occurring in the first n time slices, we create a word vector containing probability scores corresponding to each word in V. In this case, the assigned probability scores are higher for the words which have persisted over time. For this purpose, we compute LDA topics (this is explained in Section 8.3.1) from the first n time slices and compute the average probability of each word present in all the topics according to the establishment effect using the following equation:

$$P_{ref,Est} = \sum_{n=1}^{n}\sum_{t=1}^{t}\sum\nolimits_{w_i \in t} \frac{P(w_i) * 2^{-n}}{(n * t)} \tag{8.2}$$

where n is the time slice sequence number, t is the number of topics derived from each time slice, and w_i is a word from that topic. The 2^{-n} is the rate with which higher weights are assigned to established topics and, as we can see, it is the opposite of Equation 8.1. It assigns weights to the words, in a way inverse to the recency effect.

Therefore, the word vector constructed by averaging all topics in all n previous time slices based on the establishment effect will be a representation of the average occurrence of each word in V, where the most established (i.e., persisting in occurrence) words have higher weights. We refer to the word vector where the probability of each word is computed using Equation 8.2 as the establishment-reference vector.

Combining Recency and Establishment. Now that we defined the recency and establishment effects, we explain how to combine them. Our model dynamically estimates the weights of each of the effects for each time slice and corrects itself over time by learning from the data. We refer to this method as SK2RE. It utilizes the Kalman filter [73] to estimate the recency and establishment weights over time. By measuring the dataset changing behavior in terms of establishment and recency, SK2RE adapts itself to the dynamics of the data over time.

In the following, we first present a general overview of SK2RE and then elaborate on its details. The components of the SK2RE model remain similar to Figure 7.1 presented in Chapter 7 for integrating the scores from the recency and the establishment effects.

The linear interpolation for a time slice t is defined as:

$$SK2RE_t = w_{E,t} * Score_{establishment} + w_{R,t} * Score_{recency} \tag{8.3}$$

where $Score_{establishment}$ and $Score_{recency}$ are computed by the establishment and the recency effects, respectively. Furthermore, $w_{E,t}$ and $w_{R,t}$ are establishment weights and recency weights computed by the Kalman filter at time t, such that: $w_{E,t}+w_{R,t} = 1$. This means that at each time slice t, each of the two effects will be given a weight that reflects its contribution to the future topics. The weights can be either equally assigned or they reflect the effect that dominates. The Kalman filter is always initialized by assigning equal probability of 0.5 to both $w_{E,t}$ and $w_{R,t}$ and then it dynamically updates the weights based on the learning from the data, according to the following system of equations:

$$f(n) = \begin{cases} X = Af_{t-1} + \varepsilon_t & t = 2, \ldots, T \\ z_t = Hf_t + w_t & t = 1, \ldots, T \end{cases}$$

where f_t is the system state at time t, A denotes the transition of the dynamic system from $t-1$ to t, H describes how to map state f_t to an observation, z_t (i.e., measurement), and both ε_t and w_t are mutually independent Gaussian noise variables with co-variances R_t and Q_t, respectively. The dynamic system evolves over time and updates itself proportional to the Kalman gain.

Now we describe the topic-linking module which is based on the dDTM [23]. As described in Chapter 4, the dDTM model tracks the evolution of intermittent topics that may occur discretely over time, such that a topic does not need to be necessarily present over all time slices. In the topic-linking module shown in Figure 7.1, we use a component of dDTM which links together similar topics over time. Such linking can be discrete or continuous under our model (i.e., a topic may be present over all time slices or it may skip some time slices). In particular, we use a Gaussian random walk in a Markovian state space model. The Markov assumption enforces probabilities of a hidden state at time t to be computed merely depending on the previous time slice and not based on all the previous states. We utilized this assumption to compute topic chains that capture the evolution of a topic discretely over time, so that two topics will be linked over two different time slices if they are similar according to the following criterion:

$$\beta_{t,k} | \beta_{t-m,1..k} \sim \mathcal{N}(\beta_{t-1}, \sigma^2 I) \tag{8.4}$$

where $\beta_{t,k}$ is topic k at time slice t, and $m \in (1,..,n)$ with n being the number of previous time slices and σ is the maximum variance allowed from the mean of a topic in the previous time slice. By assigning a small value to σ, our model links two topics that are highly similar. Furthermore, we use the Baum-Welch [28] algorithm to learn the forward and backward probabilities of the transitions among the topics. The model takes as input the topics from the first n time slices and computes their continuation in the $(n+1)_{th}$ time slice.

After linking similar topics over every two consecutive time slices, the topic-linking module computes the recency rate of the topics for time slice n which is the number of topics in the time slice $n-1$ that have been present in the time slice n divided by the total number of topics in the same time slice. This measurement is given as the observation matrix to the Kalman filter for each time slice. Subsequently, the Kalman filter computes the evolution of recency and establishment weights using the Kalman filter system of equations and, using Equation 8.3, a SK2RE reference vector is generated.

For the purpose of computing correlation between each topic and the SK2RE reference vector, we use the Pearson correlation metric that is $P_{X,Y} = \frac{COV(X,Y)}{\sigma_X \sigma_Y}$, where $P_{X,Y}$ is the Pearson correlation of two populations X and Y, COV is the covariance, and σ is the standard deviation. In our use case, the Pearson correlation indicates the level of correlation of each topic with the SK2RE reference vector. The intuition behind using correlation as an energy function is that, topics are by definition a set of words that depend on one another and a change in one word may cause changes in the probabilities of other words. According to this intuition we chose the Pearson correlation as an energy function for comparing word vectors. Finally, we require a threshold for distinguishing continued and discontinued topics. To determine an effective threshold we use *10-fold cross validation*. Hence, we split the dataset in 10 folds, iteratively leaving out a chunk of the data, and compute the threshold which minimizes the Mean Squared Error (MSE) of prediction on the remaining folds. Then, using the computed threshold, we evaluate the left-out fold.

8.3 Evaluation

In this section we present our experimental setup, including a description of the dataset, followed by the evaluation of our approach.

8.3.1 Dataset Description

Our dataset consists of all the papers of the Neural Information Processing Systems (NeurIPS) conference published between years 1987 and 2015. Therefore, our dataset is spread over 29 years. The total number of papers that are included in our dataset is 5993. We obtained this dataset from the Kaggle website[1].

Topic extraction: The dataset is sorted chronologically and divided into time slices of fixed size (one year). We treat every paper as a document and applied LDA to extract the latent topics from each time slice. Since the number of topics (K) discussed in two different time slices might vary, it is important to estimate the number of topics per time slice. For this purpose, we went through the same model selection process explained and visualized in Chapter 7.

Labeling: In our prediction tasks we assume that given the topics of the first n time slices (the first 28 years of the dataset) we would like to predict those that will persist in the $(n+1)_{th}$ time slice (the 29th year of the dataset). We carried out the labeling process semi-automatically. At first, using a k-nearest-neighbors implementation, for each of the topics from the first 28 time slices we identify the top five neighboring topics in the 29_{th} time slice to simplify the annotation task.

Then, given a topic from the first 28 time slices and its top five neighbors in the last time slice, we asked three human assessors (who were domain experts) to determine whether or not the topic was a continuation or not. This was done for all the topics of the last 28 time slices to have a ground truth for the predic-

[1] https://www.kaggle.com/

tion task. By aggregating the votes of the three human assessors each topic was labeled.

The assessors were given instructions on how to label the topics as ‘continued’ and ‘not continued’. These instructions include putting more emphasis on the top 20 words in each topic to take a decision. That is due to the fact that the users of our system would look at the top words of each topic to understand it. As a result, our dataset consists of 839 topics out of which 305 topics are labeled as continued and 534 topics are labeled as discontinued.

8.3.2 Experimental Results

In this section we present the evaluation of our method against the state-of-the-art dynamic topic modeling approaches. The two baselines are both Bayesian networks that can connect similar topics together over time and thus, can be used to predict future topics. We conduct two experiments one involving human annotations, and the other only based on Euclidean distance between a predicted topic and its closest neighbor in the last time slice.

First experiment. We evaluate our method using standard information retrieval evaluation metrics, namely, precision, recall, F_1 measure and Mean Average Precision (MAP).

For performing the continuing topics prediction task using DTM and dDTM we compute the log likelihood of each topic (whose continuation has to be predicted) under the respecting model. Then we normalize the log likelihood scores to compute a score between 0 and 1. Then similar to the SK2RE case we use 10-fold cross validation to compute the performance of all the models.

The dDTM model estimates the number of topic chains automatically. However, both dDTM and DTM require that we manually set the number of topics per each time slice. As we explained in Section 8.3.1, we estimated the number of topics in each time slice using model selection when building our dataset. Since the average number of estimated topics per time slice was very close to 30 we initialize both DTM and dDTM by setting their number of topics to 30.

Figure 8.1 shows an interpolated precision-recall diagram. The figure shows that our novel model outperforms the baselines in terms of the precision-recall curves and MAP.

Furthermore, Table 8.1 shows a comparison between all the methods with respect to precision, recall, F_1 measure, and MAP. Our results show that the SK2RE method outperforms the two other state-of-the-art dynamic topic model used as baselines.

