UNIVERSITÉ DU LUXEMBOURG

PhD-FSTC-2019-72
The Faculty of Sciences, Technology and Communication

DISSERTATION

Defense held on 29/11/2019 in Esch-sur-Alzette

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

Siwen GUO
Born on 21 August 1988 in Harbin (China)

DEEP NEURAL NETWORKS FOR PERSONALIZED SENTIMENT ANALYSIS WITH INFORMATION DECAY

Dissertation defense committee

Dr. Christoph Schommer, dissertation supervisor
Professor, Université du Luxembourg

Dr. Leon van der Torre, Chairman
Professor, Université du Luxembourg

Dr. Pouyan Ziafati, Vice Chairman
LuxAI S.A.

Dr. Tiansi Dong
Universität Bonn

Dr. Kai Hui
Amazon, Berlin

Abstract

People have different lexical choices when expressing their opinions. Sentiment analysis, as a way to automatically detect and categorize people's opinions in text, needs to reflect this diversity. In this research, I look beyond the traditional population-level sentiment modeling and leverage socio-psychological theories to incorporate the concept of personalized modeling. In particular, a hierarchical neural network is constructed, which takes related information from a person's past expressions to provide a better understanding of the sentiment from the expresser's perspective. Such personalized models can suffer from the data sparsity issue, therefore they are difficult to develop. In this work, this issue is addressed by introducing the user information at the input such that the individuality from each user can be captured without building a model for each user and the network is trained in one process.

The evolution of a person's sentiment over time is another aspect to investigate in personalization. It can be suggested that recent incidents or opinions may have more effect on the person's current sentiment than the older ones, and the relativeness between the targets of the incidents or opinions plays a role on the effect. Moreover, psychological studies have argued that individual variation exists in how frequently people change their sentiments. In order to study these phenomena in sentiment analysis, an attention mechanism which is reshaped with the Hawkes process is applied on top of a recurrent network for a user-specific design. Furthermore, the modified attention mechanism delivers a functionality in addition to the conventional neural networks, which offers flexibility in modeling information decay for temporal sequences with various time intervals.

The developed model targets data from social platforms and Twitter is used as an example. After experimenting with manually and automatically labeled datasets, it can be found that the input formulation for representing the concerned information and the network design are the two major impact factors of the performance. With the proposed model, positive results have been observed which confirm the effectiveness of including user-specific information. The results reciprocally support the psychological theories through the real-world actions observed. The research carried out in this dissertation demonstrates a comprehensive study of the significance of considering individuality in sentiment analysis, which opens up new perspectives for future research in the area and brings opportunities for various applications.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to Prof. Christoph Schommer for giving me the opportunity to work as a doctoral student and for his patient guidance throughout my thesis. He led me on the path of research and let me grow as a research scientist in academics. His valuable advice has widen my research from various perspectives, and his encouragement has driven me to persevere. Despite the challenges and difficulties during the process, he has always been supportive. I am deeply grateful for the extraordinary experience of being mentored by him.

I want to thank Prof. Christoph Schommer, Dr. Sviatlana Höhn, and Dr. Winfried Höhn for their insightful inputs and for helping me with all the problems I encountered. I would not have achieved any progress in research or have enjoyed the doctoral life without their assistance. I treasure all our discussions that kept me passionate about the task and made dull ideas sparkle.

I am grateful to my thesis supervision committee, Prof. Christoph Schommer, Prof. Leon van der Torre, Dr. Feiyu Xu, and Dr. Sviatlana Höhn, for their constant support along the way. I appreciate their taking time from their full schedules to have meetings with me and to make sure that I am on the right direction towards the completion.

Moreover, my sincere thanks go to Daniela Gierschek, Ekaterina Kamlovskaya, Dr. Sviatlana Höhn, Dr. Joshgun Sirajzade, and Dr. Vladimir Despotovic for proofreading my thesis. I am truly thankful for their diligent reading and constructive comments and suggestions.

I also thank the University of Luxembourg for providing the pleasant and dynamic academic environment. The balance between research and education has brought me fondness for the doctoral program, and I am glad to be a member of the community.

Last but not the least, I want to thank my parents, Binli and Xuetian, for their love and care, my friends for cheering me up when I feel down, and all the people around me for staying in good health.

Contents

List of Acronyms and Abbreviations

AR	Atomic Representation
CNN	Convolutional Neural Network
Combi	Combined Granular Levels
CS	Cosine Similarity
CW	Concepts and Words
DC	Deep Contextualization
ED	Euclidean Distance
ELMo	Embeddings from Language Models
EMD	Earth Mover's Distance
GloVe	Global Vectors for Word Representation
G-SVM	Generalized Support Vector Machine
G-RNN	Generalized Recurrent Neural Network
IF	Input Formulation
LSTM	Long Short-term Memory
MD	Manhattan Distance
NLP	Natural Language Processing
P-SVM	Personalized Support Vector Machine
RBF	Radial Basis Function
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
SVM	Support Vector Machine
WE	Word Embeddings
WMD	Word Mover's Distance

List of Figures

List of Tables

Part I

Background

Sentiment is one of the key factors affecting human behavior. Studying the way in which sentiment is expressed, evolved and perceived is a crucial part of artificial intelligence. Sentiment can be explicit or implicit, public or personal, consistent or variable, rational or emotional, certain or self-doubting. Apart from differentiating exact definitions between sentiment, emotion, affect and feeling, the understanding of these terms can vary due to complex reasons of linguistic or psychological nature. Throughout the literature over a hundred years, psychologists have been theorizing on interpretations, yet no universal consensus is made. Among the attempts, one that was made by McDougall [1919] stated:

> ‘.... *a sentiment is an organised system of emotional dispositions centred about the idea of some object. The organisation of the sentiments in the developing mind is determined by the course of experience; that is to say, the sentiment is a growth in the structure of the mind that is not natively given in the inherited constitution.*’
>
> — William McDougall, An Introduction to Social Psychology

This statement describes the existence of a target associated with a sentiment, the influence of one’s experience on the sentiments, and implies the development of sentiment over time. Such concepts are commonly entailed in the depictions despite the verbatim difference. Grounded in these psychological aspects, I investigate the possibility of providing a deeper understanding of the expressions made by individuals from their own perspectives.

From the practical point of view, researchers in natural language processing (NLP) have made different assumptions on linguistic behaviors that are applicable for the tasks at hand. Moreover, those assumptions are leveraged combining approaches developed based on the nature of the text, the representation of the related information and the objectives. Focusing on individual variations in the study of sentiment analysis, in Part I, I will introduce the task and the contributions of this research (Chapter 1), provide a general view of sentiment analysis including the ubiquitous methods and applications as well as my views on the role of personalization in the study (Chapter 2), and elucidate pertinent deep learning approaches applied in this area (Chapter 3).

Chapter 1

Introduction

Sentiment analysis is defined in the Oxford dictionaries[1] as

> *'The process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc. is positive, negative, or neutral.'*

This definition outlines three types of information that are essential to the study: the text, the target (topic, product, etc.) and the writer. It also reflects the evolvement of this field from document- or sentence-level [Wiebe et al., 2001; Meena and Prabhakar, 2007] which takes the document or sentence as a whole, to aspect-level [Cheng and Xu, 2008; Pontiki et al., 2014] which considers various aspects of a target, and later to an advanced level where the text is not the only source for determining sentiments and the diversity among the writers (or speakers, users depending on the application) is considered as well. The focus of this research lies in the advanced level which is called *personalized sentiment analysis.*

1.1. Research Motivation

The significance of considering the sentiment holder (expresser) is based on the observation that people are diverse and they express their sentiments in distinct ways [Reiter and Sripada, 2002]. Such diversity is caused by many factors such as linguistic and cultural background, expertise and experience. While different lexical choices are made

[1] `https://en.oxforddictionaries.com/definition/sentiment_analysis`, last seen on August 1, 2019

by sentiment holders, a model that is tailored by the individual differences should be built accordingly. Here, a model that includes individual differences in sentiment analysis is named *personalized sentiment model*. Note that this task is distinguished from personality modeling [Markovikj et al., 2013] where such diversity in expressions is also considered in form of linguistic features in discovering users' personality.

In order to investigate the expresser-related factors that influence the analysis of sentiment, assumptions derived from a number of psychological studies are applied. Particularly, the indicated expresser, the target of the expression (e.g. the topic), and the time at which the expression was made are considered as major contributors to the expresser's lexical choices. One may argue that such consideration can be biased for there are other contributors that are involved, e.g., the intended addressee(s) of the expression as in the study of *recipient design* [Sacks et al., 1978]. In this work, a monologic setting is posited where the sentiment expressed in a piece of text is made independently without (or with low) expectation of an exchange, in contrast to a dialogical setting as in conversations. Intuitively, exploring the expresser's past is a feasible way of learning his / her preferences, which is also the approach taken by the researchers in personalized sentiment analysis. However, among the major contributors mentioned above, time is the least studied one in the literature. Thus, this study aims to bridge the gap in this research by developing a personalized sentiment model and at the same time, to examine how the time contributor affects the information obtained from the expresser's past and whether this contributor can be beneficial to the modeling of sentiments.

To facilitate and evaluate the utilization of the aforementioned assumptions in the modeling, data from social platforms are targeted, for which Twitter[2] is used in the experiments. The posts on Twitter contain statements in a form of text, image, and/or video accompanied by user identifiers, timestamps, and occasionally with other information such as hyperlinks and hashtags (to index keywords or topics). The information used in this study is the text, user identifier, and timestamp. The Twitter data align with the setting of the work, and the employment helps realize the aspect of personalization in the analysis. There can be a certain level of expectation in exchanging, however, since the posts are publicly available (users can also choose to publish protected posts[3]), it can be assumed that the masses are the target recipients and the willingness of the targets to reply depends on their own availability. Moreover, the posts on Twitter span over a large range of topics that rich information concerning various word usages can be accessed and obtained. Such data is domain-independent thus applicable to various tasks, which differ from domain-specific data such as product reviews [Cui et al., 2006; Yu et al., 2011] and political commentaries [Carvalho et al., 2011].

Specifically, a neural-based system is proposed for the modeling. The adoption of

[2] https://twitter.com/

[3] https://help.twitter.com/en/safety-and-security/public-and-protected-tweets, last seen on August 1, 2019

such a system is motivated by the prominent performance reported in many existing works in NLP over the past years when comparing to traditional methods such as conditional random fields [Yao et al., 2013], naive Bayes models [Johnson and Zhang, 2015], and logistic regression [Lai et al., 2015]. Nevertheless, it is dubitable that neural networks can always provide the best result — the choice of methods is often task-specific, hence experiments oriented to this task have to be conducted. In addition, I explore a network design that allows incorporating time information in the modeling which is not typically inherent in conventional neural networks. The incorporation enables a continuous process inside the network so that the intermediate information learned from different input contributors can be correlated.

1.2. Problem Statement

The modeling of personalization aspects is often realized by building separate individual models. A critical issue of generating a model for each user is the data sparsity. There is an inconsistency in the frequency of posting messages on social platforms per user. For instance, it was reported in 2016 that Twitter had 700 million annually active users, of which 420 million were quarterly active and 317 million were monthly active[4]. The gap between the numbers shows that the amount of messages (also called 'tweets') published per user is normally in the range of a few to a few thousand with roughly 500 million tweets sent per day[5], and the frequency of the postings varies from user to user. These statistics show that it is intuitively infeasible to create individual models given the imbalanced amount of data each model associates with, especially when the applied algorithm for classification heavily relies on the large data for training. In this thesis, a framework with neural networks is introduced to model individualities in expressing opinions, which intrinsically offers a solution for the data sparsity since only one model is required. Note that here, the users who have only posted once are excluded for no past information can be learned from such users. This leads to the common issue known as the problem of 'cold-start' in recommender systems [Schein et al., 2002] which deserves separate research. Further, I focus on frequent users to emphasize on the effect of discovering individualities. Additionally, this work relieves from the user group assumption (which states that people in the same community share similar behaviors) that contradicts the statement of Harris [2010] on the individual uniqueness.

On social platforms, another phenomenon is that the entity behind a user account is not necessarily one particular individual — it can be a public account run by a person or a group of persons who represent an organization. It is also possible for a person to have more than one account, e.g. a private account and a work account. Despite the fact that a person may act or express himself / herself differently while using different

[4] https://www.fool.com/investing/2016/11/06/twitter-has-700-million-yearly-active-users.aspx, last seen on August 1, 2019

[5] http://www.internetlivestats.com/twitter-statistics/#trend, last seen on August 1, 2019

accounts, the way of expressing opinions by the person(s) behind one account tends to be consistent. However, the writer of a text (or the publisher of a post in the setting of Twitter) is not necessarily the person who holds the sentiment. As in Liu [2015], the situation was elaborated with an example: A review of product (target) 'Canon G12 camera' by John Smith contains a piece of text *'I simply love it .... my wife thinks it is too heavy for her.'*. The example shows different opinions from two persons published by John Smith who thinks positively towards the target while his wife holds a negative opinion. An accurate research should involve a study that identifies the holder of a sentiment before generating a sentiment score for it. The negligence of this aspect in sentiment analysis is caused by the lack of demand in most applications where opinions are desired regardless of which persons express them. Nonetheless, exceptions exist for the task of establishing user groups or for security reasons where locating the holders is as prioritized as extracting opinions. To simplify the task, sentiment holder or opinion holder is mostly used to indicate the person who publishes the text when it comes to analyzing individual behaviors through short messages posted on social platforms.

Non-canonical language, such as the user-generated text on social platforms, brings difficulties in text processing and analysis, although the fundament of the challenge and the precise definition of the canonicity in such a context have been debated in some literature [Plank et al., 2015; Plank, 2016]. The processing of such text is pertinent to choosing the proper granularity for representing the text — generally as the first step in NLP tasks [Mcnamee and Mayfield, 2004; Turian et al., 2010; He et al., 2018]. Finer granularities may be more robust against language variations, whereas coarser granularities can encompass richer information. For sentiment analysis, the past two decades have witnessed a gradual shift from syntax-based methods to semantic-aware systems where extracting conceptual knowledge for language understanding is more predominant than mere text processing [Cambria, 2016]. Here, both challenges are considered regarding the granularity in the representation (the non-canonicity and conceptuality), and possible solutions will be discussed in detail.

Sentiment is dynamic. For example, a person that thinks positively towards a football team before a match may feel negatively towards the same team after. Changes can be expected in one's views about certain topics, and one's lexical choices may also vary over time. However, many existing works in this area fail to recognize the dynamic feature in sentiment. The question that arises with such recognition is whether the dynamicity positively influences the analysis as well as to what extent the influence (if any) affects the system. To answer this question, I propose to include the time factor while constructing the personalized model and then to evaluate the significance of the inclusion. In neural networks, it is not a trivial task to model precise time intervals that interact with the information carried or transformed from the input. Most neural network structures merely consider temporal sequences with respect to the order of the appearances of events [Schuster and Paliwal, 1997; Cho et al., 2014; Dehghani et al., 2018], thus are not sufficient for this study. Therefore, alternative structures that take into account the impact of various time gaps ought to be explored.

1.3. Methodology

In this research, a preliminary model and an advanced model are demonstrated for personalized sentiment analysis, where the former follows an assumption-based approach and the latter is refined from the observations of the preliminary model. The proposed models and their variants are evaluated with Twitter data, and a number of experiments are conducted to provide a comprehensive view on the research topic. The results and findings obtained in the evaluation enable us to reflect on multiple implications from different perspectives.

The methodology employed in this work can be decomposed into three folds. The first one concerns the development of the representation method for the associated information. In order to realize the primary elements pertaining to modeling individuality in sentiment, three assumptions are suggested according to the psychological theories reported in the literature. With the objective of leveraging the assumptions and thereby evaluating their effectiveness on the modeling, corresponding information is extracted from the posts (tweets) and represented in atomic form, for which a series of preprocessing is performed. Concepts and topics, as components in the atomic representation, are used in separate shallow neural networks to produce embeddings for the input of the subsequent model. Despite the discontinuity in modeling, the use of the separate embedding networks allows a better inspection and the unsupervised training of the shallow networks offers flexibility in selecting the training data. Apart from the assumption-based representation used in the preliminary study, different granular levels, namely concept-level, word-level, character-level, and combined granular levels, are later explored as well to unearth the most advantageous way for representing the related information. Furthermore, the user information is added as a solution for data sparsity and the time information is included in the advanced model so that the effect of information decay under various intervals can be studied.

The second one corresponds to the construction of the central network which takes the input generated with a pre-defined formulation method and outputs the prediction of the sentiment of the current post. In both the preliminary model and the advanced model, a three-layered recurrent neural network (RNN) with long short-term memory (LSTM, Hochreiter and Schmidhuber [1997]) is applied. The input of the network is a temporal sequence whose elements correspond to a series of past posts of an indicated user. In the preliminary model, the user information is padded to the sequence, whereas in the advanced model, it is added to each post representation. The central network discovers the relations between the past and the current posts in order to benefit the prediction with the user-related information. Since no separate networks are trained for the advanced model, a hierarchical structure is utilized to generate embeddings for the representations with different granularities prior to the RNN. That is, the lower hierarchy focuses on the semantic information inside a post concerning the intra-post relation while the higher hierarchy focuses on analyzing inter-post relations. Moreover, the attention mechanism is applied on top of the RNN for the purpose of creating direct paths to the information learned in the past, and in addition, an input selection algorithm is presented to (pre-)select the related posts from the entire posting history of a user.

The last one is the modeling of information decay in the neural-based system. Traditional RNNs do not have the ability to differentiate various gaps between time steps. An intuitive solution is to use the time information as an auxiliary input and merge it with other information in the network. However, the interaction of the information from different sources can be difficult to control and visualize. Heuristically, Hawkes process, as a special type of point process, is capable of modeling the decay of events' intensities in temporal sequences, thus can be exploited to estimate the time effect on sentiment given a user's post history. More importantly, the behavior of the decay process can be inspected using such a method. As a solution, the Hawkes process is integrated in the attention mechanism so that the decay is dependent not only on the time gap between the posts but also on the relativeness of their content. Further, a user – factor transformation is proposed to enable user-specific processes, i.e., different user behaviors regarding the decay can be modeled.

For the evaluation, manually annotated data are used in the preliminary model. The restricted size of the data prevents us from expanding to sufficient frequent users. Thus, automatically labeled data are used in the advanced model aiming to target comparably more frequent users for the study, and meanwhile represent standpoints from the expresser's view. The differences between the two annotation techniques will be discussed in detail. Since the development and evaluation are based on the chosen data from social networks, this research can be regarded as data-driven, however with the possibility of extending to other data types.

1.4. Work Contribution

The contribution of this work lies in both theoretical and practical aspects. Theoretically, the assumptions derived from the (socio-)psychological theories [Janis and Field, 1956; Nowak et al., 1990; Reiter and Sripada, 2002] are leveraged, which in turn provides evidence for the theories with the implementation of the real-world data. The conclusion drawn from the implementation reveals the significance of considering individualities in sentiment analysis while analyzing user-generated text. Specifically, there is a degree of consistency in an expresser's lexical choices with a connection to the topic or entities mentioned in the text. Moreover, the more frequent an expresser expresses (by publishing posts on social networks), the better a model can learn about the expresser and can predict the sentiment of the expresser's future expressions; the longer the past a model can relate to, the better the model is able to learn and predict. Such acts align with the behavior in human communications, which implies similar processes in acquiring knowledge between the machine and human. Additionally, the information learned from the past can be outdated in a way that recent events can have more impact on the current state of the expresser than the older ones. This decay of information also has different impacts on expressers — some people change their minds more frequently than others, which can be topic-dependent as well. While these discoveries can seem straightforward, they are not considered in most sentiment-related tasks. With this study, it can be concluded that such individualities can have a positive influence on the analysis, and they ought to be taken into account.

On the practical side, a personalized sentiment model with deep neural networks is developed that specializes in modeling user dynamics in expressions. To the best of the author's knowledge, this is the first neural-based system that targets the personalization aspect in sentiment analysis associated with open-domain text, and is also the first work that considers information decay in this respect (at the time of the development). A hierarchical structure is used for the modeling and different granularities and formulations in the lower hierarchy are compared. In particular, the advantage of using finer representations and combined representations are demonstrated in a series of experiments. A simple solution is proposed to solve the issue with data sparsity that enables the modeling for less frequent users. The higher hierarchy facilitates the ability of analyzing the past of an expresser, which can be applied separately on top of other pre-built representation models or pre-trained embeddings (as vectors of chosen granularities). On a technical side, by applying a novel approach that integrates the Hawkes process with attention mechanism, it also offers a possibility to model various time intervals of temporal sequences in neural-based systems. As a further step, user information is imported in the Hawkes process in order to learn user-specific processes, and the behaviors of these different processes can be visualized.

Chapter 2

The Area of Sentiment Analysis

Sentiment analysis, as a relatively 'late bloomer' in NLP, has been rapidly evolved over the last three decades. The evolvement is influenced and promoted by the needs in the industrial market as well as the developments in other related areas such as new techniques in language technology and computational supports in hardwares. Meanwhile, sentiment analysis has also become a standard task for semantic evaluation whereas having been an isolated, goal-oriented task in the early years. In this chapter, a brief introduction of the field will be given and a number of prevalent approaches reported in existing literature will be listed. Further, more background related to this research with respect to the annotation techniques and the use of contextual information will be elucidated, and selected aspects regarding real-world applications will be discussed as well.

2.1. Definitions

The first appearance of the term *sentiment analysis* was in Nasukawa and Yi [2003], although sentiment-related researches had begun earlier. It was originated from the community of NLP, and was formulated following the task of extracting sentiment expressions in texts. Despite the advances in the area over the years, a universally accepted definition lacks. In the literature, the use of the term can be generally classified into two notions: One is to use it as a denotation for the broader field covering a large problem space as mentioned in Liu [2015]:

> *'Sentiment analysis, also called opinion mining, is the field of study that analyzes people's opinions, appraisals, attitudes, and emotions toward entities and their attributes expressed in written text.'*

Such a definition implies the inclusion of the subareas such as subjectivity detection, opinion extraction, and emotion analysis. The other notion is as in Wilson et al. [2005] and Nakov et al. [2016], where the task of sentiment analysis inclines to identifying certain expressions with regard to the two polarities:

> *'Sentiment analysis is the task of identifying positive and negative opinions, emotions, and evaluations.'*

In this notion, sentiment analysis can be seen as a subtask of emotion analysis which assesses the level of valence that corresponds to one of the dimensions of (certain) emotion models [Russell, 1980; Schimmack and Grob, 2000]. In this research, the use of the first notion is acknowledged, but the indicated task is prone to the second one: the analysis is primarily concerned through determining polarities.

Another confusion regarding the use of the term is the differentiation between *sentiment analysis* and *opinion mining*. *Opinion mining* was first appeared in Dave et al. [2003], where the techniques used in information retrieval were exploited to search for appropriate features for a given item after which the users' opinions about the item were determined. Note that no difference between these two terms is made in the definition of Liu [2015] shown above, but the author has mentioned the tendency of using *sentiment analysis* in industry whereas both are common in academia. Most of the literature has used the terms interchangeably or have used them without additional explanations of the difference [Pang et al., 2008; Pak and Paroubek, 2010; Ravi and Ravi, 2015]. Here, a clear distinction of the terms is not recognized for the difference is subtle and it is not a focus of this work to answer this question.

2.1.1. A General Understanding of Sentiment

In the Merriam Webster's dictionary, one of the entries in the definition of sentiment is *'an attitude, thought, or judgment prompted by feeling'* while some other entries associate the definition with emotion[6]. Indeed, several related terms such as sentiment, emotion, and affect are often mentioned in each other's descriptions, and the difference among them should be clarified.

From a psychological point of view, Stets [2006] has argued that sentiments and emotions are defined in such a similar way in many literature that no separation is necessary; however, Munezero et al. [2014] have stated that they differ in the duration in which they are experienced: it has been found that compared to emotions, the formation and the course of sentiments take a longer period, which makes sentiments more stable and dispositional. Additionally, sentiments can be more object-directed [Russell and Barrett, 1999], while on the contrary, affect is not object-directed that has a primitive and non-conscious nature [Shouse, 2005; Liu, 2015]. Moreover, in many works,

[6] https://www.merriam-webster.com/dictionary/sentiment, last seen on August 1, 2019

opinions are used interchangeably with sentiments, but opinions can be expressed without being emotionally charged or constrained by social expectations while sentiments are 'partly social constructs of emotions' [Lasersohn, 2005; Stets, 2006]. The differences among the terms are mostly factored by the duration, object-directiveness, consciousness, and social-constructiveness. An illustration of the relations and distinctions of the terms is demonstrated in Figure 2.1.

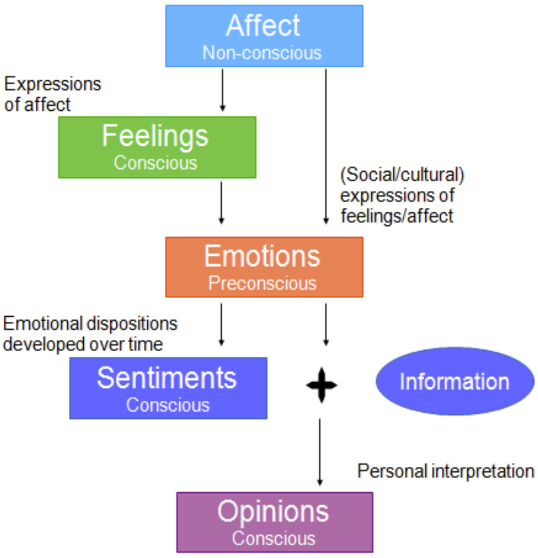


Figure 2.1.: Relations and distinctions of the terms related to sentiment [Stets, 2006].

In sentiment analysis, the expressions of the terms are seldom distinguished. Such negligence is caused by the difficulty in realizing the subtle differences among them as well as the lack of needs in the realization. Consequently, sentiment is used to indicate any type of subjectivity associated with the text being studied. For example, when the studied text is product reviews, the sentiment can be opinions; in open-domain studies, the sentiment can be any of the terms mentioned above. Exceptionally, in emotion analysis, categorized emotions may be regarded as elements in a multidimensional space (according to the emotion model adopted) in contrast to the sentiment which is believed to be unidimensional [Mohammad, 2016b]. Given the inconsistent use of the terms in psychology and sentiment analysis as well as the little effort contributed to forming the consistency in existing works, it can be excused from delving into the differentiation of the terms in this study.

2.1.2. The Interdisciplinary Aspect

The field of sentiment analysis involves not only branches from computer science such as natural language processing, machine learning, and data mining, but also other areas such as linguistics, psychology, and sociology. Therefore, the advance of these areas is in the vanguard of the development of sentiment analysis. While technological improvements have given impetus to a substantial progress in the arena, behavioral analysis can be beneficial to the study as well. The employment of the behavioral aspects depends on the nature of the text and objective of the research. For example, the task of Tang et al. [2015c] was to analyze review texts written by different users and to predict the ratings of the reviews for restaurants and movies. The authors have considered the user variations in interpreting the same word regarding the orientation and strength of the sentiment: a user may use the word 'good' for an assessment of excellent (5 out of 5) while another user may use it to express ordinariness (3 out of 5). Yang and Eisenstein [2015, 2017] have exploited social factors to analyze the different word usages on social networks with respect to polysemy. They believe that such differences are rarely idiosyncratic and that there are communities within which socially linked individuals share linguistic homophily. By detecting such communities through network topology, sociolinguistic properties are generalized without exploring demographic metadata. Indeed, many interdisciplinary aspects are realized in the literature that are shown to be beneficial to the task [Gonçalves et al., 2013; Wang et al., 2013].

2.2. Different Levels of Sentiment Analysis

The researches in sentiment analysis can be categorized by the level of text unit to which a prediction is oriented. There are mainly four levels, namely document-level, sentence-level, aspect-level, and personalized-level, and different methodologies are developed for tasks at different levels.

2.2.1. Document-level and Sentence-level

The document-level sentiment analysis determines one single sentiment for the entire document whereas for sentence-level analysis, the goal is to find a sentiment per sentence where a document may contain more than one sentence. Although the tasks are delineated differently, the relation between the two levels can be comprehended through the view of Liu [2012]:

> *'... there is no fundamental difference between document and sentence level classifications because sentences are just short documents.'*

However, since documents are normally longer than sentences, it can be more difficult to perform sentence-level analysis for less information is associated. Especially when encountering special sentences like sarcastic or conditional ones, document-level can be more robust by referring to the context which can be of significant help for such situations. Nevertheless, the same methods such as supervised and unsupervised classification algorithms can be applied for both tasks [Pang and Lee, 2004; Moraes et al., 2013; Xu et al., 2016].

2.2.2. Aspect-level

Some texts such as product reviews contain opinions on various aspects (attributes, features, properties) of the entities. Sentiment analysis at aspect-level corresponds to the task of recognizing all the entities and their aspects in a document and assigning a polarity to each aspect of the entities. Schouten and Frasincar [2015] have distinguished three processing steps when performing aspect-level sentiment analysis: identification, classification, and aggregation. The identification step is to recognize sentiment-target pairs in the text; the classification step is to assign sentiment values to the pairs; the aggregation step is to combine the sentiment values for each aspect. However, not all the steps are followed in research works — while there are works that fulfill the objective of the complete task [Brody and Elhadad, 2010; Toh and Wang, 2014], some works concentrate on subtasks in which aspect extraction [Jakob and Gurevych, 2010; Poria et al., 2016a] and aspect sentiment classification [Tang et al., 2016; Wang et al., 2016] have received the most attention.

Aspect-level analysis has opened up more challenges in the field. Although it is always important to understand implicit expressions in NLP, in this particular task, one needs to detect implicit aspects of entities that may require external knowledge for inferring. For example, to recognize the aspect *price* in the comment 'It is a quite affordable apartment', or to recognize *weight* in 'The phone is a bit too heavy', additional semantic relations must be explored. Moreover, the analysis is sensitive to the order of the terms in a sentence, especially when more than one aspect is mentioned such as in the compound sentence *'The food is pricy but the service is fantastic'*. In recent years, a number of methods have been developed to solve the task while the optimal solution awaits to arise [Schouten and Frasincar, 2015].

2.2.3. Personalized-level

Personalized-level sentiment analysis is relatively understudied compared to document- and aspect-level sentiment analysis. Personalized-level sentiment analysis, or simply personalized sentiment analysis, is the task to perform document- or aspect-level predictions when considering individualities. The challenge of this task is to handle variations concerning the tendency of lexicon use and to discover ways to incorporate different information sources. As one of the earliest works in this field, Li et al. [2010] intended to capture unique user characteristics by adopting a global model which was leveraged afterwards to refine the individual models using collaborative online learning. The task of this work was to detect subjectivity in micro-blogs, which was later extended to sentiment classification in Li et al. [2011], however only six highly frequent users were investigated. The difficulty caused by the scarce data with less frequent users was realized in the works. Later, the concept of adapting individual models from a global model was also applied in other studies via multi-task learning [Wu and Huang, 2016; Gong et al., 2016, 2017], and assumptions on social relations [Song et al., 2015, 2016; Zhao et al., 2017] or user groups [Gong et al., 2017] were leveraged to enhance the personalization aspect.

The utilization of neural networks enables the inclusion of the individual analysis independent of the user group assumption. Targeting product reviews, Chen et al. [2016b] applied two separate RNNs to generate user and product representations in order to model the individual differences in assigning rating scores and to obtain the consistencies in receiving rating scores of the same product. A convolutional neural network (CNN) was used to generate embeddings for the review text. In the end, the representations from the three parties were combined using a traditional machine learning classifier. Another work on product reviews was done by Chen et al. [2016a] who employed a hierarchical network with LSTM on word-level and sentence-level representations. Additionally, an attention mechanism based on user and product information was used on each level after the LSTM layer. By doing that, user preferences and product characteristics were introduced into the network to produce finer-represented document embeddings. Moreover, Wang et al. [2018b] investigated the cross-lingual aspect regarding the individuality and opinion bias, and a sophisticated adversarial framework with CNN and attention mechanism was proposed for the task. There are similar works that consider individual differences related to sentiment [Tang et al., 2015b; Dou, 2017; Wu et al., 2018b]; however, very few have explicitly modeled the evolvement of sentiments of an individual over time. To bridge this research gap, I focus on the personalization aspect and have developed a sentiment model towards an evaluation of the influence of time gaps while relating to the earlier posted texts.

2.3. Ubiquitous Approaches

There has been a shift in the choice of approaches during the evolvement of sentiment analysis, that is from rule-based methods to traditional machine learning techniques and to neural networks. While the shift can be task-specific given the advantages and disadvantages of the approaches, model ensembles with different techniques have also been attempted in some works [Mesnil et al., 2014; Wang et al., 2014; Araque et al., 2017]. Here, I opt to introduce the methods individually without the ensembles.

2.3.1. Rule-based Methods

Rule-based methods correspond to the group of the methods that employ sentiment lexicons and linguistic rules to conduct the analysis. The fundamental task is to perform a sentiment lexicon acquisition because 'words and phrases that convey positive or negative sentiment are instrumental for sentiment analysis' [Liu, 2015]. There are mainly three possibilities to acquire a sentiment lexicon, that is in a manual, dictionary-based, or corpus-based manner. However, since manual acquisition can be time- and labor-intensive, it is mostly used for creating seed sentiment words which are extended later using a dictionary- or corpus-based approach. The dictionary-based approach means to expand a list of seed words by exploring synonyms and antonyms [Hu and Liu, 2004], whereas the corpus-based approach expands the seeds by exploiting certain linguistic rules such as the sentiment consistency derived from the functionality of conjunction words [Hatzivassiloglou and McKeown, 1997] and the dependency in co-occurrent words or phrases according to their syntactic patterns [Turney, 2002]. The dictionary-based approach is domain-independent thus it is generally applicable, but it can be infeasible for analyzing domain-specific sentiment orientations. For example, the polarity of 'explosive' can be positive in finance as in 'explosive growth' whereas it can carry a negative sentiment in other contexts such as 'explosive temper'. Meanwhile, the corpus-based approach is the opposite, as sentiment-carrying words are extracted in the context in which they appear. After constructing the sentiment lexicon, human inspection is needed to check the possible errors in the lexicon made during the automatic process. At last, the sentiment orientations of the words in a sentence are aggregated to produce one value for the whole sentence [Taboada, 2016]. This type of methods can be helpless for special sentences such as the ones with sarcastic and comparative expressions.

2.3.2. Traditional Machine Learning Techniques

Various traditional machine learning techniques are applicable for the task. Such techniques can be divided into supervised and unsupervised (and semi-supervised depending on the rule of separation) categories according to the presence and absence of annotated data. While lexicon-based approaches can be viewed as unsupervised learning approaches [Labille et al., 2017], the boundary of the categorization can be vague since labeled data can also be used for sentiment lexicon acquisition [Oliveira et al., 2016]. Hence in this section, I focus on the supervised methods used for determining the sentiment orientation.

To perform the learning process, a number of features are extracted from the document as a form of representation. Among the adopted features, bag-of-words, *tf-idf* weights, part-of-speech tags, sentiment indicators and shifters are widely employed [Sebastiani, 2002; Whitelaw et al., 2005]. Most of the traditional machine learning techniques require extensive feature engineering for dimension reduction [Ng et al., 1997; Galavotti et al., 2000], and some approaches have been developed for this purpose [Yang and Pedersen, 1997]. For the classification task, many algorithms such as support vector machine (SVM) [Mullen and Collier, 2004] and naive Bayes [Tan et al., 2009] can be applied, and the performance of the algorithms can vary depending on the data and the task. Moreover, Liu et al. [2013] have shown that it is also possible to use the naive Bayes classifier when scaling up the size of the data considerably.

2.3.3. Neural Networks

The advance of deep learning during the last decade has given impetus to the development in NLP. Specifically, the employment of neural networks has been shown to improve the performance of many subtasks in sentiment analysis [Zhang et al., 2018]. One distinct trait of neural networks concerns the process of feature engineering. In contrast to the traditional classifiers, neural networks do not (necessarily) require the selection and extraction of features, which can be advantageous to many NLP tasks. However, the input of a neural network has to be represented in a numerical form. Later in Chapter 4, the different methods to transform raw inputs into numerical representations will be discussed. There are distinct network structures among which RNN, CNN, and attention network are prevalent choices [Liu et al., 2016; Severyn and Moschitti, 2015; Shen et al., 2018]. Given the characteristics of the different structures, some variants and combinations are proposed to ameliorate the demerits or to perfect the employment.