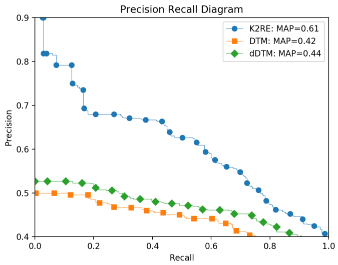


Figure 8.1. Precision and Recall of our model against DTM and dDTM.

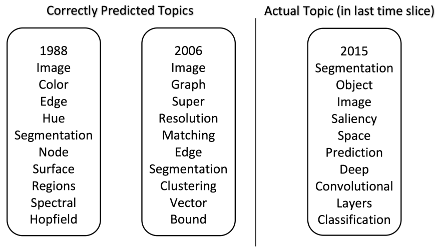


Figure 8.2. An example of a continued topic against the ground truth

Table 8.1. Comparison of approaches: Precision, Recall, F_1 measure, and MAP.

	Precision(%)	**Recall**(%)	**F$_1$**(%)	**MAP**(%)
SK2RE	61.56	84.78	71.33	61.53
DTM	50.03	85.66	63.17	42.84
dDTM	49.95	89.51	64.12	44.38

Second experiment. As a second experiment, we use a more objective evaluation without using human annotated data. For this purpose, we compute the average Euclidean distance between each topic that was predicted by our model as “continued” and its closest neighboring topic in the last time slice. By doing so, we can compare our model against the other models in terms of how correctly they could distinguish between those topics that continued and the ones that they did not. In this case the lower the distance measure the better each model has performed. Furthermore, by repeating the same procedure for the topics that did not continue we can again compute an average Euclidean distance for how similar they are to the actual topics of the last time slice. In this case the higher the distance the better a model has performed.

We present the results of this experiment in Table 8.2. As we can see from

Table 8.2. Comparison of approaches based on distance with ground-truth topics

	SK2RE	**DTM**	**dDTM**
Ave. Euclidean dist. (Continued)	0.0305	0.0543	0.0493
Ave. Euclidean dist. (Discontinued)	0.1026	0.0744	0.0850

the results presented in Table 8.2, our evolutionary SK2RE model achieves higher similarity scores to the actual topics in the last time slice compared with the two baselines. As shown in the table, the average Euclidean distance to the ground truth (topics) in the case of continued topics is lower than that of the baselines. Moreover, the distance is higher in the case of discontinued topics as predicted by the SK2RE method. This experiment, confirms that the predictions made by the SK2RE method are objectively closer to the ground truth as compared with the two baselines.

Qualitative example. Finally, we show a qualitative example of two topics which were predicted by SK2RE to continue in the last time slice against the actual topic appearing in that time slice. We chose a topic about "image processing" (shown in Figure 8.2) and we observed that *image processing* is an important topic for the NIPS conference over the years. Image segmentation based on color, changes to image super resolution and then ten years later into deep convolutional neural networks, object detection, and image segmentation. As a further example on how our model correctly predicted a discontinued topic, we present the top-10 words of a randomly chosen topic from the year 1999: "spatial, temporal, localization, space, vector, belief, sequence, probability, robot, state". As we can see, this topic is mostly related to the navigation and localization of a robot. Our analysis of the data shows that the word "localization" did not occur in the 2015 time slice, hence such topic disappeared. A funding agency or a research organization with access to such insights (e.g., which topics continue and what are the main themes of a continuing topic over time) can make informed decisions and strategic planning. Indeed, by looking at the image processing topic in 2014 we might be able to come up with more specific and detailed directions of research about 2015 (e.g., use of convolutional neural networks) but knowing the general trend and how past research is evolving into the future is also of strong importance.

8.4 Conclusions

In this chapter, we introduced an evolutionary model capable of predicting topics that continue in the next time slice in a sequential corpus of documents. For this purpose we modified the K2RE model presented in Chapter 7 to a more scalable model suitable for large datasets. Our results showed that our method outperforms the state-of-the-art dynamic topic models in the prediction task. Our evolutionary SK2RE model can learn the changes in the data over time and adapt itself to the changes. We used a corpus of scholarly papers to show the effectiveness of our model.

As a future work we plan to extend our evolutionary model to other data modalities and domains. As an example, we believe that our model can be used in recommender sytems [5, 21] for tracking user intent and context over time. Such system can anticipate users' information needs over time.

Chapter 9

Just-In-Time IR: Predicting the Topic of Your Next Query

9.1 Introduction

Proactive search technologies aim at modeling the users' information seeking behaviors for a *just-in-time information retrieval* and to address the information needs of users even before they ask. Modern virtual personal assistants, such as Microsoft Cortana and Google Now, are moving towards utilizing various signals from users' search history to model the users and to identify their short-term as well as long-term future searches. As a result, they are able to recommend relevant pieces of information to the users at just the right time and even before they explicitly ask (e.g., before submitting a query). In this chapter, we focus on another user data modality and propose a novel neural model for JITIR which tracks the users' search behavior over time in order to anticipate the future search topics. Such technology can be employed as part of a personal assistant for enabling the *proactive* retrieval of information. Our experimental results on real-world data from a commercial search engine indicate that our model outperforms several important baselines in terms of predictive power, measuring those topics that will be of interest in the near-future. Moreover, our proposed model is capable of not only predicting the near-future topics of interest but also predicting an approximate time of the day when a user would be interested in a given search topic.

With the rapid proliferation of web search as the primary mean for addressing the users' information needs, search engines are becoming more sophisticated with the purpose of improving the user experience and of assisting users in their search tasks more effectively. As an example, with the increasing and ubiquitous usage of mobile devices [7, 8], it has become more important for search engines to offer also *JITIR* [107] experiences. This means retrieving the right information at just the right time [59] to save users from the hassle of typing queries on mobile devices[**?**].

The notion of "personalized search" [130] has shown to be effective in improving the ranking of search results. However, such personalization comes at the cost of lower speed, which in some cases might even cause the retrieval of the results only after the user search session has ended. Moreover, possible discovery of newly available content related to a previous search is another application of JITIR models for presenting results to a user at a future time.

As a result, researchers have focused on improving search personalization with respect to not only the retrieved content but also the user's habits (e.g., *when and what* information the users consume). While such models can benefit desktop users in better addressing their information needs at just the right time, they are essential on mobile platforms. Indeed, Microsoft Cortana and Google Now aim at offering a proactive experience to the users showing *the right information before they ask* [127].

As pointed out by Agichtein et al. [2], knowing the user's information needs at a particular time of the day allows to improve the search results ranking. For example, the search results can be personalized based on the specific search task (of a given user at a given time) rather than based on the more general information of user interests which have been inferred by the entire user's profile. This would also support users in resuming unfinished search tasks (e.g., if a search is likely to be continued one can save the results already found for a faster or more convenient access once the task is resumed).

Figure 9.1(a) shows the behavior of a randomly selected user from our dataset in issuing search queries related to a topic about `movies` over differnet week days. For example, the user might have searched the word "imdb" along with the title of a movie. As we can see, the user exhibits a higher tendency to search for movies in the afternoons and evenings as well as on Saturdays. Hence, we can infer that the user is interested in watching movies on Saturday evenings and thus it is likely that her queries are related to movies. Moreover, as shown in Figure 9.1(b) a user changes search behavior over time. To address such changes in search behavior we propose a dynamic memory system.

Addressing the near-future information needs of the users has been also studied in the context of personal assistants, such as Google Now, Microsoft Cortana, or Apple's Siri and in the context of memory augmentation in meetings [18]. These systems offer proactive experiences [123] that aim to recommend useful information to a user at just the right time.

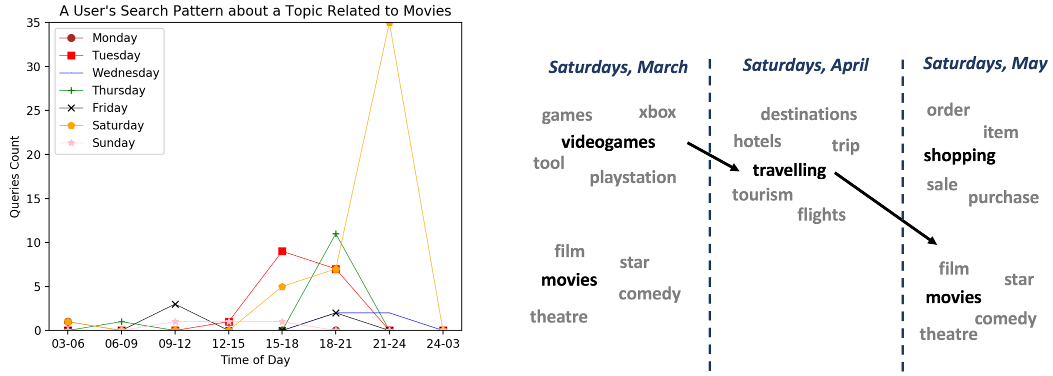


Figure 9.1. (a) The number of queries about `movies` submitted by a randomly selected user. (b) Evolution of user-search patterns on different Saturdays.