Applying the attention mechanism on top of RNN(s) is a common usage with regard to structure combination. For instance, Baziotis et al. [2017] adopted this combination for both the task of message-level (document-level) sentiment analysis and the task of topic-based polarity classification, and competitive results were achieved for both tasks. Targeting aspect-level sentiment analysis, Chen et al. [2017] have taken a step further by adding a position-weighted memory to acquire flexibility when multiple targets are mentioned in one comment and designing a recurrent attention module to distill and manufacture the related information from the weighted memory. As a refinement, inspired by the effective use of gating mechanisms in RNN, Xue and Li [2018] have applied gated tanh-ReLU units between the convolution and pooling layer in CNN for the same task; however, the way to leverage large-scale sentiment lexicons within the network remains unsolved. Since sentiment analysis has now become a standard task for evaluating language understanding, many pre-trained text representations or language models targeting general NLP tasks can be adapted to the task directly and improved performance can be expected [Radford et al., 2018; Devlin et al., 2019; Liu et al., 2019].

2.4. The Text and the Labeling

When using supervised machine learning techniques, annotated data are needed for the training process; in the rule-based methods (dictionary- or corpus-based), a list of seeds is used to define the polarity classes, where human intervention is also required. One of the drawbacks of the hand labeled data is the lack of fundamental reliability caused by subjectivity since people's interests, profession, knowledge status, and a common ability to understand the text differ from one another. One of the strategies to acquire a consensus is the 'Wisdom of Crowd' [Surowiecki, 2005], which claims that a majority of a sufficient large group of judges will lead to a stable and correct decision. However, it is still risky: dependencies among the judges as well as external influences (e.g. trends) may affect the general reliability. A more credible solution could be to use (independent) voting processes and to retain an independency among the judges by following statistical measures such as Cohen's kappa coefficient [Cohen, 1960] or Krippendorff's alpha coefficient [Krippendorff, 2011]. Moreover, some works have discussed the complexity in annotating different types of sentences and texts, and tried to measure the complexity or proposed solutions for different situations [Joshi et al., 2014; Mohammad, 2016a]. For example, news articles are difficult to annotate because the reporters are normally supposed to provide facts with a neutral tone, but they can lean towards their own opinions (consciously or unconsciously) or can be assumed to carry certain emotions as a natural reflection such as to feel sad when writing *'Seven injured in this accident'*. Sarcastic expressions and sentences meant to make supplications are also difficult to label for prior knowledge is needed in some cases.

Another drawback of manual labeling is that it is time consuming which leads to limitation with respect to the data size. For the text on social platforms, one particular phenomenon is the use of emoticons. An automatic labeling mechanism is possible by categorizing the emoticons that appeared in the messages, such as in Go et al. [2009], where emoticons were employed in a distant supervision approach. With the automatic labeling, the limitation regarding the data size can be eliminated. The authors have found that such labeling is effective, although there may be a certain level of noise caused by the variation in emoticon usage and the unreliability at the user end. Furthermore, manual and automatic labeling can be seen as annotating from different standpoints [Schommer et al., 2013]. For instance, when annotating the sentence *'Great, the price is going up'*, a seller and a buyer may have different opinions such that a seller can see it as a positive sentiment while a buyer can see it as a sarcastic sentence and label it as negative. In such case, the sentiment of this sentence should be labeled from the writer's perspective, and the perspective can be checked by the automatic labeling.

2.5. The Use of Contextual Information

In linguistic studies, the notion of 'context' varies from theory to theory. Monologism sees it as 'secondary complications', whereas in dialogist theory, it considers the reflexive relation between an expression and its setting or occasion essential [Linell, 2009]. In this work, social text is targeted and context is regarded as an important factor in the setting of social platforms. Based on the characteristics of such text and the applicability in modeling, the information used in the sentiment model is categorized into two genres: *textual information* — the information that can be extracted directly from the target text, and *contextual information* — the information that is not present in the target text but is associated with the text through the corresponding user. Individualities can be revealed from both the textual and contextual information. It can be argued that assuming users that are connected on social networks share similar opinions is insufficient for personalized modeling [Tan et al., 2011]. Hence, according to the focus of this research, each user is considered individually and the use of contextual information for the task is discussed.

The first indicational contextual information is the user identifier. In personalized sentiment analysis, the user identifier is used in distinct ways as an indispensable feature. When analyzing review texts, the underlying product, as the target of the sentiment, can also be used explicitly in the modeling [Chen et al., 2016a; Wu et al., 2018b]. In this situation, the product can be seen as textual information or contextual information depending on the texts under study. While the commentaries are on the webpage of the indicated product, the review text may omit the product information as it is implied. The use of the sentiment target can assist in learning user preferences of the products as well as in acquiring preferred lexical choices associated with the target. When dealing with other types of text, the target can be inferred from the indicated text if it is not given beforehand; however for text with an open domain, it can be difficult to infer the target and a valid inference may require external knowledge such as semantic relations for interpreting.

Studying a person's past is one of the ways to understand the person. Extracting other statements made by the same user has been shown to be efficient for the modeling [Gong et al., 2016; Lynn et al., 2017], thus they can be regarded as useful contextual information as well. However, the existing studies ignored the order or time of the statements at which they were made since the model was trained in a batch mode. In contrast, I believe that the order and the exact time when the past expressions were made are also valuable contextual information. For instance, an expression made a month ago may be less influential to the current one than an expression made an hour ago. For that, the time information ought to be included in the analysis.

2.6. Real-world Applications

Nowadays, the blooming digital media have brought and diffused massive information from individual users to the public, and mining desired content from the gathered information becomes an imminent task. Sentiment analysis, as a way to uncover public opinions, is widely used in many applications such as measuring company reputation [Colleoni et al., 2011], political campaign investigation [Wang et al., 2012], and terrorism detection [Iskandar, 2017]. While the population-level applications are desirable, there are also personalized-level applications whose potential can be foreseen. In this section, some use cases of personalized sentiment analysis are presented and the possible ethical and legal issues brought by the personalization study are discussed.

2.6.1. Use Cases

Song et al. [2016] have provided the first real-world individuality-dependent sentiment classification system targeting microblog users on Twitter. The system has the functionality of visualizing the home and user timeline in a compact way such that the sentiment-related information is also displayed. By using the system, users can browse and organize the tweets more efficiently, which is useful for users who have a large number of tweets or have followed many frequent users. In addition, they have integrated a real-time tweet sentiment analysis function for which the personalization aspect is considered and the sentiment model is trained with emoticon-labeled data. As reported in their work, the proposed extensible latent factor model is able to give a quick response despite the overhead during system start caused by the data collection and model building.

Sentiment analysis is shown to be useful in recommender systems [Yang et al., 2013; Diao et al., 2014]. It has been applied not only as an additional component for understanding user interests but also as a tool for finding like-minded people. Gurini et al. [2013] have used sentiment analysis in Twitter user recommendation with the assumption that taking into account user attitudes towards his own interests can help recommend other users to follow. However, most of the works utilize sentiment analysis at the population-level where the individuality is not considered. Nevertheless, Rosa et al. [2015] intended to discover the association between the user's profile and the sentiment intensity in order to benefit music recommendation systems. They have extracted user messages from social networks to monitor the user's emotional state at certain time slots and calculated the sentiment value according to a sentiment correction factor obtained based on the user profile. It has been found that by incorporating the user and sentiment information, the selected music can reach a better user satisfaction. Although the personalization aspect was not directly applied for the classification in their work, the effect of modeling the variations among the users was realized, which detached the research from the population-level assumption.

Another use case is consumer sentiment analysis. Chamlertwat et al. [2012] proposed a microblog sentiment analysis system for discovering consumer insight and reconfirmed that sentiment analysis can help companies make decision on their next generation product. Zhou et al. [2019] furthered the application to mining consumer repurchase intention and distinguished the relation between an initial purchase and a repurchase, which assists in formulating marketing strategies. Moreover, they have stated the potential of applying user sentiment analysis on market segmentation that can be done through profiling after which different countermeasures can be taken for different groups of people. As an extended application, tracking customer sentiment over time can help enterprise monitor changes in market reaction, e.g., to understand the cause of an uptick in sales.

Last but not least, sentiment analysis has been used for identifying opinion leaders who have the power to guide the direction of (online) public opinion [Yu et al., 2010]. Regarding political opinion mining, Stieglitz and Dang-Xuan [2013] have found that most of the frequently retweeted users do have a clear political affiliation[7] and their tweets tend to be more emotionally charged compared to the general masses. Their work reveals a number of valuable observations that can help uncover political inclinations and understand related social behaviors led by influencers.

2.6.2. Ethical and Legal Issues

The personalization aspect in sentiment analysis brings potential problems with respect to data protection, privacy, and ethics. One is concerning the data used for modeling and the other is regarding arguable activities associated with the built models. Specifically, the European Union law of General Data Protection Regulation[8] (GDPR) provides a comprehensive practice to control the governing and processing of personal data by entities other than the data subject. The impact of the regulation related to scientific research was discussed in Chassang [2017]. When requesting data from social networks, the data anonymization and the consent of using personal data must be obtained by the data controllers[9]; the actions taken by the processors[10]must not violate the purposes for which the data were collected. From the researcher side, the data acquisition follows the provider's instruction which is regulated by the law, whereas the further processing for developments as 'secondary use' must obey the general principle (GDPR Article 5).

In this thesis, public corpora are used for the implementation and additional data from Twitter are taken only for the purpose of conducting experiments. The data crawling strictly follows the developer policies and restrictions of the platform. The research aims to provide an enhanced understanding of the language and sentiment considering individual differences, and will not be used for identifying individuals by reversely adopting the personalized model to match user behavior or to discover user information maliciously.

[7] The investigation was taken in Germany.
[8] https://gdpr.eu/
[9,10] See GDPR Article 4(7,8).

Chapter 3

Deep Neural Networks in Sentiment Analysis

In the last chapter, the background of sentiment analysis was introduced gradually from a general view to the personalization aspect. In this chapter, the practical possibilities are discussed from the realization perspective with an emphasis on using deep neural networks. Neural networks are chosen for the promising performance reported in existing literature and the flexibility with respect to the feature engineering that is prerequisite in traditional machine learning techniques (Section 2.3.2). The fundaments of the network structure used in the proposed models for processing different types of information will be elucidated.

3.1. Hierarchical Structures

In NLP, hierarchical neural networks are popular for tasks that require representations from different representation levels [Yang et al., 2016; Wu et al., 2018b]. Here, I make an example of a three-hierarchy network to demonstrate the information flow within such a network and introduce prevalent structure variations.

3.1.1. An Example of Hierarchical Neural Networks

Figure 3.1 shows an example of hierarchical networks. The input of the illustrated network corresponds to the finest level of the representations, which is the character level in this example denoted as $[c_1, c_2, ..., c_l]$ with the maximum number of characters in a word l (shorter words are padded with zeros). In order to generate the input for the next level, a number of hidden layers h_c are used with the goal of merging the information of the characters from one word into one vector, i.e., $D_c * l \rightarrow D_w * 1$ where D_c is the dimension of the character representation and D_w is the dimension of the word representation. The output of the hidden layers at the character level is the input of the word level, i.e., $w_1 = h_c([c_1, c_2, ..., c_l])$, and the same process is done for other words in $[w_1, w_2, ..., w_m]$.

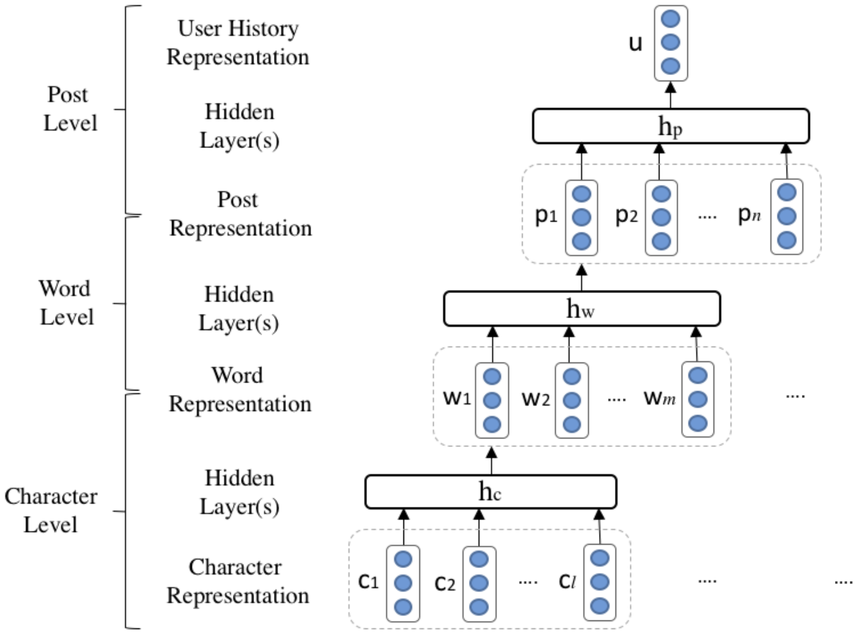


Figure 3.1.: A hierarchical network from character-level to word-level and to post-level (user-specific document-level).

The word level and post level hierarchies are structured in the same fashion as the character level. The user history representation u produced in the highest hierarchy is a vector that contains the encoded textual information of all the past posts from a particular user. This structural design is to reduce the dimensionality of the input so that the hidden layers at one hierarchy can function on one granular level and all the entries of one hierarchy are processed in the same way: the hidden layers at the same hierarchy share the same set of parameters for different inputs while the hidden layers at different hierarchies apply different sets of parameters. The hierarchical network allows performing one training process to jointly coordinate the parameters from all the hierarchies, which is efficient when provided with sufficient annotated data — the training is completely supervised. However, when suffering from insufficient annotated data, pre-trained embeddings can be adopted where the embeddings are acquired separately in an unsupervised manner.

3.1.2. Structure Variations

Hierarchical networks are inherently deep for multiple hidden layers are involved. The hidden layers can take different network structures, for instance, Tang et al. [2015a] have utilized CNN or LSTM on word representations to generate representations for sentences and applied a bi-directional gated network on sentence representations to generate the final document representation; Yang et al. [2016] used the attention mechanism on top of bi-directional RNNs with gated recurrent units for both word level and sentence level hierarchies. Note that the granular level taken as hierarchy can be dependent on the data and the application. Under the setting of Twitter, the character level can be useful for capturing finer language variations and the sentence level can be omitted for there are normally very few sentences in a tweet. As shown in Figure 3.1, a post level hierarchy can be added to produce a representation for each user history such that personalized sentiment modeling is possible. Intuitively, multiple hierarchies can significantly increase the computational complexity, thus networks with more than two hierarchies are rarely seen; however, pertinent researches are missing and experiments should be conducted to examine the efficiency of such networks.

3.2. Information Merging

The hierarchical structure discussed in the last section has displayed one type of information flow where the input is the text and the output can be the class the text is assigned to. Building on that, neural networks can be compatible with multiple input and output types, whereas given the objectives of this research, the multi-typed input and single-typed output are prioritized for the sentiment classification task.

3.2.1. Information Types

Information can originate from various sources and contribute distinctively to different tasks, therefore, to choose the correct and pertinent information is important to the modeling where the performance can be bounded by the choice. In sentiment analysis, some of the valid information types are for example:

- **Multimodality information** which includes visual, audio, and textual features as in Morency et al. [2011]. Most of the social platforms have provided the possibility of using multimodality for opinion sharing nowadays, however the tendency of the use of modalities can vary depending on the platform and the user.

- **Target information** which corresponds to the topic or entity towards which the sentiment is expressed. This information can be explicitly or implicitly mentioned in a modal form, and can also be provided by the communication platform. For instance, the information of a product or a movie can be extracted together with the commentaries on Amazon[11] and IMDb[12] as in Fang and Zhan [2015] and Singh et al. [2013]; although sometimes inaccurate, hashtags on Twitter can provide topic information [Kouloumpis et al., 2011].

- **User information** which contains user profile information, geographical information, following information, and so on [Hu et al., 2013; Zhao et al., 2015a]. The availability and authenticity of such information can not always be guaranteed, and it can be dynamic as well.
- **Recipient information** which corresponds to the knowledge about the intended recipient(s) of the expressions. This type is not well-researched in sentiment analysis, but a person's lexical choices largely depend on the recipient design as mentioned in Sacks et al. [1978] which can be critical for certain tasks (depends on the objective and the nature of the text).
- **Time information** which can be the time an expression was made or the time a sentiment was felt (which may be implied in the text), e.g., *'The celebration after the event was a lot of fun'*.

Different information types may be represented in different forms and have distinctive distributions, for which encoding mechanisms must be employed to transform the information in a way that different types can be merged together in the network.

3.2.2. Merging Methods

There are a number of possibilities to merge different information types in a network. As the example in Figure 3.2 shows, the encoded auxiliary input can be added to the main network at any hidden layer. Chen et al. [2016a] applied attention mechanism on user and product representation, after which the outputs are fed to the hidden layers at both the word- and the sentence level hierarchies. In their work, the attention mechanism can be treated as the encoder of the auxiliary input, and by using the mechanism, the user and product information is intertwined with the textual information. Alternatively, the encoded auxiliary input can be merged into the main network by applying other methods or functions, where one of the simple ways is to concatenate the vectors when they have the same length at all other dimensions. Note that there can be multiple auxiliary inputs, and each can have an encoder of its own or share with other inputs depending on its nature and form of representation; for simplicity, only one auxiliary input is shown in this figure.