In this chapter, we focus on yet another application of JITIR, predicting the topics of the users' future search queries. Specifically, we propose a model which predicts the topic of the search queries submitted by the users in the next 24 hours. Moreover, our model leverages the user's behavior patterns over time in order to predict the topic of the user's query on a specific weekday (e.g., Mondays, Tuesdays) and at an approximate time of the day. The main contributions of our research presented in this chapter are:

- We propose a time-series model based on neural networks to predict the topic of near-future queries of users.
- Our model is equipped with a dynamic memory learning users' behavior over time. This memory evolves over time when the search patterns change. We demonstrate that our dynamic memory architecture is beneficial as it increases the prediction performance. Further, we believe that this model could be useful in other domains that involve temporal data.

9.2 Research Goals

We can summarize the goals of our work as follows: (1) predicting the topics of future search queries of a user, and (2) predicting the day of the week and the approximate time of the day when the topic will be queried by a user.

Given the search history of each user u in the last n consecutive time slices, as well as a set of corresponding query topics $\mathbb{Z}$, we aim to predict the topic $z \in \mathbb{Z}$ of the query of user u in the $(n+1)_{th}$ time slice.

For achieving this, we first model the search tasks as topics using LDA. Then, leveraging a time-series model we discover the latent patterns in search tasks and predict the continuation of a search task in the near-future. In other words, we aim at predicting the topics of the user's future queries. Such technology will enable the proactive retrieval of relevant information in a JITIR setting. However, estimating the time of the day when a user would access a particular content is the second piece of the puzzle in order to recommend content more precisely and more effectively. Thus, our second goal consists in correctly predicting when (day and time of the day) the users will consume what content (topic) knowing the users' habits in requesting the various topics at the different times.

9.3 Query Topic Prediction Model

We now present our novel time-series evolutionary model for predicting the topic of a user's near-future queries. The model is based on the notion of reinforcement learning so that it adapts itself to the data over time and corrects itself. We formally define our model as a function f which takes as input the search history of users and predicts which topics occur in the near future.

The model consists of a dynamic memory in the form of a word embedding connected with a *Bi-directional Long Short Term Memory* (BiLSTM) [118] used to capture the behaviour of a user over time. The dynamic memory implements two different effects of persistence and recency. At each point in time, based on the possible changes in the input data, it updates the word vectors to provide as input to the BiLSTM network.

In the following, we first describe the dynamic memory system in Sections 9.3.1 and 9.3.2. Then, we present the BiLSTM network in Section 9.3.3.

9.3.1 A Dynamic Memory based on Word Embeddings

Our intuition behind the design of such memory model is that people often show similar behavior over time (i.e., persistence) but they also have a tendency to explore new things (i.e., recency). As a result, over a time-line people may show very different behaviors and the model should be capable of capturing them in order to accurately anticipate the users' future behaviors [153]. Therefore, we believe that dividing the temporal input data into a number of time slices and weighting them based on identified patterns in the data is important.

The dynamic memory is based on the word2vec word embeddings. Throughout this chapter whenever we use the term word2vec, we refer to a Skip-Gram with Negative Sampling (SGNS) word embedding model. Levy et al. [86] showed that the SGNS method is implicitly factorizing a word-context matrix, whose cells are the PMI of the respective word and context pairs, shifted by a global constant. They further elaborate that word2vec decomposes the data very similar to Singular Value Decomposition (SVD) and that under certain conditions an SVD can achieve solutions very similar to SGNS when computing word similarity. Apart from scalability and speed, SGNS is capable of removing bias towards rare words using negative sampling. Other than the few differences, at the concept level both SVD and SGNS are very similar. They both build a word-context matrix for finding similarities between words.

Based on these principles we propose a novel and effective method for integrating multiple word2vec memory components where each is trained with data from a different time slice of the input data. Let $m_t \in \vec{M}$ where m_t is a word2vec memory trained on data form time slice t and $\vec{M}$ is a vector of all word2vec models. Instead of using only one single memory to capture the global patterns in the dataset, we propose to use a different word vector from model m_t to represent time slice t where $t \in 0, 1, \ldots, n$. Then, we integrate all these word vectors into one final vector. Therefore, a temporal dataset of web search queries can be divided into n different time slices, and one word2vec memory m_t is trained for each time slice. We assume that all the vectors have the same embedding dimensions, so given two vectors m_t and m_{t+1} we can combine them using the *orthogonal Procrustes* matrix approximation. Let W^t be the matrix of word embeddings from m_t which is trained on data at the time slice t. We align across m_t and m_{t+1} which are derived from consecutive time slices while preserving the cosine similarities by optimizing:

$$argmin_{Q^TQ=I} ||W^tQ - W^{t+1}|| \tag{9.1}$$

Matrix Q is described in the following. We note that this process only uses orthogonal transformations like rotations and reflections. We have solved this optimization problem by applying SVD [117].
We can summarize the steps of the approach as follows:

1. The vocabulary of the resulting word vectors from the two time slices are intersected and the ones in common are kept. We note that due to our definition of an active user as well as the way we map queries to unique user, topic and time identifiers, vocabulary remains the same over all time slices (see Section 9.4.1).

2. We compute the dot product of the two matrices (for doing so, we first transpose one of the matrices).

3. The SVD of the matrix resulting from the dot product is computed. The result consists of three factorized matrices commonly known as U, the diagonal matrix S, and the matrix V.

4. We compute the dot product of U (left singular matrix) and V (right singular matrix) to have as resulting matrix Q. Since S contains information on the strength of concepts representing word-dimension relations which are not needed here as they are not modeled in word2vec, we discard the matrix S. The existence of the S matrix is also one important difference between SVD and word2vec, which word2vec does not compute.

5. Finally, we compute the dot product of Q and the embedding matrix W^t. For further detailed information we refer to [117] where the *orthogonal Procrustes* approximation using SVD is described.

We repeat the process of model alignment for all n word2vec models spread over the entire time-line.

9.3.2 Modifying the Dynamic Memory

Now that we have explained the process of combining different word2vec models, in this section we explain how our proposed model takes into account the recency and persistence of the searching behaviors of users over time. Before combining W^t and W^{t+1} which are word-dimension matrices from two word2vec models (as described in Section 9.3.1), we modify each matrix based on the following effects:

Recency Effect. It modifies the strength of word embeddings by assigning higher weights to the word vectors observed in the most recent time slice. We formally define the recency effect as follows: given the query topics of the last n consecutive time slices of a sequential dataset, we would like to predict which query topics continue in the $(n+1)_{th}$ time slice. By assuming a vocabulary v of all the words occurring in the first n time slices, we construct a word vector containing the probability scores corresponding to each word in v. The assigned probability scores are higher for the words appearing in the most recent time slices. After modifying the word vectors, we then perform alignment of models as described in Section 9.3.1. According to the recency effect presented in the following equation we modify the word embedding matrices W^ts by $P_{Rec} = \sum_{n=1}^{N} \sum_{w_i \in W^t} P(w_i) * 2^n$, where n is the time slice number, P indicates probability, and w_i is a word from the word embedding matrix W^t. The 2^n is the rate with which recent word vectors are assigned higher weights. The resulting constructed word vector is an average representation of the probability of all the words present in all the n time slices.

Therefore, this effect assigns higher weight to a word which has occurred in the most recent time slice of a sequential corpus. We refer to the word vectors which are computed by the recency effect as the *recency matrix*.

Persistence Effect. Given the word embeddings of the last n consecutive time slices, we would like to predict which query topic continues in the $(n+1)_{th}$ time slice. Given a vocabulary v of all the words occurring in the first n time slices, we construct a word vector containing the probability scores corresponding to each word in v. The assigned probability scores are higher for the words which have persisted over time. We compute the updated probability of each word according to the persistence effect using $P_{Pers} = \sum_{n=1}^{N} \sum_{w_i \in W^t} P(w_i) * 2^{-n}$, where n is the time slice number, P indicates probability, and w_i is a word from W^t. The 2^{-n} is the rate with which the higher weights are assigned to persistent words. Therefore, the more persistent words (i.e., persisting in occurrence) have higher weights, and we refer to the word vector computed with the persistent effect as the *persistence vector*.

Combining Recency and Persistence: Recency and persistence scores are combined in a linear interpolation to modify the original word embedding matrix. The linear interpolation at the time t is defined as:

$$EmbeddingMatrix_{w,t} = w_{P,t} * Score_{Pers.} + w_{R,t} * Score_{Rec.} \tag{9.2}$$

where $Score_{Pers.}$ and $Score_{Rec.}$ are computed by the persistence and the recency effects, respectively. Furthermore, $w_{P,t}$ and $w_{R,t}$ are persistence weights and recency weights computed at each time slice. They have the following relation and are learned from the data: $w_{P,t} + w_{R,t} = 1$. This means that at each time slice t each of the two effects corresponds to a weight. The weights can be equal (i.e. when the effects have the same intensity) or different, but their sum would be always 1.

The weight $w_{R,t}$ is then computed as square root of the sum of the difference in the number of occurrences of each query topic compared with the previous time slice divided by the number of all the queries at the same time slice. Subsequently, $w_{P,t}$ is computed based on $w_{P,t} + w_{R,t} = 1$. As a result, the dynamic memory evolves over time and updates itself proportionally to the rate of the changes in the data.

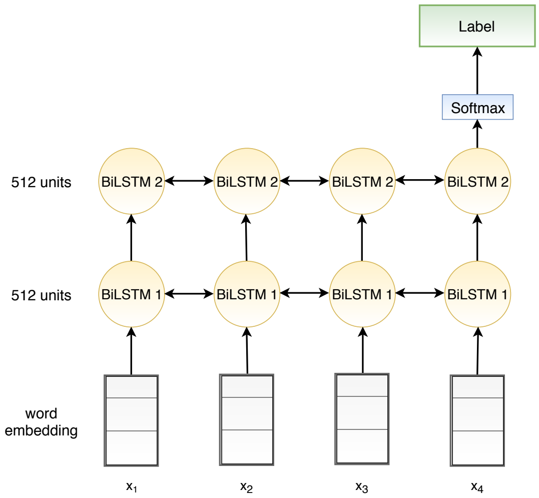


Figure 9.2. The architecture of our proposed model. x_i stands for input at time step i

Finally, we map each query to a topic using LDA. Further details of this process are explained in Section 9.4.1. We specify each query with the ID of the user who submitted it along with the given week day and time bucket (i.e., which is an approximate time of day) of the query. Then, we train n word2vec SGNS models on this data in order to train the dynamic memory. For word2vec, we use embedding size of 300, without discarding any of the input words. The result will be n word embedding matrices derived from n time slices which are aligned and combined into one word embedding matrix which is given as input to the BiLSTM.

9.3.3 Bi-directional Long Short Term Memory (BiLSTM)

We train the BiLSTM network using the word embeddings of the dynamic memory. We use the BiLSTM neural network as function for generating a sequence of events given an input query. In other words, we aim at modeling the sequence of observations (i.e., the searches about certain topics) in a time-series fashion. Thus, given the user ID, the future week day and the time bucket, the model will predict the topics of the near-future queries.

As shown in Figure 9.2, the architecture of our model consists of word embeddings from the dynamic memory provided as input to the BiLSTM network. We model each query in four recurrent time steps in order to predict the topic of near-future queries along with their weekday, and approximate time of the day (i.e., we refer to it as the time bucket in dataset description). On the other hand, when we want to only predict the near-future topics without specifying their approximate time of the day, we train the same network with two recurrent time steps (i.e., one for the user ID and the other for the topic). In both cases we set number of word2vec models to six (n=6) to model almost every two weeks with one word2vec model. Furthermore, our model includes two fully connected BiLSTM layers, with each layer containing 512 cells or units. We applied a SoftMax layer to the final output from the BiLSTM networks.

Our intuition behind this architecture is to first find a collaborative generalization of patterns of users in issuing queries about certain topics at particular points in time by using the dynamic memory based on the word2vec model. Then, using the BiLSTM neural network we leverage the local dependencies between certain behaviors in a temporal manner. The BiLSTM network serves as a time-series model that determines the occurrence of a future event (i.e., a future query's topic) by modeling the sequences of events (i.e., sequences of topics).

9.4 Evaluation

9.4.1 Dataset Description

In the experiments, we use the publicly available AOL query log [100] which has been used in other research works on query caching and search result personalization. It consists of about 36M query records and spans a period of three months (from March to May 2006). Each record consists of the anonymous ID of the user, query keywords, timestamp, and rank and URL of the clicked result.

Our goal is to predict the topics of the future queries issues by a user, hence we selected those users who have a high number of queries. Formally, we define *active users* those who have searched at least one query every week and over a span of three months have issued at least 1,000 queries. From this set which contains 1,197 active users, we randomly selected 500 users to train and test our proposed model as well as the baselines. The query log made of the queries issued by these 500 users consists of 755,966 queries.

Training and testing data. Our experiments aim at predicting the topics of the future queries searched by a user, so we sorted the query log by time and split it into training and test sets. The training set is used for learning the topics of interests of a user, while the test set to check the prediction performance. For our experiments, the test set consists of the queries issued in the last 24 hours (which results of 10,848 queries) while the rest of the queries is used for training.

Modeling search tasks as topics. In order to model the topics of the search tasks we used the LDA topic model [34]. The latent topics discovered by LDA can be used as a ground truth for carrying out our experiments. In particular, we utilize the LDA model to extract the topics of various search tasks. Since the search queries are short and lack context, we decided to enrich them with the content of clicked pages. In more detail, given the queries from the training set and the URL of their clicked results, we gathered the content of $351,301$ unique web pages. We treat each query and the text of its corresponding clicked result as a document, and we run LDA over the collection made of these documents. LDA returns K list of keywords representing the latent topics discussed in the collection. Since the number of topics (K) is an unknown variable in the LDA model, it is important to estimate it. For this purpose, similar to the method proposed in [24, 58], we went through the same model selection process explained and visualized in Chapter 7. The training of each LDA model in the case of our dataset takes nearly a day, so we could only repeat it for a limited number of K values. In particular, we trained the LDA model with K equals to 50 up to 500 at steps of 50, and the optimal value was 150.

Labels for Predicting the approximate time. The search queries have timestamps, so we could extract the day of the week and the time of the day when they were issued. We divide the 24 hours into eight time buckets of three hours each. Each time bucket represents an approximate time of the day and we can use this for predicting the approximate time of the day when a query topic will appear. Hence, given a user, our ultimate purpose is to predict the right query topic and when it will be requested (i.e., the week day and the time bucket).

9.4.2 Evaluation Metrics and Baselines

Evaluation Metrics. We performed a rigorous testing of our proposed method and compared it against several baseline methods. For our evaluation, we used the standard information retrieval evaluation metrics: precision, recall, and F_1.

Baseline Methods. Since our proposed method is based on a collaborative filtering principle, we chose as baselines the following top-performing techniques:

1. ***Probabilistic Matrix Factorization (PMF)*** is a model for collaborative filtering that has achieved robust and strong results in rating prediction [114].

2. ***Non-negative Matrix Factorization (NMF)*** can analyze data with a high number of attributes [83]. It reduces the dimensions of data by converting the original matrix into two matrices with non-negative values.

3. ***User-based K Nearest Neighbours (userKNN)*** is another popular method which uses similarities among the users' behaviours to recommend items to users.

4. ***SVD++*** is a collaborating filtering method, where previously seen patterns are analyzed in order to establish connections between users and items [77]. The approach merges latent factor models that profile both the users and the items with the neighborhood models that analyze the similarities between the items or between the users.

5. ***TimeSVD++*** is one of the most successful models for capturing the temporal dynamics and has shown strong results in various prediction and ranking tasks which seek to model a generalized pattern over time [78]. The regularization parameter and the factor size are selected using a grid search over $\lambda \in \{10^0, \dots, 10^{-5}\}$ and $k \in \{20, 40, 80, 160\}$.

6. ***BiLSTM+w2v*** we also add as a baseline our own model with only one word2vec model trained as input (i.e., see n=1 in Table 9.2).

9.4.3 Experimental Results

We conduct two experiments, one for comparing our model against baselines in terms of near-future query topic prediction, and the other for analyzing the effect of the number of trained word2vec models on the final predictive performance. **First experiment.** The aim of our first experiment is predicting the topics of the queries issued by a user in the next 24 hours. Table 9.1 reports the results of our approach compared to the baselines. We observe that our method outperforms all the baseline models in terms of predicting the topics of one's queries in the future 24 hours with statistically significant improvement. We averaged the prediction results over the 500 users of our sampled data. As a result of this experiment, we could observe that our model is superior in predicting the topics of future queries compared to the other collaborative-filtering baselines.

Our proposed model features incorporating some principles that we believe have caused the superiority of our model. First, the dynamic memory not only learns users' search behavior but also considers the temporal dimension when modeling data. Furthermore, our model uses the recency and persistence effects and adjust itself to the data by measuring the behavior of the data and subsequently updating itself when needed. None of the baseline models, despite being powerful models, can model such complexities.

This could translate into better understanding of the users' needs and behavior which could be utilized for enhanced personal and timely assistance.

Second experiment. In the second experiment, we would like to investigate whether or not running and combining different word2vec models can improve the performance compared to one trained word2vec model. In particular, we divided our input data into chunks (e.g., weeks) and trained several word2vec models over them. Then, we compared the performance of one word2vec model trained over the whole time-line against the performances achieved with different numbers of word2vec models. We started by only having one word2vec model up to twelve models (one for each week of our dataset).

The results of this experiment are reported in Table 9.2 and show that training the word2vec model over different time slices performs better than having only one word2vec model trained over the entire dataset Moreover, we could observe that increasing the number of models allows to gain higher performance, however, after some point the performance reduces. We can conclude that training several models is better than one, but the number of models should be chosen depending on the application. We could observe that the best results can be achieved training the models with roughly two weeks of data.

Our research goal was to design an intuitive time-series method for modeling the user behavior, specifically regarding search queries. The broader vision and strategy that we tried to incorporate into the model was that the users have the tendency to repeat the behavior (e.g., searching about the same topic in a sequence), but they also have consistent behaviors (e.g., searching for the same topic every Saturday night). Hence, incorporating these two dimensions into our model helped to improve the prediction performance. The concept behind our model may also be used in a personal assistant environment for modeling other

Table 9.1. A comparison of our proposed method against the baselines.