Other than merging the auxiliary input into the network, the encoded main input can also be merged with intermediate vectors along the path to create residual connections. In Figure 3.2, the dotted line on the left side between the main encoder and

[11] https://www.amazon.com/

[12] https://www.imdb.com/

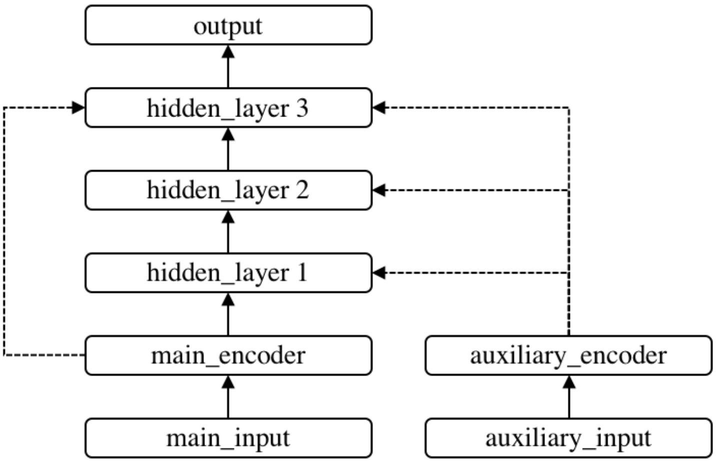


Figure 3.2.: Merging information through different paths in a network.

the third hidden layer is an example of such a case. The residual connections are often done in very deep networks in order to emphasize the input [Conneau et al., 2017]. Given the scale of the deep network proposed in this thesis, it is not necessary to use this structure.

3.3. Time Control in Neural Networks

In this section, I will discuss particularly the merging of time information in the neural networks which is one of the major contributions of this work. There are a few works that concern such an aspect when applying RNNs, and the limitations of the works motivate further research on the subject.

3.3.1. Existing Works on Time Fusion with RNNs

Among the introduced neural network structures, RNN is known for the ability of exhibiting temporal relations inside an input sequence. However, when encountering event sequences where the events appear in an asynchronous mode, the information carried in the dynamic behavior with respect to the various time intervals can be intractable for an RNN without extra measure. To solve this problem, the approaches proposed by the existing works can be mostly sorted into two categories: One is to alter the function inside the network and the other is to combine the network with statistical modeling.

The work by Neil et al. [2016] is an example of the former solution, which considers rhythmic oscillations in sensor data. The authors have taken LSTM as the basic structure for its gated property and extended the composition of a LSTM cell by adding a time gate. The additional time gate is controlled by three parameters: One is to control the period of the oscillation, one is to control the ratio between the active duration and the full period, and the last one is to control the phase shift of the oscillation. With this design, the time gate is capable of influencing the update of the cell state and the output, and the learning phase is accelerated. The phased LSTM can thus handle periodic sequences that have various time intervals among them.

Xiao et al. [2017] have taken the latter approach and employed point process together with the RNN to analyze event sequences. Many point processes have two components with respect to the intensity function: the background intensity and the history influence. They have used two separate RNNs for the two components that each takes time series or event sequences as input. They have suggested the potential of the model in solving real-world problems without prior knowledge, which compensates the inability of point processes in modeling dynamics without pre-conditions or assumptions.

3.3.2. Hawkes Process: Fundaments and Possibilities

Hawkes process [Hawkes, 1971] is a special kind of point process, which is widely used for modeling ‘arrivals’ of events over time. The usage of Hawkes process varies from earthquake modeling [Ogata, 1998] to crime prediction [Mohler et al., 2011], and to financial analysis [Bacry et al., 2015]. As an example close to this study, Hawkes process is also used to predict retweets on Twitter for popularity analysis [Kobayashi and Lambiotte, 2016; Zhao et al., 2015b]. Specifically, it is a one-dimensional point process with a *self-exciting* character. A process is said to be *self-exciting* if an ‘arrival’ causes the conditional intensity function to increase (Figure 3.3) [Laub et al., 2015]. Hawkes process models a sequence of arrivals of events over time, and each arrival excites the process and increases the possibility of a future arrival in a period of time.

As in Laub et al. [2015], the conditional intensity function of Hawkes process is

$$\lambda^*(t) = \lambda + \sum_{t_i<t} \mu(t - t_i) = \lambda + \sum_{t_i<t} \alpha e^{-\beta(t-t_i)} \quad , \tag{3.1}$$

where λ is the background intensity and should always be a positive value, and $\mu(\cdot)$ is the excitation function. Here, exponential decay is used as the excitation function because it is a common choice for many tasks. The value of α and β are positive constants where α describes how much each arrival lifts the intensity of the system and β describes how fast the influence of an arrival decays.

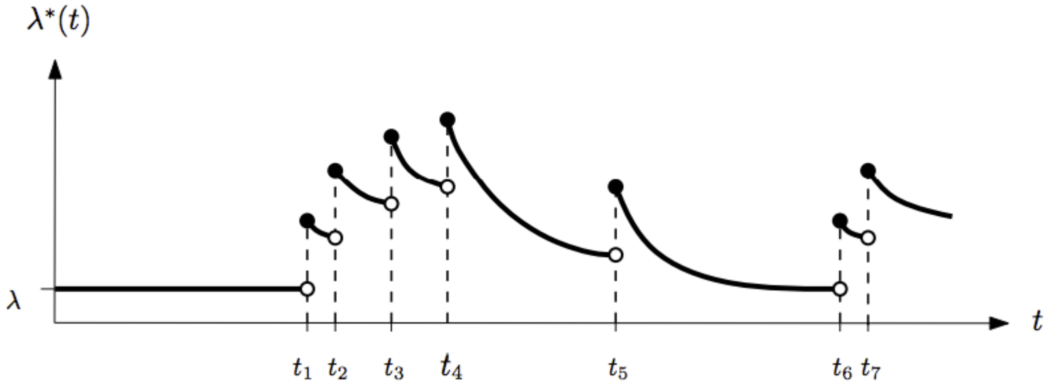


Figure 3.3.: An example of the conditional intensity function for a self-exciting process [Laub et al., 2015]

Hawkes Process in Sentiment

Traditional Hawkes process models the influence of the past events on the future event which assumes that these are the same event (or similar in nature). For that reason, the value of α is constant, i.e. each arrival affects the system in the same way. As a heuristic study, an opinion can be seen as an 'event' that positively influences future opinions, and such influence decreases with time. However, people's opinion may be affected by their past opinions when there are some connections between the targets (topics or entities) of the opinions. Under the setting of this work, the preceding opinion can be irrelevant to the current one, in which case the influence from the preceding opinion should not be boosted. As a solution, a procedure can be performed to filter out the irrelevant expressions, or different levels of relevance between the expressions can be inferred and then merged into the process by altering the value of α. An algorithm designed to filter out unrelated posts will be introduced in Section 5.3, and the notion of altering the value of α will be discussed in Section 6.2.

Incorporating Hawkes Process in Neural Networks

Cao et al. [2017] intends to integrate a Hawkes process in a neural-based system. To avoid pre-defining a time decay function (excitation function $\mu(\cdot)$), they have proposed a non-parametric method where the time range in an observation is split into a number of disjoint intervals and discrete variables are learned for the intervals as the decay effect μ. The approach was developed for the task of popularity analysis through retweet cascades prediction. By embedding the user identity in the input, the resulting network is able to encode the influence of different users on the entire retweet path. Because the events on the path are the same, i.e., the events are the retweets of the same tweet, their solution can not be directly applied to the analysis of sentiment as explained in the last paragraph.

In spite of the possibility of modeling sentiment with Hawkes process, the indicated task of this research is distinguished from existing works that employ the process, and the suitable way to realize the sentiment-based process within neural networks remains unknown. To the best of the author's knowledge, this is the first work that discovers solutions for analyzing the evolvement of sentiment.

Part II

Personalized Sentiment Modeling

Sentiment is personal, as described in the free dictionary[13]

> *'Sentiment — a personal belief or judgment that is not founded on proof or certainty.'*

Given the variety in people's beliefs and judgments as well as the statement of Harris [2010] that 'no two alike', it has been argued that individuality plays a significant role in determining the current sentiment of a target sentiment holder. As the central part of this work, a personalized sentiment model is constructed that intends to capture the traits discussed in the previous chapters. Earlier versions of Part II were published in Guo et al. [2018a,b] and Guo et al. [2019a].

In Part II, I first discuss the means for textual information representation in personalized sentiment analysis including different granular levels for the representation (Chapter 4), then elucidate the structure of the model designed for analyzing post history (Chapter 5), and finally explain the inclusion of time information in the model in order to study the decay of the information learned in the past (Chapter 6). Two different structures were used in my earlier publications. They mainly differ by the design of the input sequence, where the preliminary model adds the user identifier at the end of the sequence as individual nodes and the enhanced one pads the identifier at each node in the sequence. The latter solution provides the network a more consistent input distribution so that the learning process can be executed more efficiently. In order to produce embeddings for the input with atomic representation, a hierarchical structure is utilized and a recurrent network with attention mechanism is applied at the higher level. Moreover, in Section 5.3, I introduce an input selection algorithm to choose the posts that are related to the current topic as well as to look through the entire recorded history of a user. This piece of work was published later and used in the advanced model [Guo et al., 2019c]. Further, to model information decay in sentiment analysis, the Hawkes process is integrated within the attention mechanism to provide a novel and effective solution for modeling exact time gaps in temporal sequences with neural networks. Two types of the Hawkes process, universal Hawkes process and user-specific Hawkes process, are demonstrated in Section 6.2. The final version of the model that integrates different input formulations was published in Guo et al. [2019b].

[13] https://www.thefreedictionary.com/

Chapter 4

Information Representation for Personalized Sentiment Analysis

There are different modalities of information on social platforms that can be used for different purposes. Besides text analysis, extracting information from images is also a popular use in tasks related to social media. For example, Sakaki et al. [2014] combine the two modalities for the task of user gender inference; Baecchi et al. [2016] use the graphical information for the general task of sentiment analysis in microblogs for which Twitter is used as an example. Moreover, audio can convey valuable information as well, as in Poria et al. [2016b, 2017] where such information is realized by applying multimodal fusion. In this work, the use of textual and contextual information is central as introduced in Section 2.5.

The representation of the related information in neural networks can influence the final performance extensively. The granular level of the information unit is an important factor for such influence, where a coarser granularity may contain richer semantic or sentiment information but is less flexible to variations. Generally, a representation with combined granular levels can result in better performance; however the increased size of the input may also challenge the modeling. In the input, textual information can be represented explicitly (e.g., concepts as in Poria et al. [2014]) and / or implicitly (e.g., embeddings as in Pennington et al. [2014] and Peters et al. [2018]). When using embedding techniques, explicitly represented input can be transformed into implicit representation as well. In this chapter, I will discuss the representation for different information types and levels in the setting of sentiment analysis.

4.1. Assumption-based Information Representation

Since the targeted research is motivated by the individual behaviors in the context of (social) psychology, I take theories from the field and extract information that can be used to leverage the theories. Given the complex nature of the text from social platforms, a series of text preprocessing techniques has to be applied for an explicit representation. After that, numerical representation must be created to be used by the neural networks.

4.1.1. The Assumptions

The following assumptions have been discovered in the existing literature and are used as an elementary support in this research.

Assumption I:
Different individuals make different lexical choices to express their opinions.

Reiter and Sripada [2002] have provided clear evidence for the argument that the precise definition of certain words can vary between people, even though a rough level of agreement can be achieved. Particularly, the objects or events associated with the words can differ, for which a detailed user model is desired for interpreting word meanings expressed by the user. Reiter and Sripada's argument targets general research in text understanding within the context of natural language generation and is also applicable to sentiments or opinions where a deep understanding is as significant.

Assumption II:
An individual's opinion towards a topic is likely to be consistent within a period of time, and opinions on related topics are potentially influential to each other.

Janis and Field [1956] examined the individuality in the consistency of opinion changes by conducting real-life behavioral tests where the subjects were involved in a series of communications with a wide variety of topics. Their findings have shown that there are consistent individual differences in the matter, which indicates that the period of time an opinion may last is user-specific. This conclusion motivates the use of earlier posts of a user to assist in the analysis of the current post. Furthermore, opinions on related topics are correlated as mentioned in Ren and Wu [2013].

Assumption III:
There are connections between individuals' opinions and public opinion.

Nowak et al. [1990] simulated the interactions between the lower level units (e.g., individuals) and higher level units (e.g., social groups) regarding the social behaviors and opinions. The influence from one level to another (in both directions) is observed, which provides evidence for this assumption.

In the preliminary study (Section 5.1), the three assumptions are leveraged separately in the sentiment model by adding the associated components in the input sequence. After that, extensions of the assumptions are made to refine the model, and later implicit representation is used for the extensions.

4.1.2. The Atomic Representation

In order to emphasize the introduced assumptions, atomic representation is utilized in the preliminary study where the used features are explicitly included. For **Assumption I**, the concepts from SenticNet[14] are chosen as the signal terms for indicating lexical choices. SenticNet allows capturing implicit meaning of a piece of text by using web ontologies or semantic networks. The concepts contain conceptual and affective information. For instance, *'It is a nice day to take a walk on the beach'* contains concepts *nice, nice day, take, take walk,* and *walk beach.* In the atomic representation, concepts are used because of their simplicity in representing the text with the aim of concentrating on the influence of the additional user-related information.

For **Assumption II**, topic extraction is performed for each text. Here, the topic of a text is a set of entities that the sentiment relates to, but is not necessarily the direct target of the sentiment. For example, in the sentence *'I hate that the street is always packed with drunk people in new year's eve'*, the direct target of the negative sentiment is the *drunk people*, but the topic is a collection of the terms *street*, *drunk people*, and *new year's eve.* Since this research is set to the document-level, providing a link from the sentiment to the target does not directly contribute to the task. On the other hand, given the focus on the personalization, one's lexical choices can be potentially linked to the appearance of all the entities in a document. In this way, the influence of an opinion on related topics can also be studied.

To include the connections between individual and public opinions as described in **Assumption III**, the polarity values are taken from the SenticNet as well. The values are sentiment scores between -1 (extreme negativity) and +1 (extreme positivity) investigated in terms of four affective states (pleasantness, attention, sensitivity, and aptitude) in accordance with the hourglass emotion model as shown in Figure 4.1. They reflect a common understanding of the associated terms. The public opinions on concepts are set directly according to the polarity values, while the public opinions on topics are calculated by averaging the summed polarity values of the concepts over posts with the same topic.

[14] http://sentic.net/

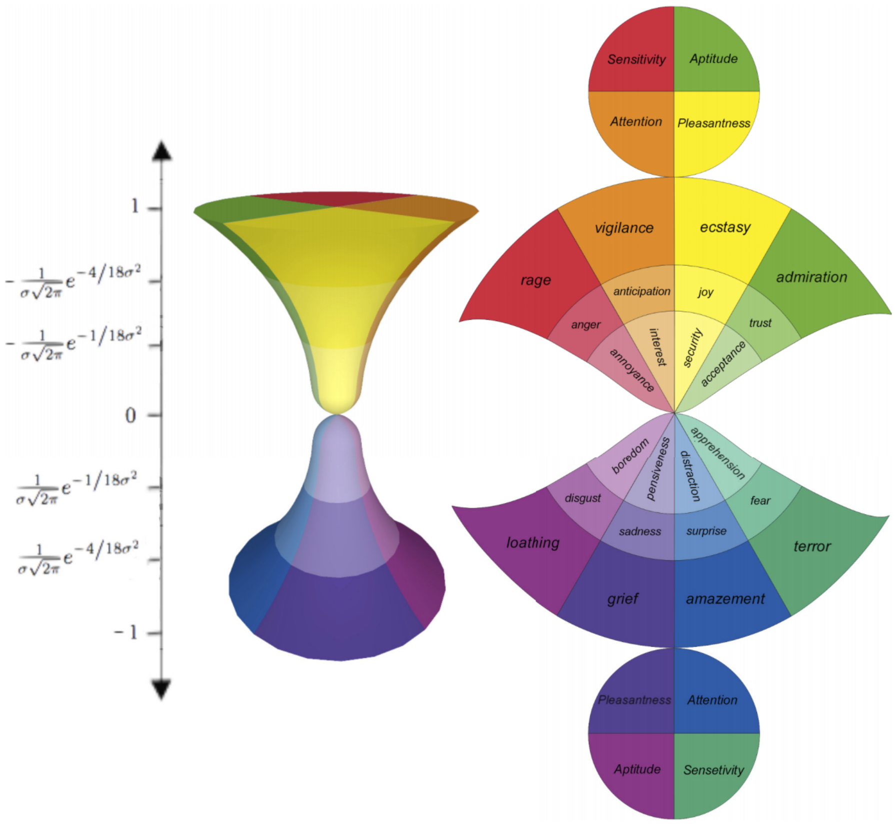


Figure 4.1.: The hourglass emotion model by Cambria et al. [2012].

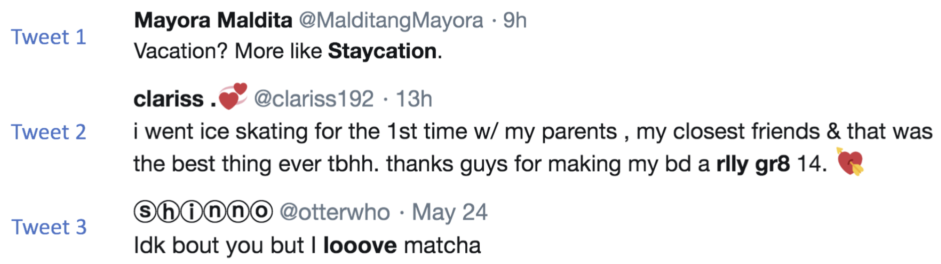
4.1.3. From the Non-canonical Language to Feature Extraction

The Non-canonicity in User-generated Text

The informal text from social networks that is targeted in this research deviates from the language standard. Such text largely affects the performance on NLP tasks [Plank, 2016]. User-generated text may contain misspellings, abbreviations, word stretches, neologisms, symbol omissions, and other types of non-canonicity. Examples of such text are shown in Figure 4.2.

In **Tweet 1**, there is a neologism 'Staycation' which means *'a holiday spent at home (or home country) or near home rather than traveling to another place'*; according to Wikipedia[15], it first appeared in 2003. In **Tweet 2** and **Tweet 3**, there are abbreviations 'tbhh' (*'to be hella honest'*, where 'hella' is a slang term for 'really') and 'Idk' (*'I don't know'*), symbol omission and substitution ('rlly gr8', which stands for *'really*

[15] https://en.wikipedia.org/wiki/Staycation



Tweet 1

Mayora Maldita @MalditangMayora · 9h
Vacation? More like **Staycation**.