	Precision	Recall	F_1
PMF (%)	12.53	31.03	19.78
NMF (%)	13.65	35.11	21.23
UserKNN (%)	14.20	38.06	22.09
SVD++ (%)	12.60	30.43	20.20
TimeSVD++ (%)	28.46	14.62	20.00
BiLSTM+w2v (%)	34.28	36.04	35.12
Our Model (%)	48.19	38.44	42.77
Our Model+time prediction (%)	26.23	34.41	29.77

Table 9.2. A comparison of the prediction performance varying the number of word2vec models in terms of F_1 (n=1 means one model trained over the whole dataset, etc.).

	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10
24 h prediction(%)	35.12	38.63	39.14	41.56	42.93	42.77	41.46	40.34	40.52	40.12
24 h + time-bucket prediction(%)	28.64	29.32	29.57	29.83	30.21	29.77	29.40	29.43	29.21	29.06

types of data, tracking the user behavior over time and providing the user with the right information just-in-time the user might need it. Envisaging that a system can correctly predict the topic of your near-future query more than 40% of times among all possible options (i.e., in this case 150 topics) while also predicting the time bucket when you will show interest in that topic and presenting relevant information or targeted ads to you even before you have started searching on that topic is a very interesting result.

9.5 Conclusions

In this chapter, we addressed the problem of predicting topics of future queries for JITIR. For this purpose, we proposed a novel method and compared it against six baseline methods which have been extensively used in the literature for temporal and non-temporal collaborative filtering. We showed through experimental results that our method, generalizing the users' behavior and modeling the temporal recurrent patterns, outperforms all the baselines. The developed method could be implemented as a part of a proactive search system that aids people in their every day lives.

One interesting future work would be adapting our method to other domains. For example, analyzing various data modalities gathered by current personal assistant tools such as Microsoft Cortana could be an interesting direction.

Part III

Thesis Conclusions

Chapter 10

Conclusions and Future Work

10.1 Conclusions

In this dissertation, we thoroughly investigated personal assistant systems that can (1) extract useful information from unstructured user data and present them to the user in an easy-to-understand summary, and (2) anticipate users' information needs proactively.

In the first part of this dissertation we focused on the first goal mentioned above. The research contributions we presented in this part and our findings were:

We investigated a variety of topic models and different ways of using them. In a user study we found out that our PMI-based method that preserves the order of topic occurrences is more understandable by an average person. Furthermore, we summarized daily social interactions and meetings of users in the form of event snippets as a part of SILAS. Using the dDTM topic model, we also enabled tracking the evolution of topics over time, such that a user would be able to connect topic threads with people they came into contact with over time. Additionally, we explored abstractive summarization of meetings in the form of natural language. We showed that our proposed summarization model can outperform the state-of-the-art solutions and together with the proposed transfer-learning approach, can achieve superior performance against the baseline models.

To conclude we developed methods for summarization of social interactions and meetings, that could perform summarization by extracting topics, order-preserved topics, topic chains indicating evolution over time, as well as natural language summaries producing condensed versions of long documents presented in Chapters 3, 4, and 5. The content of these chapters are supported by [13, 17, 23].

We investigated the effect of topics extracted from conversation transcripts paired with images taken by a wearable camera presented to a user in the form of memory cues. We conducted two user studies, one focused on recalling past events using the lifelog of one participant, and the other meeting log of five groups of at least two individuals. The results of these two user studies corroborated that such memory cues are effective means for augmentation of human memory in recalling past events. However, we also faced challenges in conducting in-the-wild user studies counting on the participants' commitments to abide by the rules of the user study that we explained to them. Due to such noise and other reasons explained in detail in Chapter 3, although our results showed the effectiveness of the memory cues on the majority of the participants, we could not achieve better results [17]. We believe that increasing the number of participants for a future user study may decrease the effect of such noise and enhance the overall results.

The topical focus of one's daily conversations might change suddenly and is highly unpredictable. The appearance of a topic in one's daily conversations may be highly intermittent appearing on some days but not being discussed on other days. As discussed in Chapter 4, the DTM algorithm assumes that each topic is present on all time slices of a sequential dataset This assumption makes the DTM not a feasible choice for most daily conversation logs, daily news streams, and social media content. To address this problem we introduced the dDTM topic model showing its capable of tracking the evolution of intermittent topics over time.

Abstractive summarization of text is considered as a sequence-to-sequence problem where a sequence of words in a source document is mapped to a shorter sequence of words of a summary. Our novel topical attention mechanism is the first of its kind model that brings the customization capability to neural summarization models. The topical attention mechanism can effectively include certain topics in a summary or exclude them from it. Therefore, our solution is the first method for customizable abstractive summarization of text documents using neural architectures.

One of the main challenges behind data-driven abstractive summarization of meeting transcripts is the small sizes of the existing datasets. In this thesis (Chapter 5) we presented a transfer-learning approach that increases summarization performance by a huge margin. To the best of our knowledge, this is the first data-driven neural architecture for abstractive summarization of meetings data.

In summary, in the first part of this thesis we investigated various methods for producing summaries of conversations and meetings for personal assistance. These summaries can be presented to a user in different settings including JITIR for proactive assistance. In the second part of this dissertation we focus on JITIR models that anticipate users' information needs proactively. These models predict a user's near-future interests and thus present relevant and desired content to the user without any explicit queries or requests. The main research contributions that we focused on in the second part of this thesis and our findings were:

In Chapter 6 we conducted a feasibility study along with experimentally testing a model based on HMMs for prediction of memorable/ non-memorable conversation segments [15]. Our results showed that the EDA signal as well as the sentiment expressed in a conversation are effective means for prediction of memorable/ non-memorable segments of a conversation. We empirically showed that prediction of forgetting in conversations is feasible and is a viable future research direction.

In order to prepare a person for a future meeting, in Chapter 7 we proposed several time-series models that predict conversation topics that would continue in the next meeting [18]. Our proposed time-series algorithms model the recency effect, the establishment effect and the average effect. We also proposed an evolutionary model that can model all the three mentioned effects by adapting to the behavior of the data and find out what combination of these three effects can better interpret the data. Our experimental results showed that the establishment effect can outperform the recency effect.

Moreover, our K2RE evolutionary model is capable of outperforming all other models as well as a HMM baseline in terms of predictive performance. All our proposed models, compute context information for each word in a conversation using a GMM.

In Chapter 8 we tested the various components of the K2RE model on a large dataset of scholarly papers and found out that the GMM component is the most computation-intensive component of the model. We then removed the GMM component and instead used the recency and establishment vectors and combined them in order to create a reference word vector. We then used the reference vector to predict topics that continue in the next time slice of a sequential dataset. Our results showed that the new SK2RE method is highly efficient and can outperform the DTM topic model in terms of future topics prediction [24].

In order to illustrate that JITIR modeling can be extended to other data modalities for personal assistance, in Chapter 9 we designed a study for predicting the topic of one's next search query submitted to a search engine [25].

The benefit of such research is retrieval of desired information for a user without any explicit queries by the user as a proactive personal assistant system. To obtain this goal, we relied on the successfully tested recency and establishment effects, and proposed a deep learning architecture capable of modeling collaborative patterns between users and predict the topic of each user's next query. Our experimental results showed that our model outperforms several important collaborative filtering baselines in terms of predictive performance.

Time-series models have application in tracking changes of temporal signals over time. However, when temporal signals become multi-variate and more complex, there is a need for sophisticated time-series modeling that can track changes in complex patterns over time. Such complex patterns can be presented in the form of matrices presenting documents and their words (e.g. matrices generated by word2vec).

In order to be able to compare a number of such matrices learned over different time slices, its important to make all these matrices pointing to the same positions in the word2vec space such that a word vector from one model is comparable to the same word vector in a different model. In order to achieve that in Chapter 9 we used the Procrustes matrix approximation by applying SVD [117]. This model can effectively track complex evolving patterns over time and is therefore suitable for solving similar time-series problems.

10.2 Future Work

Here at the end of this dissertation we go back to where we started, designing SILAS. In Chapter 3, we presented SILAS a system for summarizing one's daily life and specially one's conversations into event snippets. As explained SILAS processes images, conversation transcripts, GPS coordinates along with corresponding time steps. In Chapter 4, we presented the dDTM model which can compute topic chains over time. The dDTM model was also integrated into SILAS.

This combination allows for searching for people who appear in one's lifelog images and find out what were the topics of discussions that took place with those people. Similarly, one could search for a topic and find out which people or locations are associated with that topic. Such system can serve as a personal life search engine where one can store various modalities of one's life.

In Chapter 6, we went beyond the mentioned data modalities and made use of biophysical sensor data such as EDA. We explained in Chapter 6, how EDA is used as a measure for gauging one's level of attention and hence level of recall of past events and conversations. By integrating more modalities of data into SILAS as future work, we can effectively build a rich database of one's life. For instance, by also monitoring and storing one's interactions with one's smartphone and computer devices, one's health-related data captured by a smartwatch, one's sent and received emails, one's daily search queries submitted to a search engine, one's online social media activities, videos and movie titles that one watches we can make a very rich database.

Subsequently, by using JITIR and other prediction and inference models, similar to the models that we introduced in Chapters 7, 8, and 9 sophisticated personal assistant systems can be built that one can rely on for increasing productivity and having a more healthy and fulfilling life. Utilizing such rich dataset can result in discovering complex associations in various data modalities and give valuable advice to one accordingly. For example, one can receive health advice or progress reports as well as to do steps regarding certain meetings.

In order for such vision to be fully implemented without compromising one's privacy and security, certain steps need to be taken: (1) The owner of the lifelog shall have full and exclusive control over own data stored over a trusted cloud or local repository. (2) The prediction and inference models used shall have transparency and the ability to explain the reasons behind their predictions. As as an example, the K2RE and SK2RE models that we introduced in Chapters 7 and 8, can make some explanation regarding a topic whose predicted as a continuing topic in a future meeting in terms of recency or establishment weights. Moreover, our CATS model can effectively generate summary sentences on a user's desired topics. Despite these steps towards transparency and explainable inferences in this dissertation, much work needs to be done in this challenging area of research. (3) Other people's privacy who appear on one's lifelog should be respected by building mechanisms that ask for their explicit permission to appear in one's lifelog, or be fully anonymized in the data that is stored.

Langheirich [81] explains that social computing in general has certain social implications that makes it sensitive in terms of privacy issues. These social implications are "ubiquoty" (i.e. ubiquitous computing such as lifelogging is everywhere and impacts our daily lives), "invisibility" (i.e., computing devices are shrinking in size), "sensing" (i.e., computing devices are becoming more powerful, capable of sensing and recording various dimensions of our lives including our emotions), and "memory amplification" (i.e., advancements in retrieval systems can make it feasible to perceive memory prosthesis or amplifier). Langheinrich further explains that given these social implications, "it is us who need to understand the potential and danger of our advancements, and develop sound conventions and guidelines according to well-established principles that will help us drive technology into a responsible and socially acceptable direction.” In order to increase the societal acceptance of a personal assistant that utilizes one’s lifelog dataset to assist one, we must first establish guidelines for using them and develop technologies that ensure the privacy of people as well as data security.

By combining the two main themes of this thesis, *Summarization* and *JITIR*, several interesting directions for future work is envisaged. One direction is the investigation of JITIR voice notifications using natural language to users for personal assistance.

Another extended version of this view-point is conversational information retrieval where a user interacts with a virtual assistant via natural language. Current conversational assistants, however, have virtually no JITIR capabilities that if added can bring a significant improvement to the paradigm of personal assistance.

Integration of various sensor data, recording different aspects of one’s life can significantly improve JITIR in such conversational assistant.

Now that we presented a broad picture for future work, we present more specific future work for the two parts of this thesis:

Regarding Part I, we believe that SILAS can be greatly extended with other data modalities given the wealth of current and emerging wearable sensors that record various personal information. As explained above, such step can enable sophisticated inferences about one’s behavior and coming up with valuable information and advice to assist one in every day life.

The dDTM can be further engineered to identify trending topics in one’s yearly, monthly, weekly and daily conversations. Currently, the model can be run on data with various time lengths, but does not have any components that can associate similar topics produced by different runs on different time lengths. Such step may be interesting for having a hierarchical view of main topic threads that were discussed in a year and how each one reveals further details as dDTM is applied to monthly data which should have a higher level of granularity.

The CATS summarization model can be extended to control the focus of its output summaries using a reference document or a user profile recording user interests. Furthermore, an extended CATS model can enable conversational search of the data that SILAS or another personal assistant system has access to. This enables the use of natural language to communicate with a personal assistant system.

Regarding part II, biophysical signals other than EDA such as heart rate or blood volume pressure can be investigated for prediction of memorable/ non-memorable segments of a conversation. Moreover, various effects such as the effect of one meeting attendee forgetting/ remembering a segment of meeting on the other attendee is another direction for future work.

Adapting the K2RE method to other user intent and context tracking domains and comparing it with other methods in those domains is another future work. For example, analyzing datasets which current personal assistants such as Microsoft Cortana or Google Now gather (that track user behavior to anticipate their information needs) could be a possibility.

In the case of our just-in-time search query topic prediction model, we are interested in investigating the effect of search trends on the behavior of each user. Moreover, taking into account user search session information can be another direction for future work.

Bibliography

[1] Mohannad Abuzneid and Ausif Mahmood. Enhanced human face recognition using LBPH descriptor, multi-knn, and back-propagation neural network. volume 6, pages 20641–20651, 2018.

[2] Eugene Agichtein, Ryen W. White, Susan T. Dumais, and Paul N. Bennet. Search, interrupted: Understanding and predicting search task continuation. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 315–324, 2012.

[3] Mohammad Aliannejadi and Fabio Crestani. Venue appropriateness prediction for personalized context-aware venue suggestion. In *Proceedings of the International ACM SIGIR Conference*, SIGIR '17, pages 1177–1180, 2017.

[4] Mohammad Aliannejadi and Fabio Crestani. Personalized context-aware point of interest recommendation. *ACM Trans. Inf. Syst.*, 36(4):45:1–45:28, 2018.

[5] Mohammad Aliannejadi, Seyed Ali Bahrainian, Anastasia Giachanou, and Fabio Crestani. University of lugano at trec 2015: Contextual suggestion and temporal summarization tracks.

[6] Mohammad Aliannejadi, Dimitrios Rafailidis, and Fabio Crestani. A collaborative ranking model with multiple location-based similarities for venue suggestion. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR*, pages 19–26, 2018.

[7] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. In situ and context-aware target apps selection for unified mobile search. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1383–1392, 2018.

[8] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. Target apps selection: Towards a unified search framework for mobile devices. In *Proceedings of the International ACM SIGIR Conference*, SIGIR '18, pages 215–224, 2018.

[9] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the International ACM SIGIR Conference*, SIGIR '19, pages 475–484, 2019.

[10] Mohammad Aliannejadi, Dimitrios Rafailidis, and Fabio Crestani. A joint two-phase time-sensitive regularized collaborative ranking model for point of interest recommendation. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2019 (in press).

[11] Ioannis Arapakis, Joemon M. Jose, and Philip D. Gray. Affective feedback: An investigation into the role of emotions in the information seeking process. In *Proceedings of the 31st Annual International ACM SIGIR Conference*, SIGIR '08, pages 395–402, 2008.

[12] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[13] Seyed Ali Bahrainian and Fabio Crestani. Cued retrieval of personal memories of social interactions. In *Proceedings of the First Workshop on Lifelogging Tools and Applications*, LTA '16, pages 3–12, 2016.

[14] Seyed Ali Bahrainian and Fabio Crestani. Tracking smartphone app usage for time-aware recommendation. In *Digital Libraries: Data, Information, and Knowledge for Digital Lives - 19th International Conference on Asia-Pacific Digital Libraries, ICADL 2017, Bangkok, Thailand, November 13-15, 2017, Proceedings*, pages 161–172, 2017.

[15] Seyed Ali Bahrainian and Fabio Crestani. Towards the next generation of personal assistants: Systems that know when you forget. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR 2017, Amsterdam, The Netherlands, October 1-4, 2017*, pages 169–176, 2017.

[16] Seyed Ali Bahrainian and Fabio Crestani. Predicting the topics to review in preparation of your next meeting. In *Proceedings of the 8th Italian Information Retrieval Workshop, Lugano, Switzerland, June 05-07, 2017.*, pages 13–16, 2017.

[17] Seyed Ali Bahrainian and Fabio Crestani. Are conversation logs useful sources for generating memory cues for recalling past memories? In *Proceedings of the 2nd Workshop on Lifelogging Tools and Applications*, LTA '17, 2017.

[18] Seyed Ali Bahrainian and Fabio Crestani. Augmentation of human memory: Anticipating topics that continue in the next meeting. In *Proceedings of the 2018 Conference on Human Information Interaction & Retrieval*, CHIIR '18, pages 150–159, 2018.

[19] Seyed-Ali Bahrainian and Andreas Dengel. Sentiment analysis using sentiment features. In *Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 03*, pages 26–29, 2013.

[20] Seyed Ali Bahrainian and Andreas Dengel. Sentiment analysis of texts by capturing underlying sentiment patterns. In *Web Intelligence*, volume 13, pages 53–68, 2015.

[21] Seyed Ali Bahrainian, Seyed Mohammad Bahrainian, Meytham Salarinasab, and Andreas Dengel. Implementation of an intelligent product recommender system in an e-store. In *International Conference on Active Media Technology*, pages 174–182, 2010.

[22] Seyed Ali Bahrainian, Marcus Liwicki, and Andreas Dengel. Fuzzy subjective sentiment phrases: A context sensitive and self-maintaining sentiment lexicon. In *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 01*, pages 361–368, 2014.

[23] Seyed Ali Bahrainian, Ida Mele, and Fabio Crestani. Modeling discrete dynamic topics. In *Proceedings of the Symposium on Applied Computing*, SAC '17, pages 858–865, 2017.

[24] Seyed Ali Bahrainian, Ida Mele, and Fabio Crestani. Predicting topics in scholarly papers. In *European Conference on Information Retrieval*, pages 16–28, 2018.

[25] Seyed Ali Bahrainian, Fattane Zarrinkalam, Ida Mele, and Fabio Crestani. Predicting the topic of your next query for just-in-time ir. In *Advances in Information Retrieval*, pages 261–275, 2019.