Tweet 2

clariss .💞 @clariss192 · 13h
i went ice skating for the 1st time w/ my parents , my closest friends & that was the best thing ever tbhh. thanks guys for making my bd a **rlly gr8** 14. 💘

Tweet 3

ⓢⓗⓘⓝⓝⓞ @otterwho · May 24
Idk bout you but I **looove** matcha

Figure 4.2.: Examples of the use of non-canonical language in Twitter messages.

great'), and letter repetitions (e.g., 'looove' – *'love'* in the original form). Such usages are common on social platforms, and there are certain patterns reflecting relations between the user (or domain) and the usage. The information contained in the usage is representable by finer granular levels, for instance, phonemes and characters; a series of preprocessing steps are required for the representation when using coarser granular levels such as words or phrases.

Preprocessing for Feature Extraction

Concepts and Entities are coarser granular representations. The precision of extracting such terms from user-generated text largely depends on the preprocessing of the text. However, the non-canonical nature of the text makes preprocessing a very challenging task. In the preliminary study, regular expressions are used to regularize the informality.

A list of preprocessing procedures is shown below:

- Normalize the text according to a pre-defined list of replacement rules, such as chk –> check, nbd –> no big deal.
- Remove URLs, mentions, reserved words, and emoticons.
- Remove non-ASCII characters.
- Remove repetitive letters in word elongations.
- Remove repetitive punctuations.
- Remove onomatopoeia.

There are certain limitations in the implementation of regular expressions. They are sensitive to the order of applications, and all the possible scenarios must be considered exhaustively. For example, there are no words with more than two repetitive letters in English; however, if two repetitive letters were kept when encountering more, the word *'bloood'* would become *'blood'* and *'looove'* would become *'loove'*, where the former could be processed correctly but the latter could not. Additionally, one may use an external knowledge base to remedy the situation, such as to check a given vocabulary for each action.

After the preprocessing, entities are extracted from the text based on grammatical rules. Moreover, lemmatization is performed when extracting concepts — the provided list of concepts keeps the regular form of the terms. For concepts with more than one word, permutations are required for retrieving to match the terms that appear in a different order, e.g., to match *'communication ability'* and *'the ability to communicate'* with the concept *'ability_ communicate'*.

Note that the task of identifying topics or entities in the post is differentiated from the topic modeling task described in Blei and Lafferty [2009], with the latter being a separate research area which cannot be applied to this task directly.

4.1.4. Concept and Topic Embeddings

After extracting features from the text, the atomic representation must be transformed into a numerical one. Such transformation can be performed using use one-hot vectors. Since the positions of the terms do not play a role in the analysis, n-hot vectors are also applicable, where n is the number of terms extracted from the text at hand. A common issue with this representation is data sparsity — a large difference between the size of the vocabulary of all the existing terms and the number of terms in a (short) document will result in vectors with scarce entries of ones and dominating zero entries. Such a representation burdens the data storage and conveys only indexing information of the terms.

Concept Embedding

To deal with the sparsity problem in representing words or phrases, embedding methods are usually a good choice. Similar to Word2Vec [Mikolov et al., 2013] which generates word embeddings based on the co-occurrence of the words, concepts are used as the granular base and are placed at the input and output of a shallow, fully connected network (Figure 4.3(a)). Since posts from social networks are usually short messages with small numbers of concepts and the order of the concepts contains no extra information, a target concept is fed to the output layer and its context in a post is placed at the input layer as one training sample. Furthermore, the weights between the hidden layer and the output layer are taken as the embeddings of the concepts. One significant characteristic of the learned embeddings is that similar concepts are located close to each other in a high dimensional space.

Some of the reviewed academic publications suggest using another method for creating a representation space for sentiment analysis, namely, grouping words by their sentiment orientations such as AffectiveSpace [Cambria et al., 2015] and SSWE [Tang

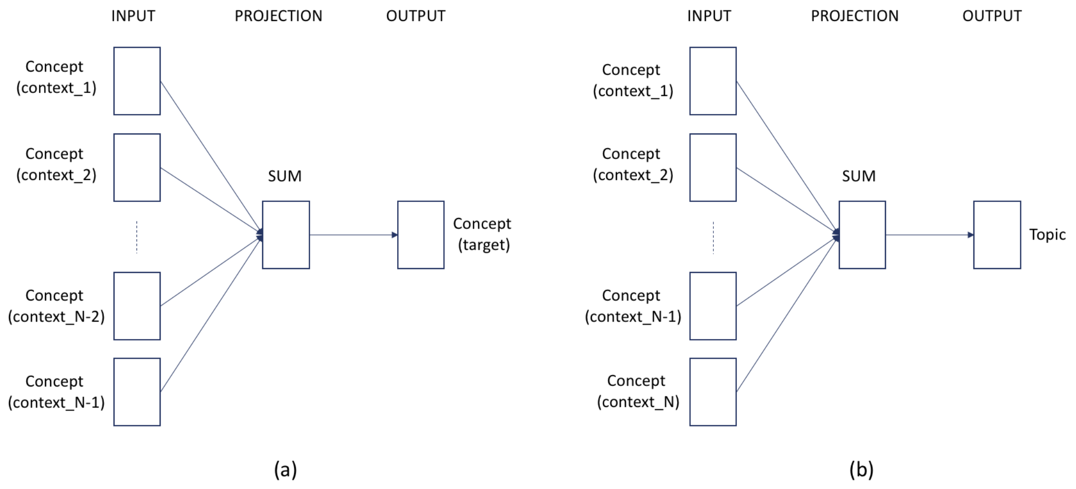


Figure 4.3.: The structure of the shallow neural network for generating concept embeddings (a) and topic embeddings (b), where N is the number of concepts in the targeted piece of text.

et al., 2014]. However, an objective representation is much more desired considering the difference between the perspectives of an individual and the public. Therefore, the embeddings are used based on semantic relations instead of sentiment relations.

Topic Embedding

Given the relationship between opinions and topics introduced in **Assumption II**, embeddings for the topics appeared in the texts are created. Similar to the concept representation described above, a shallow network is constructed with topic as target and presenting concepts as context to find embeddings for topics (Figure 4.3(b)). The network is built under the assumption that the more a concept and a topic occurred together, the more descriptive the concept is towards the topic. After the training, two topics will appear close to each other in the high-dimensional space if they are associated with similar sets of concepts. As in the previous case, the weights between the hidden layer and the output layer are used as the topic embeddings. Alternatively, the networks for learning the embeddings of concepts and topics can be merged for simplicity. Figure 4.4 illustrates a fragment of a t-SNE projection of the topic embeddings. Related topics e.g. ‘google’, ‘microsoft’, ‘twitter’, ‘apple’, and ‘moto g’ are located close to each other (upper right corner).

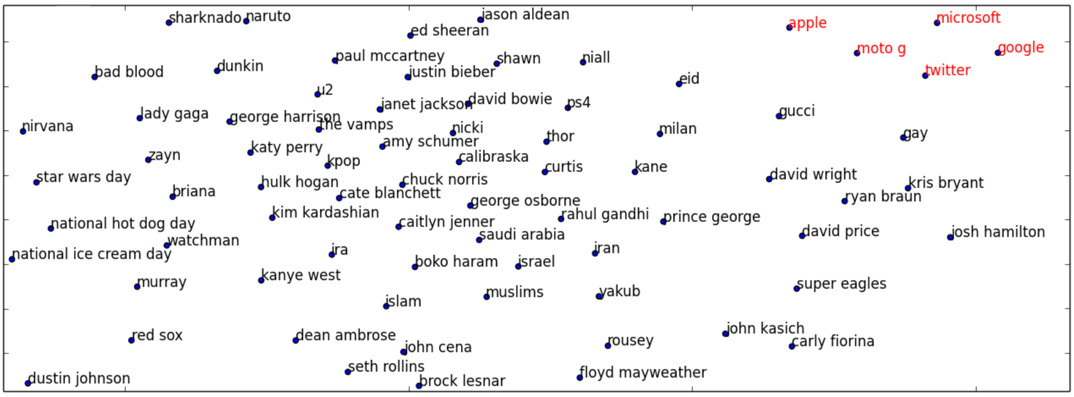


Figure 4.4.: A fragment of a t-SNE projection of the topic embeddings trained on the combined corpora (Section 7.1.2). Topics with greater similarity (e.g. terms highlighted with red color) are located closer to each other.

4.2. Representation in Different Granular Levels

The level of granularity in text representation plays an important role in understanding the text. There are works based on characters [Dos Santos and Gatti, 2014], bag-of-words [Whitelaw et al., 2005], n-grams [Bespalov et al., 2011], or concepts [Cambria and Hussain, 2015].

4.2.1. Concept-level

Concepts, as mentioned in Section 4.1.2, contain conceptual and affective information and can be seen as the 'signal terms' regarding lexical choices. The documented concepts serve the representation as a static knowledge base where the information is expressed in an abstract way — the common sense of the text is represented without a connection to any event, while events can be time-sensitive and the meaning of the concepts associated with the events can vary. In reality, trends also play a role in the construction of the concepts for which the knowledge base has to be up to date. The lack of real-time adaptation of the concepts can be compensated by exploring other information types. In the atomic representation, entities extracted from the text can be used as additional features (supplements) to support the understanding of the use of concepts. Similarly, negation cues have the ability to invert the orientation of a sentiment, thus they should also be considered as features [Jia et al., 2009]. Lexicons of negation cues such as the ones described in Reitan et al. [2015] can be used for this purpose.

4.2.2. Word-level

Word-level representation corresponds to the use of word-based n-grams where the common choices of the value of n are 1, 2, and 3 after which the computational cost is too high [Bespalov et al., 2011]. Note that concepts are not necessarily coarser granules than words — there are concepts with one word; however the use of uni-grams in the word-level representation is emphasized so that words are mostly finer than concepts. Meanwhile, the most distinguishing feature that makes words different from concepts is the information carried by the representation: Word-level representation has a verbatim nature which considers each word equally while concept-level representation scans the text for appearances of the pre-defined sentiment-related terms.

Representing short texts with words has the same data sparsity issue as the concept-level representation. In order to represent the information more efficiently (to reduct the dimensionality), embeddings are created by adding an embedding layer in the neural network or mapping the words to the pre-trained word vectors. The merit of using an embedding layer is that the learned word vectors are shaped by the final outputs (labels) that associated with the specific task — the representation of the words can carry information that contributes to the task. However, such vectors are less transferrable because of the task specificity. Another drawback is that for the tasks which require human labor in labeling, the words may be under-represented due to the possible size limitation of the used data. On the contrary, the alternative – using pre-trained word vectors – provides a more general-purpose representation which is less restrained by the data size because the vectors can be trained using an unsupervised technique. Pre-trained word vectors such as GloVe [Pennington et al., 2014] and Word2Vec [Mikolov et al., 2013] generate embeddings according to the co-occurrences of the words. In Bojanowski et al. [2017], the word vectors are enriched with subword information. Moreover, it is also favorable to fine-tune the vectors with the task at hand [Kim, 2014]; the method used for fine-tuning can be designed as needed where dense, convolutional, and recurrent structures are all applicable.

4.2.3. Character-level

Characters represent a finer granularity compared to words and concepts. Character-level representation is more resilient to morphological variations, which can be beneficial for many NLP tasks — especially for processing user-generated text where such representation is able to capture details that are largely neglected by other representations. Similar to word-level representation, character-based n-grams are utilized in some works; however, there is no clear distinction between 'character n-grams' and 'subword units' except that the choice of n may be restricted in the subwords [Bojanowski et al., 2017].

Many neural-based approaches can be used to create character embeddings. For instance, Dos Santos and Gatti [2014] employed a convolutional structure that leverages character-level information to perform sentiment analysis of short texts; Peters et al. [2018] proposed a deep contextualized representation (ELMo) that takes characters as input and generates embeddings for the text by exploiting a deep bidirectional language model. The works using character-level representation are usually compared to the word-level representation and offer competitive results in many linguistic tasks.

4.2.4. Combined Granular Levels

Given the advantages and disadvantages of the representations at different levels, a combined representation can be beneficial to the task as the implicit connections between different granularities can be learned. For example, *'destructive_ behavior'*, which is an entry in the concept list with a sentiment score of -0.78, can be linked to *'destructive_ behaviour'* which has a British spelling of the word 'behavior' and is not recorded in the pre-defined list. In this way, the knowledge learned at one level is transferrable to another level. The downsides of the combination are the elongated representation of text and the possibility of retaining redundant information. Furthermore, when a deep network is utilized to generate a representation, the time needed for the (re)training process can increase dramatically.

Chapter 5

Analysis of the Past with Recurrent Neural Network

One way to understand the perspective of a user is to look into the user's past. In the setting of social networks, the information of a user's past can be reflected in the posts that the user has published earlier (before the time of the investigation). In this chapter, I first introduce a simplified model that is used in the preliminary study, and then propose an advanced model with revisions made based on the observations from the preliminary study. The two models differ by the design of the input sequence and the use of the attention mechanism in the advanced model. Note that to explicitly leverage the assumptions described in Section 4.1.1, only atomic representation is used in the preliminary model. In the last section, an information selection algorithm is introduced as a refinement that enables the model to relate to all the history of a user and prevent significant information loss.

5.1. The Preliminary Model

The structure of the preliminary model is shown in Figure 5.1. The model has a many-to-one structure, and the central part of the model is three-layered RNNs (h^1, h^2, and h^3). Each of the hidden layers contains a number of LSTM cells which help to preserve and extract valuable information from temporal / sequential data — in this case, from a series of posts of a user.

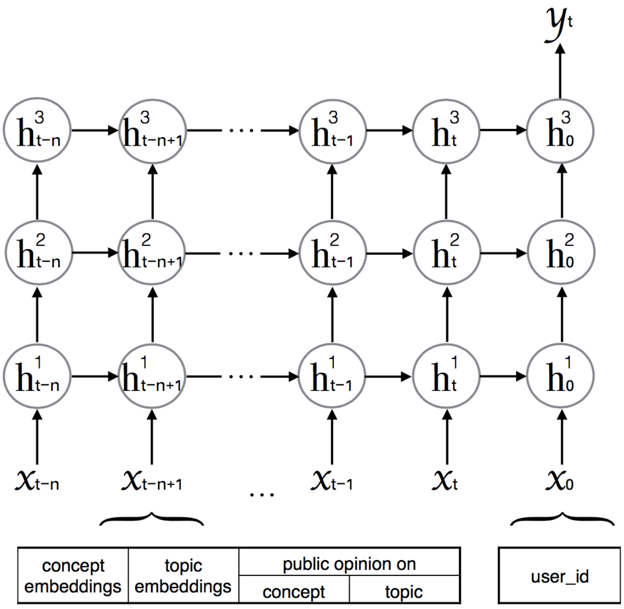


Figure 5.1.: The preliminary sentiment model with RNNs and two types of neurons at the input layer: The user identifier (x_0) and the post of the user at a specific time point (x_{t*}) [Guo and Schommer, 2017].

5.1.1. Design of the Input Sequence

A Simple Solution for Data Sparsity

Personalized models generally suffer from the issue of data sparsity caused by different frequencies of users posting messages on the social platforms. To deal with this issue, I take inspiration from Johnson et al. [2017] where an additional token is added to the input sequence to indicate required target language for multilingual neural machine translation. Here, a user identifier is added in the input to indicate the expresser of the text. In this way, the individuality of a certain user can be captured by the model, while at the same time, the relations between users can be learned automatically. More importantly, the data sparsity issue is resolved since only one model is required. In the preliminary model, the user identifier is added in a number of individual neurons.

Input Construction

As shown in Figure 5.1, the input sequence of the recurrent neural network consists of two parts. The first part is the identifier of the user who published the post. Instead of building a model for each user, a user index x_0 is added at the end of the input sequence and encoded as a one-hot vector. This enables the network to learn user related information and compare different users. All users with only one post are assigned the same index because no historical relations can be learned for them. In this way, these users are considered as one user that acts aligned with the public with fluctuations. This solution also saves the space for storing the user index for these users. When the proposed model is used upon another sentiment model, these users can be excluded until there are at least two posts from the same user. For users with more than one post, their sentiments towards different topics are learned individually. This part is required to examine the effect of using **Assumption I**.

The second part of the input sequence corresponds to the current and the past posts of a user, and each post contains four components: concept embeddings of the post $E_{concept}$, topic embeddings of the post E_{topic}, public opinion on the concepts $P_{concept}$, and public opinion on the topic P_{topic}. Concept and topic embeddings (Section 4.1.4) are used to introduce **Assumption II** to the network. In this model, public opinions are pre-defined and extracted from an external source as described in Section 4.1.2. By applying the components of public opinions $P_{concept}$ and P_{topic}, the effectiveness of **Assumption III** can be examined in the network. Hence, the post of the user at a specific time point (x_{t*}) is represented by a concatenation of four components $x_{t*} = [E_{concept}\ E_{topic}\ P_{concept}\ P_{topic}]_*$.

In practice, the required dimensions of the two parts can be of different lengths. To keep a consistent length for each input node, either more than one node is allocated to the user index or padding is performed for the second part (the posts). The latter is used in the experiments to enhance the impact of the earlier posts.

5.1.2. The Personalized Recurrent Network

To utilize the preliminary model (Figure 5.1), the posts are first sorted by the user identifier, and then by the creation time of the posts. Thus, the input sequence X is a matrix of $[x_{t-n}, x_{t-n+1}, ..., x_{t-1}, x_t, x_0]$ where x_t is the current post, x_{t-*} are the posts published before it by the same user x_0 ($*$ indicates the index of the earlier post), and n is the number of past posts considered. Note that in the preliminary study, the different gaps between two successive posts x_{t-*} and x_{t-*-1} are not explicitly modeled. For the user with more than one but less than $n+1$ posts, a number of vectors with zeros are padded before the earliest post of the user. The output y_t is the sentiment orientation of the current post, which can be positive, negative, or neutral. Both x_* and y_t are vectors, and n is a constant number. The LSTM cell used in this architecture follows Graves et al. [2013], however without using peephole connections. As reported in Greff et al. [2016], there is no significant difference in the performance using the peephole connections or other tested modifications.