[26] Michele Banko, Vibhu O Mittal, and Michael J Witbrock. Headline generation based on statistical translation. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 318–325, 2000.

[27] Leonard E. Baum and J. A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. *Bull. Amer. Math. Soc.*, 73(3):360–363, 05 1967.

[28] Leonard E. Baum and George R. Sell. Growth transformations for functions on manifolds. *Pacific Journal of Mathematics*, 1968.

[29] E Berry, A Hampshire, J Rowe, S Hodges, N Kapur, P Watson, G Browne, G Smyth, K Wood, and AM Owen. The neural basis of effective memory therapy in a patient with limbic encephalitis. *Journal of Neurol Neurosurg Psychiatry*, 80:1202–1205, 2009.

[30] Agon Bexheti and Marc Langheinrich. Understanding usage control requirements in pervasive memory augmentation systems. In *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia*, MUM '15, pages 400–404, 2015.

[31] Agon Bexheti, Evangelos Niforatos, Seyed Ali Bahrainian, Marc Langheinrich, and Fabio Crestani. Measuring the effect of cued recall on work meetings. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, UbiComp '16, pages 1020–1026.

[32] Agon Bexheti, Anton Fedosov, Ivan Elhart, and Marc Langheinrich. Memstone: A tangible interface for controlling capture and sharing of personal memories. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '18, pages 20:1–20:13, 2018.

[33] David M. Blei and John D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 113–120, 2006.

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

[35] Jason J Braithwaite, Derrick G Watson, Robert Jones, and Mickey Rowe. A guide for analysing electrodermal activity (eda) & skin conductance responses (scrs) for psychological experiments. *Handbook of Psychophysiology*, pages 1017–1034.

[36] Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. Deep communicating agents for abstractive summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1662–1675, 2018.

[37] Yen-Chun Chen and Mohit Bansal. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, ACL '18.

[38] Yi Chen and Gareth JF Jones. Augmenting human memory using personal lifelogs. In *Proceedings of the 1st augmented human international conference*, 2010.

[39] Jianpeng Cheng and Mirella Lapata. Neural summarization by extracting sentences and words. *arXiv preprint arXiv:1603.07252*, 2016.

[40] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, pages 22–29, 1990.

[41] Trevor Cohn and Mirella Lapata. Sentence compression beyond word deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 137–144, 2008.

[42] David Corney, Dyaa Albakour, Miguel Martinez, and Samir Moussa. What do a million news articles look like? In *Proceedings of the First International Workshop on Recent Trends in News Information Retrieval, Padua, Italy, March 20, 2016.*, pages 42–47, 2016.

[43] Fabio Crestani and Heather Du. Written versus spoken queries: A qualitative and quantitative comparative analysis. *Journal of the American Society for Information Science and Technology*, 57(7):881–890, 2006.

[44] Anne M.; Filion Diane L. Cacioppo John T. (Ed); Tassinary Louis G. (Ed); Berntson Gary G. Dawson, Michael E.; Schell. Sentiment in short strength detection informal text. *Handbook of psychophysiology, 2nd ed.*, pages 200–223, 2000.

[45] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38, 1977.

[46] Aiden R. Doherty, Alan F. Smeaton, Keansub Lee, and Daniel P. W. Ellis. Multimodal segmentation of lifelog data. In *Large Scale Semantic Access to Content (Text, Image, Video, and Sound)*, RIAO '07, pages 21–38, 2007.

[47] Hermann Ebbinghaus. Memory: A contribution to experimental psychology. 1885, translated in 1913.

[48] Ashlee Edwards, Diane Kelly, and Leif Azzopardi. The impact of query interface design on stress, workload and performance. In *Proceedings of 37th European Conference on IR Research*, ECIR '15, pages 691–702, 2015.

[49] Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479, 2004.

[50] Mik Lamming et.al. The design of a human memory prosthesis. *The Computer Journal*, pages 153–163, 1994.

[51] Chong Feng, Fei Cai, Honghui Chen, and Maarten de Rijke. Attentive encoder-based extractive text summarization. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM'18, pages 1499–1502, 2018.

[52] Ferenc Galkó and Carsten Eickhoff. Biomedical question answering via weighted neural network passage retrieval. In *European Conference on Information Retrieval*, pages 523–528, 2018.

[53] Sebastian Gehrmann, Yuntian Deng, and Alexander M. Rush. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, EMNLP '18, pages 4098–4109, 2018.

[54] Jim Gemmell, Gordon Bell, Roger Lueder, Steven Drucker, and Curtis Wong. Mylifebits: Fulfilling the memex vision. In *Proceedings of the 10th ACM International Conference on Multimedia*, pages 235–238, 2002.

[55] Zoubin Ghahramani. Hidden markov models. chapter An Introduction to Hidden Markov Models and Bayesian Networks, pages 9–42. World Scientific Publishing Co., Inc., 2002.

[56] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610, 2005.

[57] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber. Lstm: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2222–2232, 2017.

[58] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences*, 2004.

[59] Ramanathan Guha, Vineet Gupta, Vivek Raghunathan, and Ramakrishnan Srikant. User modeling for a personal assistant. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 275–284, 2015.

[60] Cathal Gurrin, Alan F. Smeaton, and Aiden R. Doherty. Lifelogging: Personal big data. *Foundations and Trends in Information Retrieval*, 2014.

[61] Cathal Gurrin, Hideo Joho, Frank Hopfgartner, Liting Zhou, and Rami Albatal. Ntcir lifelog: The first test collection for lifelog research. 2016.

[62] Josef Hallberg, Basel Kikhia, Johan Bengtsson, Stefan Sävenstedt, and Kåre Synnes. Reminiscence processes using life-log entities for persons with mild dementia. *1st International Workshop on Reminiscence Systems, Cambridge, UK*, pages 16–21, 2009.

[63] Morgan Harvey, Marc Langheinrich, and Geoff Ward. Remembering through lifelogging: A survey of human memory augmentation. *Pervasive and Mobile Computing*, pages 14–26, 2016.

[64] Jennifer Healey and Rosalind Picard. Digital processing of affective signals. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 6, pages 3749–3752, 1998.

[65] Marti A. Hearst. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Comput. Linguist.*, 23(1):33–64, 1997.

[66] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701, 2015.

[67] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of International ACM SIGIR Conference*, SIGIR '99, pages 50–57, 1999.

[68] Wan Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, and Min Sun. A unified model for extractive and abstractive summarization using inconsistency loss. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

[69] Pei-Yun Hsueh and Johanna D. Moore. Improving meeting summarization by focusing on user needs: A task-oriented evaluation. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, pages 17–26, 2009.

[70] A. Jaimes, H. Bourlard, S. Renals, and J. Carletta. Recording, indexing, summarizing, and accessing meeting videos: An overview of the ami project. In *14th International Conference of Image Analysis and Processing - Workshops (ICIAPW 2007)*, pages 59–64, 2007.

[71] Ro Jaimes, Kengo Omura, Takeshi Nagamine, and Kazutaka Hirata. Memory cues for meeting video retrieval. In *Proceedings CARPE 04*, pages 74–85, 2004.

[72] Nebojsa Jojic, Alessandro Perina, and Vittorio Murino. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, NIPS '10, pages 1027–1035.

[73] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82 (Series D):35–45, 1960.

[74] Vaiva Kalnikaité and Steve Whittaker. Software or wetware?: Discovering when and why people use digital prosthetic memory. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 71–80.

[75] C.H. Kennedy. *Single-case Designs for Educational Research*. Pearson/A & B, 2005.

[76] Basel Kikhia, Josef Hallberg, Kåre Synnes, and Zaheer Ul Hussain Sani. Context-aware life-logging for persons with mild dementia. In *Engineering in Medicine and Biology Society, 2009*, 2009.

[77] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 426–434, 2008.

[78] Yehuda Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 447–456, 2009.

[79] Wojciech Kryscinski, Romain Paulus, Caiming Xiong, and Richard Socher. Improving abstraction in text summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, EMNLP '18, pages 1808–1817, 2018.

[80] Mik Lamming and Mike Flynn. Forget-me-not: Intimate computing in support of human memory. pages 125–128, 1994.

[81] Marc Langheinrich. Privacy by designâĂŤprinciples of privacy-aware ubiquitous systems. In *Proceedings of International conference on Ubiquitous Computing*, UbiComp '01, pages 273–291, 2001.

[82] G.F. Lawler. *Introduction to Stochastic Processes, Second Edition*. Chapman & Hall/CRC Probability Series. 2006.

[83] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[84] Matthew L. Lee and Anind K. Dey. Lifelogging memory appliance for people with episodic memory impairment. In *Proceedings of International conference on Ubiquitous Computing*, UbiComp '08, pages 44–53, 2008.

[85] Matthew L. Lee and Anind K. Dey. Wearable experience capture for episodic memory support. In *12th IEEE International Symposium on Wearable Computers (ISWC 2008)*, pages 107–108, 2008.

[86] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185, 2014.

[87] Wei Li, Xinyan Xiao, Yajuan Lyu, and Yuanzhuo Wang. Improving neural abstractive document summarization with explicit information selection modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, EMNLP '18, pages 1787–1796, 2018.