Let $(i_k, f_k, C_k, o_k, h_k)$ denote respectively the input gate, forget gate, cell memory, output gate, and hidden states of the LSTM cell. The update of the cell state is then described with the following equations:

$$i_k = \sigma(W_i[x_k, h_{k-1}] + b_i) \tag{5.1}$$
$$f_k = \sigma(W_f[x_k, h_{k-1}] + b_f) \tag{5.2}$$
$$C_k = f_k \odot C_{k-1} + i_t \odot \tanh\left(W_C[x_k, h_{k-1}] + b_C\right) \tag{5.3}$$

where σ denotes the sigmoid activation function, $k = 0 \rightarrow x_0 = \mathit{user\_id}$ for the input node at the end of the sequence, $k = t$ for the previous input node indicating the current post, and $k = t - *$ for other input nodes corresponding to the earlier posts. With the help of the gates i_k and f_k, the cell k selects new information and discards outdated information to update the cell memory C_k.

For the output of the cell,

$$o_k = \sigma(W_o[x_k, h_{k-1}] + b_o) \tag{5.4}$$
$$h_k = o_k \odot \tanh\left(C_k\right) \tag{5.5}$$

where o_k selects information from the current input and the hidden state, and h_k combines the information with the cell state. Moreover, $x_* = [E_{concept}\ E_{topic}\ P_{concept}\ P_{topic}]_*$ is set for $* \neq 0$, as introduced in the last section. Such concatenation of components has been shown effective by Ghosh et al. [2016]. The information flow in a single LSTM cell is illustrated in Figure 5.2.

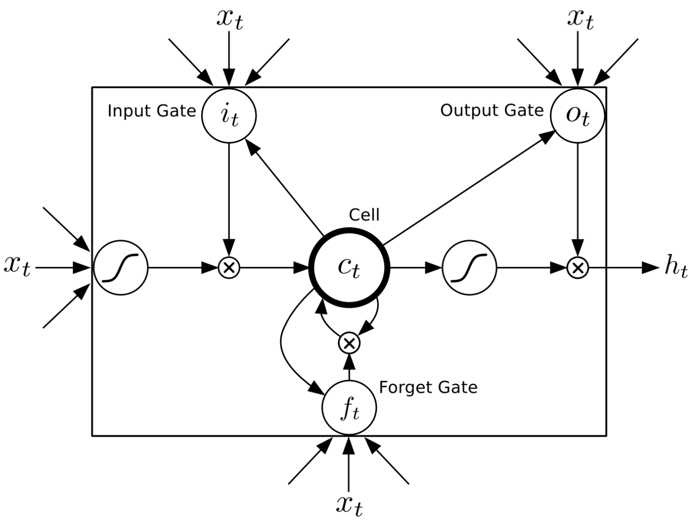


Figure 5.2.: An exemplary LSTM cell for $k = t$ [Graves et al., 2013].

With this design, the network is able to recognize a user identifier from the input sequence so that the drifting distance between user opinions and public opinions can be learned by accessing information from the past. This approach offers a better alternative for implicit or isolated expressions. For instance, the post *'This totally* *changes my mind about Apple products.'* contains unclear sentiment orientation that the expressed sentiment can only be determined by knowing the past opinion of the user about *'Apple products'*. For the posts with no concepts extracted, the network is able to make predictions by comparing the topic of the post with other posts that are associated with the same topic. Similarly, for posts with new (unseen) topics, the presenting concepts are considered.

Another distinction of LSTM is that it does not suffer from vanishing or exploding gradient problem [Hochreiter, 1998] unlike simple recurrent networks. This is due to the implementation of an identity function, which indicates whether the forget gate is open or not, keeping the gradient constant over each time step. This trait of gated networks enables the model to learn long-term dependencies of concepts and topics over time.

5.1.3. Limitations of the Preliminary Model

The preliminary model is a simplified network that is used for evaluating the effectiveness of introducing the mentioned assumptions in determining sentiment. Although experiments have shown positive results (Section 8.1.2), there are several aspects that can be modified to improve the performance.

First, the input of the network takes two different types of information – the user index and the tweet representation – at different nodes, and the network has to react with the same set of parameters. This setting makes the network more difficult to train. Second, the applied method of representing the posts is not sufficient to include necessary information of the text. Negation cues, as signal terms, can invert the polarity of a sentiment, hence they should be included in the representation. Moreover, the single topic given for each post can be unilateral since multiple entities are mentioned in some cases. Furthermore, the influence of the past opinions can be affected by time, i.e., the time gaps between the posts of a user can reflect the importance of the past opinions.

5.2. The Advanced Model

Hierarchical networks are used to transfer information from a lower level to a higher level as discussed in Section 3.1. Here, a post-level representation and a user-level representation are considered, and the embedding networks in the preliminary model are merged with the recurrent network so that the representations of the posts are learned automatically through the network by the sentiment label y_t (Fig. 5.3).

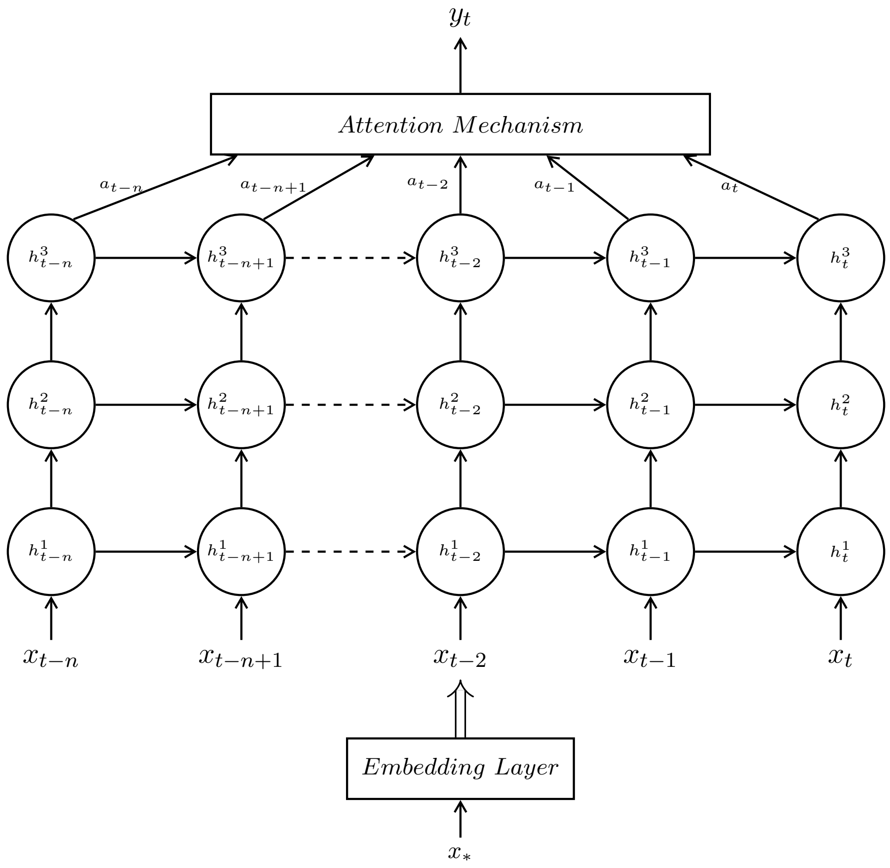


Figure 5.3.: A hierarchical, personalized sentiment model with recurrent neural network and attention mechanism.

5.2.1. Refined Input Structure with Atomic Representation

In the input sequence of the hierarchical network, each post x_* is represented by a set of concepts, entities, negation cues, and the user identifier while using atomic representation. The concepts are from the same knowledge base as the preliminary model, whereas entities are extracted from the text instead of using the single topic so that the relation between the concept and the target can be more flexible. Additionally, explicit negations are included in the input based on a pre-defined list of rules (Section 4.2.1). As a better alternative, instead of occupying individual nodes at the input layer of the recurrent network, the user identifier is placed in the post representation (concatenated with other components in each node) to obtain consistency in the inputs.

There are a number of posts from which no explicit concepts, entities or negation cues can be extracted, and such posts are represented by the occurring components. There may be extreme cases when a post is simply represented by the user identifier, and in such situations earlier posts will play an important role in predicting the sentiment of the current post. Moreover, public opinions can be redundant, because the opinions of majorities can be learned automatically given enough training samples from a sufficient number of users. At the same time, whether a person's opinions tend to align with the public opinion can be learned directly. Since the representation is concept-based, the order of words appearing in the text does not play a role in the representation. As a result, a single embedding layer is applied to map the terms into a dense, low-dimensional space.

5.2.2. Input Structure with Pre-defined Input Formulation

As discussed in Section 4.2, different granularities can be applied to represent the text. Different from atomic representation, finer granules or pre-trained embeddings are fed to the RNNs in other ways. For example, when GloVe word vectors [Pennington et al., 2014] are used, one can simply take the sum of the vectors dimension-wise or apply a fully connected layer to generate a vector to be used at the input node of the higher hierarchy; more sophisticatedly, a bidirectional RNN can also be used since the order of words plays a role in the understanding of the text. When using ELMo representation [Peters et al., 2018], the generated vector of a sentence (here, a post) can be fed to the following network directly. To use the combined representation, different information types or granules are processed individually, and are concatenated afterwards to produce one vector as the post representation. The method to formulate the text representation in constructing the input sequences is named *Input Formulation*.

To elucidate, an input sequence X_i consists of an entry of the current text at the end of the sequence (which contains textual information and the encoded user index) and a number of earlier posts by the same user (contextual information), i.e.,

$$X_i = [H_{i-n}, ..., H_{i-2}, H_{i-1}, F_a(x_i)] \tag{5.6}$$

where

$$H_j = \begin{cases} F_a(x_j) & \text{if } \ u(x_j) = u(x_i) \\ 0 & \text{else} \end{cases} ,$$

n is the number of earlier posts considered, F_a is the input formulation chosen beforehand, and u is the user index of the text.

5.2.3. The Personalized Network with Attention Mechanism

The stacking of the hierarchical network happens between the generation of the post embeddings and the construction of the input sequence for the recurrent neural network. Similarly, a many-to-one recurrent network with LSTM cells is used in the model. With a consistent formulation of the representation at each input node, the network can be trained efficiently. Again, the input sequences are first sorted by the user identifier and afterwards by the creation time of the text.

Attention mechanism is widely used in NLP [Yang et al., 2016; Vaswani et al., 2017]. In the preliminary model, all the information learned in the network is accumulated at the node closest to the sentiment label (h_0^3), which can be treated as an embedding for all seen posts in an input sequence. Although LSTM has the ability to preserve information over time, in practice it is still problematic to relate to the node that is far away from the output — LSTM tends to focus more on the nodes that are closer to the output y_t. There are studies that propose to reverse or double the input sequence [Zaremba and Sutskever, 2014], however in many cases, attention mechanism can be a better alternative.

Here, the hard- and the soft attention (alignment) mechanism are differentiated [Bahdanau et al., 2015]. The hard attention implements a stochastic sampling to focus on specific regions (positions), which consequently masks out other regions; the soft attention implements a deterministic method that gives a weight to each region. Given the differentiable property of the soft attention, it is a more popular choice when applied with neural networks. A traditional (soft) attention model is defined as:

$$u_i = \tanh\left(W_t h_i + b_t\right) \tag{5.7}$$

$$a_i = u_i^T w_s \tag{5.8}$$

$$\lambda_i = softmax(a_i) h_i \tag{5.9}$$

$$v = \sum_i \lambda_i \tag{5.10}$$

where

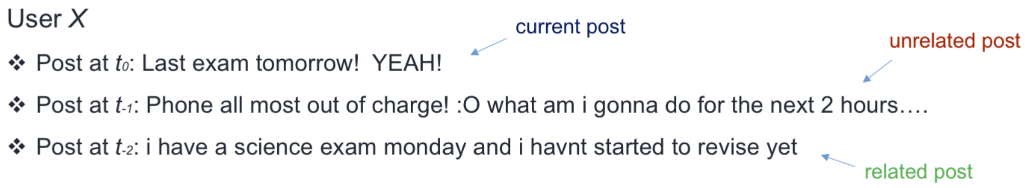
$$softmax(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \quad .$$

λ_i is the attention output at the specific time t_i with the same dimension as h_i, and v sums up the output at each t_i dimension-wise and contains all the information from different time points of a given sequence. The context vector w_s can be randomly initialized and jointly learned with other weights in the network during the training phase.

5.3. Input Selection from the Post History

For users with various lengths of history, there can be problems with regard to the lengths of the input sequences in each training batch. Practically, people use ‘padding’ or ‘bucketing’ to handle this issue. The former sets the length of the input sequence to the maximal length observed in a corpus, and the shorter ones are padded with zeros. This method is not feasible in this task, because such a representation can be very sparse given the number of a user’s posts ranging from a few to a few thousand.

The latter groups the input sequences by ranges of lengths which can be seen as a relaxation of the former method, but zero padding is still needed depending on the size of each 'bucket'. Previously, the length of the input sequence (the time steps T) was chosen empirically and was used for all the users, i.e., the same number of earlier posts are considered no matter how frequently the user posts. For the users with history longer than T, the information before is lost. There are cases when the recent posts of a user are unrelated to the current one while related ones have appeared long before — the earlier posts taken by the system only provide noise. An example is shown in Figure 5.4 where the current post at t_0 and the past post at t_{-2} from User X are about the same topic (taking exams) and the past post at t_{-1} is not related to the current one, thus the post at t_{-1} should be filtered out.



User *X*

❖ Post at t_0: Last exam tomorrow! YEAH!

❖ Post at t_{-1}: Phone all most out of charge! :O what am i gonna do for the next 2 hours....

❖ Post at t_{-2}: i have a science exam monday and i havnt started to revise yet

Figure 5.4.: An example of the three consecutive posts from the same user where an input selection can be helpful.

In this thesis, a selection technique is specially developed for this task and provides a more flexible solution for this problem. The proposed approach is based on an extension of the assumption about opinions on related topics (**Assumption II**): The current opinion is affected more by a past opinion on related topics than on unrelated ones. To leverage this assumption, the relatedness of topics between posts of a user is analyzed by calculating the distance between the topic embeddings. With the algorithm, the network is able to take selected posts from the entire history of a user based on the similarity.

5.3.1. Similarity measures

In order to compare the similarities between the topics, the topic embeddings are used as described in Section 4.1.4. Five measures are concerned to calculate the similarity or distance between two sets of topics. Euclidean distance (**ED**) measures the straight-line distance between two terms; Manhattan distance (**MD**) measures the sum of the absolute differences of the coordinates between two terms; cosine similarity (**CS**) measures the cosine of the angle between two terms. These three measures are calculated dimension-wise after finding the centroid of the topics in each set. The earth mover's distance (**EMD**) [Pele and Werman, 2009] measures the cost to transport a term to another. In the experiment, EMD is calculated in two ways where one is to measure the distance between the centroids of the topics while the other is to compute directly between both sets since it is capable of processing documents with different lengths.

Furthermore, the word mover's distance (**WMD**) [Kusner et al., 2015] is used as well, which is a special case of EMD implemented with GloVe word vectors [Pennington et al., 2014]. When implementing EMD and WMD, Euclidean distance is chosen as the ground distance.

5.3.2. The Selection Procedure

A selection procedure is designed to overcome the problem with the information loss when a user has a history longer than the pre-defined number of time steps. All the previous posts of a user are considered in the process, which provides the RNN a number of posts that are related to the current topics.

Algorithm 1 Input Sequence Generation

```
Input: Corpus with attributes: [user, time, topic, content];
                                  ▷ $content = [user\_id, topic, negations, concepts]$
       Distance measure $D(a, b)$;          ▷ Distance between element $a$ and $b$
       Threshold for the measure $\delta$
Output: Input sequences $X$ with shape:
        [length of the corpus $N$, number of time steps $T$]
Initialization: $X[:][-1]$ = corpus[:][‘content’], $k = 1$
for $i = 0$ to $N$ do
    if corpus[$i$][‘user’] = corpus[$i-1$][‘user’] then
        $k = k + 1$
        if $k \leq T$ then
            $X[i][:-1] = X[i-1][1:]$
        else if corpus[$i$][‘topic’] = [‘’] then
            $X[i][:-1] = X[(i-T+1):i][-1]$
        else
            distances = [$D$(corpus[$i$][‘topic’], corpus[$i-j$][‘topic’])
                        for $j = 1$ to $k-1$]
            $l = min(len([m$ for $m$ in distances if $m \leq \delta]), T)$
            selected = $argsort$(distances)[:$l$]
            if $len$(selected)$\neq T$ then
                selected.$extend$([$i-n$ for $n = 1$ to $T$ if $i-n$ not in
                                  selected][:($T - len$(selected))])
            end if
            $X[i][:-1] = X$[selected][$-1$]
        end if
    else
        $k = 1$
    end if
end for
```

Algorithm 1 shows the selection procedure. The distance measures are as listed in the last section, while for the cosine similarity, a reverse of the value is used since the more similar two terms are, the closer they are in the vector space. Given a fixed number of time steps T in a recurrent network, a similarity threshold δ, and a target text x_i from user u, the recent T earlier posts (x_j) that have a similarity score larger or equal to δ are chosen. Note that the preceding posts are used without a selection when the number of preceding posts by the same user is less or equal to the length of the input sequence, or when no topics can be extracted for the current post. Additionally, when the number of the selected posts is smaller than the length of the input sequence T, the algorithm takes the preceding posts of the current one as complements according to the creation time (prioritizing on the recent ones). After the selection, the posts in each sequence are ordered by time.

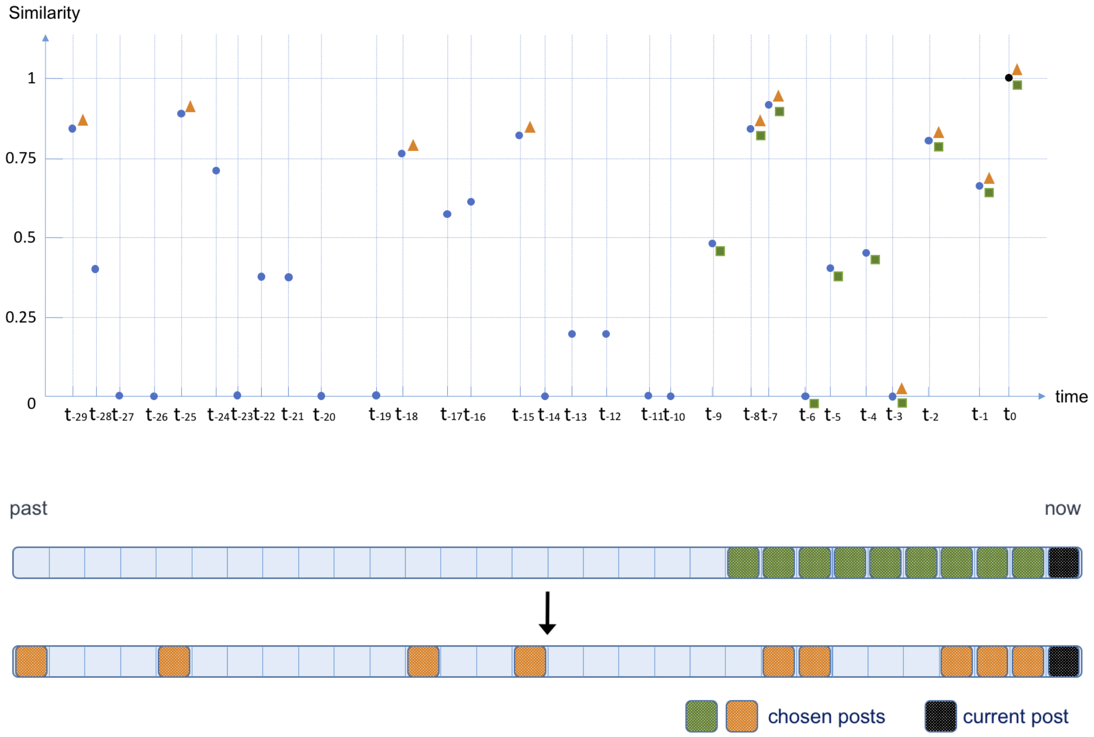


Figure 5.5.: An example of constructing an input sequence with the selection method (marked by orange color) and without (marked by green color). The current post is marked by the black dot at t_0, the number of time steps is 10, and the threshold of the similarity measure is set to 0.75.

As shown in Figure 5.5, the user has 30 posts, and the ones that are close in time are chosen when no selection method is used. With the algorithm, 7 posts from the past – with similarity above the threshold – are selected while t_{-1} and t_{-3} are added as well to fill the empty slots in the input sequence. Therefore, the posts that are created recently with unrelated topics are replaced with related posts generated further before. The selection procedure is executed more frequently with a smaller value of T, and the number of execution will keep growing while the user continues posting on the platform in the future (as in reality) — reshaping the model is not required.

Chapter 6

Modeling Information Decay in Sentiment Analysis

The information emerging on social networks has a dynamic nature that associates with temporality. By analyzing a person's past, semantic knowledge on a person's preferences in lexical choices (**Assumption I**) as well as affective knowledge on particular topics (**Assumption II**) can be acquired (Section 4.1.1). In this chapter, the time factor described in **Assumption II** is investigated: *An individual's opinion towards a topic is likely to be consistent within a period of time, and opinions on related topics are potentially influential to each other.* It can also be deduced from the assumption that the consistency of an opinion is time-sensitive and so does the consistency of the opinion on a related topic. In particular, this research is inspired by Nguyen et al. [2012] in which the authors have observed that

> *'... events occurred in the past have decayed impact on future sentiment. Using features extracted from too large history window will suppress important features that happened immediately before the prediction time.'*

The observation was made through a series of experiments conducted within a comparably short period of time, and the experiments were designed targeting the change of public opinions. It indicates that the past events indeed have an impact on future sentiment, and the period of time that a sentiment stays consistent can be shifted according to the time gap between the past event and the prediction. As an extension of the assumption and the observation, it can be argued that the opinion at a specific time point is affected more by recent opinions that contain related content than the earlier or unrelated ones. Moreover, the work of Janis and Field [1956] has shown that people have different consistencies in retaining opinions, therefore, a mechanism that models the decay of information individually by applying user-specific Hawkes processes is proposed (Section 6.2).

6.1. Associating Time Information in the Sentiment Model

Motivated by the findings in social psychology on the dynamics of opinions, I intend to associate time information in the analysis of sentiment. While targeting public opinions, Nguyen et al. [2012] has shown that the impact of the past declines in the course of 20 hours. This indicates the significance of considering information decay when using historical data to benefit the prediction. However, it can be suggested that the decline may happen slower when it comes to individuals — people have different focuses on topics and their statuses only get updated when the events fall in their interests. Therefore, individual changes can take longer time spans and can be more irregular. In this section, some possible methods to incorporate time in modeling the sentiment are discussed and a novel approach to capture the decay effect over time is proposed.

6.1.1. Technical Aspects in Temporal Data Modeling

In order to introduce the time information in the model, technical possibilities in fulfilling the task are explored. Based on the preliminary study, different model designs are considered where a basic structure is used as fundament. Here, following the introduction in Section 3.3, the technical challenges pertaining to the specified task and the basic model are explained, and the variants will be presented and compared in the following sections.

Limitations in Existing Works

Although traditional RNNs, as the central part of the sentiment model, are popular for extracting patterns from temporal sequences, they do not have the ability to analyze irregularly emerging events by design. Naturally, such events can be separated into different time intervals, and zero-padding can be applied for empty slots; however, this solution is infeasible when the emerging of events is hardly predictable. The same problem persists with attention model — despite its capability of relating to the past in a more flexible manner, the attention is allocated based on the content of the posts regardless how long the posts have been published.

Technically, there are very few works that investigate the different gaps between nodes in recurrent neural networks. As mentioned in Section 3.3, in order to model asynchronous events, Neil et al. [2016] proposed a phased LSTM, which altered the design of the traditional LSTM by adding a time gate. However, the phased LSTM is not suitable for the task of this research because

1. in their approach, the time gate is triggered by periodic oscillations while modeling sensory events, whereas the publishing of a post is not systematic — the expected time gaps are highly various;
2. the time gap is not connected with other information in the phased LSTM cell, whereas here, the impact of the time gap only matters when an earlier post and the current post are related.

Since existing approaches fail to match the requirements of this task, alternatives that concern precise time points and the connection between the texts are to be explored.

6.1.2. The Basic Sentiment Model

To examine the modeling of temporal data, the advanced sentiment model introduced in Section 5.2 is taken as the basic model. The basic model can be demonstrated in the form of a simple structure with a conventional ensemble of embedding – recurrent – attention blocks (Figure 6.1).

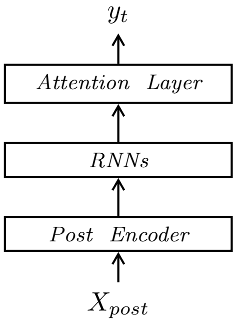
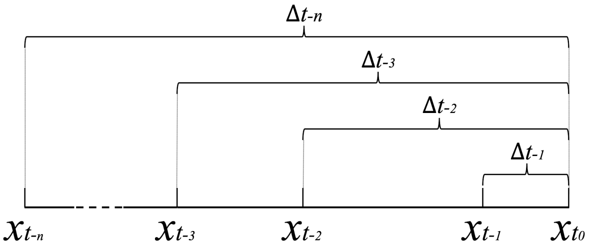


Figure 6.1.: The basic personalized sentiment model. X_{post} corresponds to a number of posts at the past and the current time points. y_t is the sentiment label at the current time.

The *post encoder* transforms the input posts into vectors according to the applied *input formulation*. For instance, a fully-connected layer can act as an encoder when atomic representation is used. The RNNs follow the description in Section 5.1.2 and the attention layer follows Section 5.2.3. The basic model is extended in three ways in order to test experimentally the influence of time gaps on personalized sentiment modeling. The same basic structure and input formulation are used for each extended model. Moreover, the atomic representation (Section 5.2.1) is used primarily to study the effectiveness of the methods. The extended models are designed to include the time information in the training process such that the time gap interacts with the content of the post while they jointly influence the prediction. Details will be discussed in the following sections.

6.1.3. Integrating Time in the Input

One of the intuitive ways to include additional features is to add them at the input. Likewise, the model in Figure 6.2 uses a *time encoder* – a fully connected layer – at the input to embed different time gaps. Note that the post encoder and the time encoder have to operate separately, since their inputs have different distributions. A time gap Δt_i at time t_i is the time difference between the earlier post x_{t_i} and the current post x_{t_0} as shown below:



The time gap is calculated as the number of hours between the past and the current post as a float positive value. The choice of time unit (hour) refers to the use in Nguyen et al. [2012] for a similar task. x_{time} consists of the time gaps at all the time steps. Afterwards, each encoded time gap is concatenated with the encoded post representation according to the timestamp. The concatenated sequence is fed to the RNNs so that the network is able to learn the connection between the content of the posts and their publishing time. The time embeddings act as an auxiliary input in comparison to the basic model in Figure 6.1.

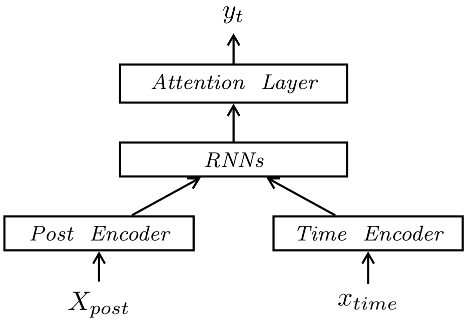


Figure 6.2.: The personalized sentiment model with time embedded in the input, where x_{time} corresponds to the time gaps between each past post and the current post.

Given the difference in user frequencies with respect to the postings, the value of Δt_i can be highly various. Hence, it is necessary to perform a normalization in order to transform the values into a certain range. For that, a sigmoidal function which normalizes the value of Δt_i is applied in the time encoder. After concatenating the output of the time encoder with the post vector, the time information is treated by the

RNNs as a feature with no distinctions from others. This means that the model has to learn the decay behavior of the information by analyzing the dependencies between the time feature and the text related features. Theoretically, such dependencies can reflect the information decay implicitly; however, since the behavior is user - and topic-various, the learned information will be predominantly determined by the posts appeared in the training data. A large number of posts published by frequent users can be helpful to generalize the modeling of the behavior.

Because the user identifier is also included in the input (Section 5.2.1), individual differences regarding the decay behavior can potentially be captured by the network as well. Nevertheless, the implicit relations are difficult to inspect and visualize in this method while all the information is combined (or concatenated) before applying the recurrent network.

6.1.4. Integrating Time in the Attention

Another option is to add the time information in the attention model. In Figure 6.3, the time gaps are encoded in the same way as before, however the output of the time encoder directly influences the attention layer by reshaping the attention value at each time point. Recall the original equation used in the traditional attention mechanism (Equation 5.9):

$$\lambda_i = softmax(a_i)h_i$$

which is now replaced by the following equation:

$$\lambda'_i = softmax(a_i E(\Delta t_i))h_i \tag{6.1}$$

where $E(\Delta t_i)$ is the i-th encoded time gap. In this way, if an information decay can be learned by the time encoder, the impact of the attention value will be decreased when the time gap is large. Again, v sums up λ'_i over time step i.

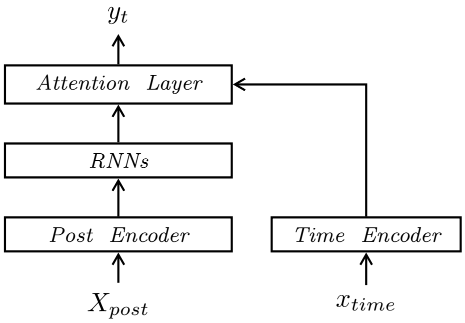


Figure 6.3.: The personalized sentiment model with time embedded in the attention layer.

Since $a_i = u_i^T w_s$ (Equation 5.8), after the replacement, we have $a_i E(\Delta t_i) = u_i^T w_s E(\Delta t_i)$ as the input of the $softmax$ function, thus we can also see $w_s E(\Delta t_i)$

as an extended context vector from w_s. This indicates that the context vector is not randomly initialized anymore — it is initialized by the encoded time gaps. Moreover, w_s can be seen as a scaler for the encoded time gaps, which describes how much the time information affects the information contained in h_i and u_i. Similar with the approach described in Section 6.1.3, no explicit decay functions are applied in the network. The implicitness of the decay behavior undermines the inspectability; however, comparing to embedding time in the input, the influence of the time factor can be more effective on the prediction by placing the time variable closer to the final output.

6.1.5. Introducing Time Factor with Hawkes Process

The methods introduced in the last two sections utilize a time encoder to embed the time information in the system. The relation between the post information and the time information is learned implicitly through the use of training weights in the network. In contrast, an alternative is proposed which models the decay behavior explicitly by using the Hawkes process (following the heuristic discussion in Section 3.3.2). The structure of the sentiment model with Hawkes process is shown in Figure 6.4.

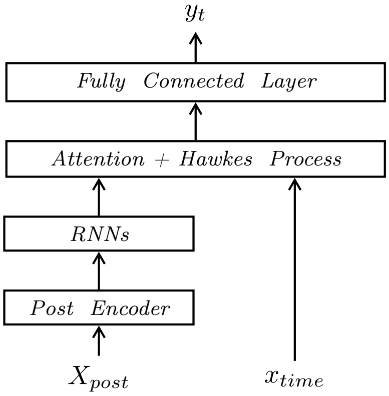


Figure 6.4.: The personalized sentiment model with time introduced in the Hawkes process.

Hawkes process can be seen as an added function component which reshapes the original process based on the provided time variable. Thus, the time input in this model is used directly at the attention layer without a time encoder — the post information that can be passed along the network is explicitly mediated by the time information in the Hawkes process. Since the time information is used separately from the post information, it is easier to observe the behavior of the process as well as individual differences in the matter.

6.2. From Universal Hawkes Process to User-specific Hawkes Process

In the last section, I have elucidated the possibilities to realize the modeling of information decay in the network. Given the explicitness in simulating the time effect, different ways concerning the use of the Hawkes process in the neural-based system will be explored. A method that considers individuality in the decay behavior will be presented in the end.

6.2.1. Hawkes Process with Attention and an Empirical Setting

In order to apply Hawkes process in the neural network, the effect of past opinions is reshaped within the attention mechanism. Recall in Section 3.3.2, the conditional intensity function of Hawkes process is

$$\lambda^*(t) = \lambda + \sum_{t_i<t} \alpha e^{-\beta(t-t_i)}$$

and the last step in the attention model is to take a summation over the information in all the time steps (Equation 5.10):

$$v = \sum_i \lambda_i$$

Here, the exponential decay factor is added at each time point based on the attention output so that the historical text which contains more relevant content affects the current opinion more intensively than those with less relevant content. The following two equations describe the shaped output with the attention mechanism and the Hawkes process:

$$\begin{aligned} v'(t) &= v + \varepsilon \sum_{i:\Delta t_i \geq 0} \lambda'_i e^{-\beta \Delta t_i} && (6.2) \\ &= \sum_{i:\Delta t_i \geq 0} (\lambda_i + \varepsilon \lambda'_i e^{-\beta \Delta t_i}) && (6.3) \end{aligned}$$

where

$$\lambda'_i = \begin{cases} \lambda_i & \text{if } \lambda_i > 0 \\ 0 & \text{else} \end{cases}$$

In Equation 6.2, the first element v is the background intensity which acts as a base factor and describes the content in the text throughout the time. ε represents a decay impact factor and balances the importance of adding the Hawkes process in the output; $\varepsilon = 0$ would indicate that the information decay does not play any part in the decision making. Theoretically, there should be no upper bound for the value of ε, however it is illogical to take a value that is much greater than 1 ($\varepsilon \ngg 1$). $\Delta t_i = t - t_i$ is the time difference between the current time t and the time t_i. β is the decay rate for the time difference, and the value of β varies according to the time unit chosen in Δt_i and the task. Conventionally, the parameters in the Hawkes process (here, we have ε and β) are constants whose values must be chosen priorly. Here, α in Equation 3.1 is replaced by λ_i' so that the effect of an arrival is not constant anymore. λ_i' is a rectifier which takes $max(0, \lambda_i)$. With the rectifier, the effect remains non-negative and only relevant events (targets) are considered. λ_i is calculated according to Equation 5.9. Given Equation 5.10, Equation 6.3 is deduced so that at each time point in the past, the attention output is boosted by the process factor when it is a positive value. The final output $v'(t)$ of the Hawkes-attention layer is the sum of the modified attention outputs over time for the current post created at time t. An illustration of the Hawkes-attention layer is shown in Figure 6.5, where the lower level corresponds to the outputs from the last recurrent layer as in Figure 5.3, a_i is derived from h_i with the post information, and μ_i applies the excitation function based on the time gap at time step i.

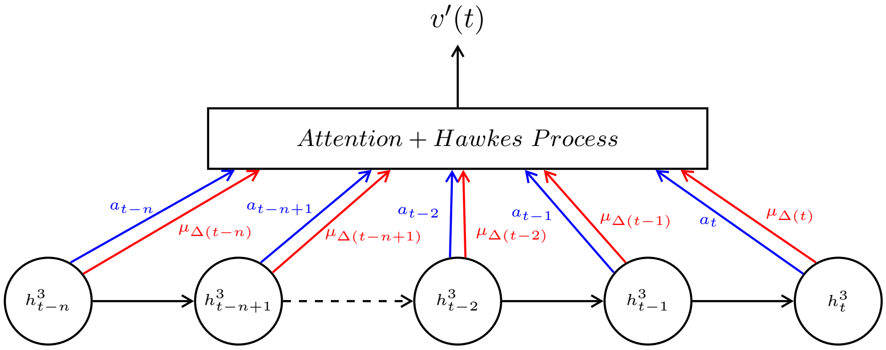


Figure 6.5.: The attention mechanism with the Hawkes process.

The advantages of using exponential kernels in Hawkes process descend from a Markov property as explained in Bacry et al. [2015], which motivates the use of this excitation function in the model. The property can be extended to the case where the value of β is non-constant, which is useful in assigning different decay rates for different users or for different levels of intensity (attention outputs).