[88] Wen Li, Carsten Eickhoff, and Arjen P de Vries. Probabilistic local expert retrieval. In *Advances in Information Retrieval*, pages 227–239. 2016.

[89] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004.

[90] H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165, 1958.

[91] Ida Mele, Seyed Ali Bahrainian, and Fabio Crestani. Linking news across multiple streams for timeliness analysis. In *Proceedings of the 26th ACM International on Conference on Information and Knowledge Management*, CIKM '17, 2017.

[92] Ida Mele, Seyed Ali Bahrainian, and Fabio Crestani. Event mining and timeliness analysis from heterogeneous news streams. *Inf. Process. Manage.*, 56(3):969–993, 2019.

[93] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, EMNLP '04, pages 404–411, 2004.

[94] 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.

[95] Yashar Moshfeghi and Joemon M. Jose. An effective implicit relevance feedback technique using affective, physiological and behavioural features. In *Proceedings of the 36th International ACM SIGIR Conference*, SIGIR '13, pages 133–142, 2013.

[96] G. Murray. Automatic summarization of meeting. In *PhD thesis, University of Edinburgh, UK*, 2007.

[97] Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of the 20th Conference on Computational Natural Language Learning*, CoNLL '16.

[98] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[99] Douglas W. Oard and Joseph Malionek. In *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries*.

[100] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. A picture of search. In *Proceedings of the 1st International Conference on Scalable Information Systems (InfoScale '06)*, page 1, 2006.

[101] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*, 2017.

[102] Katalin Pauly-Takacs, Chris JA Moulin, and Edward J Estlin. Sensecam as a rehabilitation tool in a child with anterograde amnesia. *Memory*, 19(7): 705–712, 2011.

[103] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. 2014.

[104] Marcin Pietrzykowskiand and Wojciech Salabun. Applications of hidden markov model: state-of-the-art. In *International Journal of Computer Technology and Applications*, 2014.

[105] Silva A R, Pinho S, Macedo L, and Moulin C J. Does sensecam improve general cognitive performance? *Am J Prev Med*, 44(3):302–307, 2013.

[106] Lawrence R. Rabiner. Readings in speech recognition. chapter A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, pages 267–296. 1990.

[107] B. J. Rhodes and P. Maes. Just-in-time information retrieval agents. *IBM Syst. J.*, 39:685–704, 2000.

[108] Bradley Rhodes and Thad Starner. Remembrance agent: A continuously running automated information retrieval system. 1996.

[109] Bradley J. Rhodes. The wearable remembrance agent: A system for augmented memory. In *Proceedings of the 1st IEEE International Symposium on Wearable Computers*, ISWC '97, 1997.

[110] Mark O Riedl and Robert Michael Young. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research*, 39:217–268, 2010.

[111] Esteban Andres Rissola, Seyed Ali Bahrainian, and Fabio Crestani. Personality recognition in conversations using capsule neural networks. In *IEEE/WIC/ACM International Conference on Web Intelligence*, WI '19, pages 180–187, 2019.

[112] Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

[113] Nuzhah Gooda Sahib, Anastasios Tombros, and Tony Stockman. A comparative analysis of the information-seeking behavior of visually impaired and sighted searchers. *Journal of the American Society for Information Science and Technology*, 63(2):377–391, 2012.

[114] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, pages 1257–1264, 2007.

[115] Sepideh Saran. Faceblur: A privacy preserving mobile application. *Master Thesis, Technical University of Kaiserslautern*, 2016.

[116] D.L. Schacter, D.T. Gilbert, D.M. Wegner, and B.M. Hood. *Psychology*. Palgrave Macmillan, 2012.

[117] Peter H Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966.

[118] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[119] Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[120] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083, 2017.

[121] Abigail Sellen, Andrew Fogg, Mike Aitken, Steve Hodges, Carsten Rother, and Kenneth R. Wood. Do life-logging technologies support memory for the past?: an experimental study using sensecam. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 81–90, 2007.

[122] Guokan Shang, Wensi Ding, Zekun Zhang, Antoine Jean-Pierre Tixier, Polykarpos Meladianos, Michalis Vazirgiannis, and Jean-Pierre Lorré. Unsupervised abstractive meeting summarization with multi-sentence compression and budgeted submodular maximization. *arXiv preprint arXiv:1805.05271*, 2018.

[123] Milad Shokouhi and Qi Guo. From queries to cards: Re-ranking proactive card recommendations based on reactive search history. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 695–704, 2015.

[124] Padhraic Smyth, David Heckerman, and Michael I. Jordan. Probabilistic independence networks for hidden markov probability models. *Neural Comput.*, 9(2):227–269, 1997.

[125] Yang Song and Qi Guo. Query-less: Predicting task repetition for nextgen proactive search and recommendation engines. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 543–553, 2016.

[126] Josef Steinberger. Using latent semantic analysis in text summarization and summary evaluation.

[127] Yu Sun, Nicholas Jing Yuan, Yingzi Wang, Xing Xie, Kieran McDonald, and Rui Zhang. Contextual intent tracking for personal assistants. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 273–282, 2016.

[128] Simon Sweeney and Fabio Crestani. Effective search results summary size and device screen size: Is there a relationship? *Information processing & management*, 42(4):1056–1074, 2006.

[129] Y. Taigman, Ming Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1701–1708, 2014.

[130] Jaime Teevan, Susan T. Dumais, and Eric Horvitz. Personalizing search via automated analysis of interests and activities. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 449–456, 2005.

[131] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. Sentiment in short strength detection informal text. *J. Am. Soc. Inf. Sci. Technol.*, 61(12):2544–2558, December 2010.

[132] Paul Thomas, Daniel McDuff, Mary Czerwinski, and Nick Craswell. Misc: A data set of information-seeking conversations. In *Proceedings of the 1st International Workshop on Conversational Approaches to Information Retrieval*, 2017.

[133] Anastasios Tombros and Mark Sanderson. Advantages of query biased summaries in information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 2–10, 1998.

[134] Johanne R Trippas, Damiano Spina, Lawrence Cavedon, Hideo Joho, and Mark Sanderson. Informing the design of spoken conversational search: Perspective paper. In *Proceedings of the 2018 Conference on Human Information Interaction&Retrieval*, pages 32–41, 2018.

[135] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, ACL '16, 2016.

[136] S. Tucker and S. Whittaker. Temporal compression of speech: An evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 2008.

[137] Simon Tucker and Steve Whittaker. Accessing multimodal meeting data: Systems, problems and possibilities. In *International Workshop on Machine Learning for Multimodal Interaction*, pages 1–11, 2004.

[138] Simon Tucker and Steve Whittaker. Time is of the essence: An evaluation of temporal compression algorithms. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 329–338, 2006.

[139] Tobias Grossmann Vaish, Amrisha and Amanda Woodward. Not all emotions are created equal: The negativity bias in social-emotional development. *Psychological bulletin 134.3*, pages 383–403, 2008.

[140] Luke Vilnis and Andrew McCallum. Word representations via gaussian embedding. *arXiv preprint arXiv:1412.6623*, 2014.

[141] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700, 2015.

[142] Paul A. Viola and Michael J. Jones. Rapid object detection using a boosted cascade of simple features. In *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), with CD-ROM, 8-14 December 2001, Kauai, HI, USA*, pages 511–518, 2001.

[143] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theor.*, 13(2):260–269, 2006.

[144] A. Waibel, T. Schultz, M. Bett, M. Denecke, R. Malkin, I. Rogina, R. Stiefelhagen, and Jie Yang. Smart: the smart meeting room task at isl. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, volume 4, pages IV–752–5 vol.4, April 2003.

[145] Chong Wang, David Blei, and David Heckerman. Continuous time dynamic topic models. In *Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence*, UAI '08.

[146] Lu Wang and Claire Cardie. Summarizing decisions in spoken meetings. In *In Proceedings of the Workshop on Automatic Summarization for Different Genres, Media, and Languages*, WASDGML '11, pages 16–24, 2011.

[147] Xuerui Wang and Andrew McCallum. Topics over time: A non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, 2006.

[148] Xing Wei and W. Bruce Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the International ACM SIGIR Conference*, SIGIR '06, 2006.

[149] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[150] Jihang Ye, Zhe Zhu, and Hong Cheng. What?s your next move: User activity prediction in location-based social networks. In *Proceedings of the SIAM International Conference on Data Mining. SIAM*, 2013.

[151] Ryuichi Yoshida and et al. Feasibility study on estimating visual attention using electrodermal activity. In *8th International Conference on Sensing Technology*, 2014.

[152] David Zajic, Bonnie Dorr, and Richard Schwartz. Bbn/umd at duc-2004: Topiary. In *Proceedings of the HLT-NAACL 2004 Document Understanding Workshop*, pages 112–119, 2004.

[153] Fattane Zarrinkalam, Mohsen Kahani, and Ebrahim Bagheri. Mining user interests over active topics on social networks. *Information Processing and Management*, 54:339–357, 03 2018.

[154] Ruiqiang Zhang, Yuki Konda, Anlei Dong, Pranam Kolari, Yi Chang, and Zhaohui Zheng. Learning recurrent event queries for web search. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP ’10, pages 1129–1139, 2010.