Although the non-parametric method introduced in Section 3.3.2 by Cao et al. [2017] can be applied in the proposed model, the selection of the number of intervals undermines the flexibility of the process which is a crucial point in this research because of the irregularity of the events. The parameters ε and β, which balance the importance and the rate of the decay, can be chosen beforehand. The exact values for the parameters can be found experimentally by grid search given an empirical range, e.g., $(0, 1]$ and $(0, 0.1]$. Other ways to discover the values for the parameters will be discussed in detail in the next section.

6.2.2. Trained Hawkes Processes

Since the Hawkes process is controlled by the parameters introduced in the system, more flexible ways are provided in order to define the values by using a trainable setting instead of the empirical one. Following the Hawkes-embedded model structure, two alternatives are introduced: One alternative is to set the parameters targeting the whole population (universal Hawkes process) and the other is targeting individual users (user-specific Hawkes process).

Hawkes-embedded Model Structure

To employ the concept described in Section 6.2.1, the sentiment model is taken as described in Section 6.1.5. The structure of the Hawkes-embedded sentiment model follows a conventional embedding – recurrent – attention process as shown in Figure 6.6.

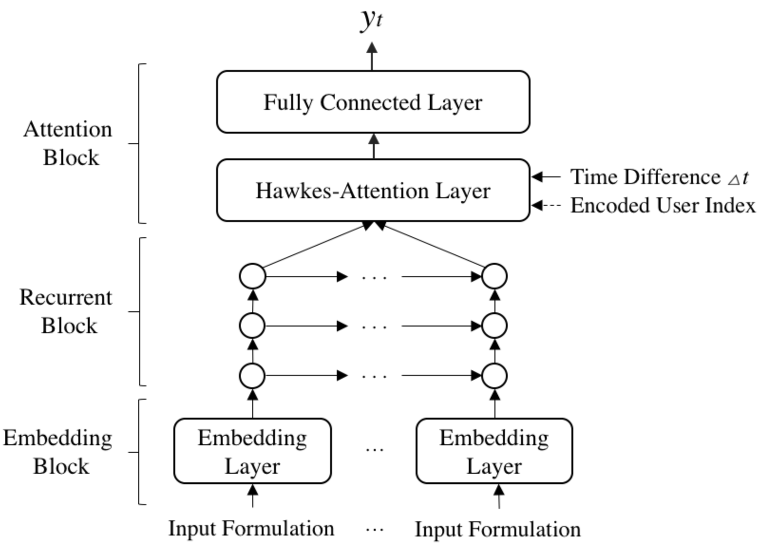


Figure 6.6.: The structure of the Hawkes-embedded sentiment model.

First, a formulation method is applied in order to represent the current and a number of earlier posts of a user. The embedding layer takes the formulated inputs and produces a vector for each time step at the recurrent layer. After that, the outputs of the recurrent layer together with the auxiliary time differences are fed to the attention block. Encoded user index (indicated with the dash line) is used in the Hawkes-attention layer when different settings of Hawkes process are considered for different users. A fully connected layer is added after applying the Hawkes process in order to regularize the output of the Hawkes-attention layer for the network to train (as used in Cao et al. [2017]). It takes the exited (decayed) information representation v' (Equation 6.2) as input and outputs the final prediction y_t which is the sentiment label of the target text.

The *Input Formulation* applied in the model is determined by the choice of representation method. In this work, five input formulations are selected which signify typical choices in related tasks:

Formulation 1: Atomic Representation (AR)

While atomic representation is used, the input formulation follows the method in Section 5.2.1.

Formulation 2: Pre-trained Word Embeddings (WE)

While pre-trained word vectors are used, the vectors are aggregated dimension-wise to produce one vector for each post. The user index is encoded by itself and then combined with the representation of the post at each time point of an input sequence.

Formulation 3: Concepts and Words (CW)

Since the pre-trained word vectors do not consider the contexts of the target words, the representations of words and concepts are combined in this formulation. The word embeddings are taken the same as the one in the **WE** formulation. The concepts appearing in the text are encoded together with the associated user index so that the relation between the user and the use of concepts can be learned. Afterwards, the two types of representation of the same post are concatenated to generate an input sequence for the recurrent layer.

Formulation 4: Deep Contextualization (DC)

Peters et al. [2018] proposed a deep contextualized representation (ELMo) that takes a finer granular level (characters) to generate embeddings for the text by leveraging a deep bidirectional language model. Prominent results are shown across a number of linguistic tasks using this representation. Here, ELMo is applied for the text at each time point, and then the representation of the texts and the encoded user index are combined at each time point.

Formulation 5: Combined Granular Levels (Combi)

This formulation combines three granular levels, namely character-level (**DC**), word-level and concept-level (**CW**). The representations are embedded separately and are concatenated afterwards as mentioned in Peters et al. [2018].

These five formulations are utilized for the purpose of evaluating the performance of the two alternatives of the Hawkes process. To create the input sequence according to the input formulation, Equation 5.6 is adopted.

Universal Hawkes Process

Instead of using a grid search over an empirical range to select the values for ε and β in Equation 6.3, the training process of neural networks can be leveraged. That is to indicate that ε and β can be empirically initialized and jointly learned with other weights during the training phase. Thus the weights needed to be trained at the attention layer become $[W_t, b_t, w_s, \varepsilon, \beta]$ which was formerly $[W_t, b_t, w_s]$ in Equation 5.7 and 5.8. Formally, the original output of the attention layer is

$$Attention_t = f_{att}(g(X_t), W_t, b_t, w_s) \tag{6.4}$$

where f_{att} corresponds to the attention mechanism as in Section 5.2.3 with trainable parameters W_t, b_t and w_s, g is the non-linear function with respect to the three-layered recurrent network, and X_t is the input sequence at t that follows an input formulation as in Equation 5.6. Hence further, the output of the Hawkes-attention layer is

$$Output_t = f_{haw}(Attention_t, \underbrace{ReLU(Attention_t), \Delta t, \varepsilon, \beta}_{\text{Remainder}}) \tag{6.5}$$

where f_{haw} corresponds to the function of the revised Hawkes process. The first component inside the brackets takes the original attention output and passes the value through the process without any modifications. The attention mechanism is partial differentiable and can be trained during the backpropagation (an algorithm that optimizes the parameters in neural networks through gradient descent [Schmidhuber, 2015]) as used in many earlier tasks. The second component can be merged with the first one because the rectified linear unit (ReLU) is also a function of $Attention_t$. However, since the attention output is modified in the model, further discussion is needed to clarify the differential property of the revised Hawkes process. Given the linear additivity, the variables in the *Reminder* – the additional term in Equation 6.3 – can be considered in a standalone calculation. When computing the gradient of the ReLU function, a subderivative in the interval [0,1] can be taken since the function is nondifferentiable at 0. It is not crucial in practice because the variable of the function rarely arrives at 0; nonetheless, the subderivative is mostly chosen to be 0 for simplicity. Although Δt is a variable in the Hawkes process, it is not a trainable parameter of the network but a direct input. Therefore, it is considered as a constant and acts as a coefficient while calculating the derivatives with respect to the trainable variables. Since the reminder is given as

$$R(\lambda'_i, \Delta t, \varepsilon, \beta) = \sum_i \varepsilon \lambda'_i e^{-\beta \Delta t_i} \tag{6.6}$$

it is obvious that R is differentiable with respect to ε and β. Because of the added remainder, there can be cases where some values exceed the original boundary of the attention model. Note that this only affects the upper bound after the ReLU function is applied. To reshape the values, additional activation function is used in the fully connected layer afterwards.

With ε and β trained when optimizing the network, no additional prior knowledge is required to estimate the behavior of the information decay. However, even with the same training samples, there is no guarantee that the trained values will always be the same because of the built-in randomness of the executive environment in reality. The same situation holds for the other trained parameters in the model as well, which is a common issue pertaining to neural networks. Nevertheless, the performance of the system is generally not undermined by the issue — the optimized network can be resilient to minor variations. In the end, the universal Hawkes process is defined by the two constants ε and β after the training phase.

User-specific Hawkes Process

Although the user information has been included in the input, the decay behavior modeled by the process is considered universal for all the involved individuals. However, Janis and Field [1956] have stated that

> *'... some individuals tend to be indiscriminately influenced by the many persuasive communications to which every modern urban community is continually exposed, while other individuals tend to be generally unresponsive to such communications.'*

The different levels of 'persuasibility' across the population lead to various frequencies of people changing their opinions. It suggests that some people' opinions decay faster than the others. Therefore, different processes are desired to model the various frequencies. While a message appears, the Hawkes process is stimulated by the incident and the decay effect of the process follows the behavior of the user who posted the message. Eventually, the individual's perception on related incidents in the near future is affected accordingly.

The user-specific Hawkes process can be moderated by the defined parameters. In order to train the process, the values of ε and β are computed for each user by applying learned transformation vectors on the encoded user index. In this way, different behaviors concerning the information decay can be analyzed. That is, ε and β are calculated as

$$\varepsilon = a_{\varepsilon}^{\top} E(u) \tag{6.7}$$

$$\beta = a_{\beta}^{\top} E(u) \tag{6.8}$$

where E is the user-index encoder for which a simple embedding layer or a fully connected layer can be applied. The transformation vectors a_{ε} and a_{β} are learned during the training process, and other settings remain the same with the universal Hawkes process. Thus, the output of the Hawkes-attention layer with user-specific processes is

$$Output_t = f_{haw}(Attention_t, \underbrace{ReLU(Attention_t), \Delta t, E(u), a_{\varepsilon}, a_{\beta}}_{\text{Remainder}}) \tag{6.9}$$

Here, I elucidate the differentiability of the modified function. The derivatives with respect to the first two components in the remainder are calculated the same as in Equation 6.5, whereas the others do not change the differential property as long as none of the elements in $E(u)$ is 0. As explained, it is rare that the previous layer outputs a value of 0. In the end, the vectors a_ε and a_β can capture individual factors on the importance and the rate of the information decay. Although the same issue with the randomness persists, the individual differences can be easily monitored for the analysis.

Part III

Model Evaluation

Based on the background and the theoretical establishment in Part I and II, different aspects in individuality pertaining to sentiment analysis are evaluated in Part III. Specifically, as introduced in the previous chapters, data from social platforms is used for the evaluation where Twitter data is taken as an example. In the evaluation, the assumptions mentioned in Section 4.1.1 are examined with different data resources, and experiments are conducted from different angles in order to investigate the different aspects.

Details concerning the experimented data and models are given in Chapter 7. Comprehensive model comparisons for the studied aspects are provided in Chapter 8. Given the focus of the personalization, the key factors of the task are discussed in Chapter 9. The reported results shown in the thesis were previously published in Guo et al. [2018a,b, 2019a,b,c].

Chapter 7

Experimental Setup

In this chapter, the settings of the experiments designed for the underlined research with respect to the data resources, external tools, and technical model setups are introduced. Different settings are used in the experiments according to the aim of the specific task. Particularly, an existing knowledge base which offers concept-level representation is explored in the preliminary study, and external resources associated with different granular levels are utilized when evaluating the refined model.

7.1. Data Resources

Targeting data from social platforms, three publicly available datasets are taken and the Twitter API[16] is used to extract information needed for the task. As discussed in Section 2.4, the datasets are categorized into manually annotated data and automatically annotated data, where the dataset in the latter category can be seen as labeled from the expressers' perspectives.

7.1.1. Data Requirements

Ideally, to examine the proposed sentiment models, the experimented data are aimed to comply the requirements that:

1. there is a sufficient number of frequent users,
2. the text is domain-independent,

[16] see https://developer.twitter.com

3. the desired textual and contextual information is present,
4. the corpus is annotated from the writers' perspectives.

The performance of neural-based models largely depends on the size of the data. Given the nature of this research, frequent users are the main targets of the defined task. Hence, automatically annotated data can be a better choice for its freedom in labeling and the easiness in acquiring frequent users. Moreover, the mismatch regarding the standpoints is eliminated despite the possibility of having irregular emoticon usages at the user-end. Manually labeled data are used for the preliminary study so that an universal comparison can be made to evaluate the effectiveness of the assumptions, while automatically labeled data are used for the advanced model (with and without information decay) so that a more comprehensive evaluation on the personalization can be provided.

7.1.2. Manually Annotated Data

Sanders Twitter Sentiment Corpus[17] and the development set of SemEval-2017 Task 4-C Corpus[18] are manually labeled datasets and are used for the evaluation in the preliminary study. The SemEval corpus is comparatively more objective than the Sanders corpus, because the annotation of SemEval is done by crowd-sourcing while for Sanders, the classification is done by one person.

The statistics of the datasets used for the models is shown in Table 7.1. For the SemEval corpus, germane labels are merged into three classes (positive, negative, and neutral) in order to combine the corpus with the Sanders corpus. These two corpora are combined since they are both hand-labeled and the combination enriches the data with more frequent users (only 51 users have tweeted more than 5 times). Furthermore, the independency between a corpus and the topic-concept relation can be verified: The SemEval corpus contains 100 topics, while the Sanders corpus contains only four topics that are 'apple', 'google', 'microsoft', and 'twitter'. As shown in Figure 4.4, 'moto g' is located very close to these four topics because they are more correlated than others, although it is from the other corpus.

For training, a subset of the combined dataset is used while the rest (30%) is reserved for testing. The training set is further separated for development and validation (30%). The test set provided by SemEval is not adopted, because it contains only new topics which are not suitable for examining topic dependencies learned by the network. The preliminary model is able to deal with tweets with unseen topics, but with such a test set, the relations between the unseen topics and learned topics will be lost and the system becomes topic-independent.

[17] http://www.sananalytics.com/lab/twitter-sentiment/
[18] http://alt.qcri.org/semeval2017/task4/

Dataset		Sanders	SemEval	Sentiment140
Labeling		Manual	Manual	Automatic
Polarity	Positive	424	6,758	79,009
	Negative	474	1,858	42,991
	Neutral	2,008	8,330	—
	Total	2,906	16,946	122,000
#. Topics or Entities		4	100	311 (min. 20 times) or 761 (min. 10 times)
User Frequency		51 users with min. 6 tweets 971 users with min. 2 tweets		2,369 users with min. 20 tweets

Table 7.1.: Statistics of the datasets used for the preliminary and the advanced model.

7.1.3. Automatically Annotated Data

Sentiment140[19] is automatically labeled with two classes (positive and negative) and is used in the advanced model. Originally, Sentiment140 contains 1,600,000 training tweets, however tweets published by users who have tweeted at least 20 times before a pre-defined date are extracted so that only frequent users are considered in this model. The extracted subset contains 122,000 tweets in total. In the dataset, 22.4% of the tweets are generated by users with frequency not less than 50, which results in 27,304 instances; 6.1% are generated by users with frequency not less than 100, which results in 7,449 instances. As explained in Section 4.1.2, entities are used instead of topics which results in 15,305 entities extracted from the text. Then, the extracted entities are filtered according to the number of occurrences in the corpus to avoid errors during the extraction and to exclude rare terms. Details are shown in Table 7.1.

Furthermore, the dataset is split into a training set, a validation set and a test set for training and evaluation. As before, the original test set from the corpora is not suitable for the experiments. The reason is that the provided test set contains only unseen users, which makes it impossible to verify the user preferences learned by the advanced model. Additional information on the data separation can be found in Table A.1.

7.2. External Resources and Tools

In this section, I provide details concerning the three external resources that are used to help construct the input representation from different granular levels and additional tools that are exploited to help accelerate the implementation of the neural networks. The resources and tools are chosen in a task-oriented way given the availability at

[19] http://help.sentiment140.com/for-students/

the time of development; similar results and observations may be achievable using alternatives.

7.2.1. SenticNet

As mentioned in Section 4.1.2, concepts from SenticNet are taken for the atomic representation. In the experiments, the SenticNet is used as a knowledge base, which provides a set of semantics, sentics, and polarity developed for concept-level sentiment analysis. As described by SenticNet[20]:

> *'... semantics are concepts that are most semantically-related to the input concept (i.e., the five concepts that share more semantic features with the input concept), sentics are emotion categorization values expressed in terms of four affective dimensions, and polarity is floating number between* -1 *and* $+1$*.'*

During their developing phases, different amounts of concepts are given among which SenticNet 4 with 50,000 entries and SenticNet 5 with 100,000 entries are integrated in the proposed models. However, no significant difference can be observed for the sparsity of the alignments between the knowledge base and the corpus. Since the research focuses on the analysis of sentiment, the emotion-related sentics are omitted in the experiments, but the polarity derived from the sentics is considered.

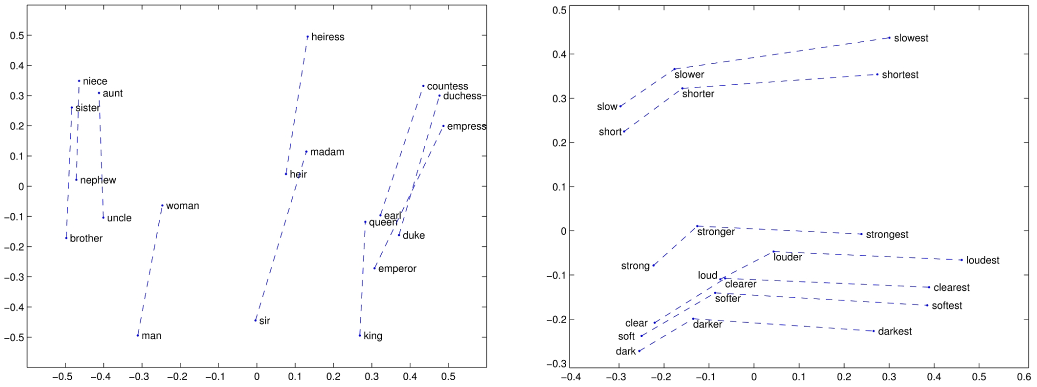
7.2.2. GloVe Word Embeddings

GloVe (Global Vectors for Word Representation [Pennington et al., 2014]) is a prominent approach for pre-trained word vectors. The vectors are trained based on the word co-occurrences, and they contain additional semantic relations that reveal certain linear, inner-words substructures in the corresponding vector space. Examples of such relations are illustrated in Figure 7.1, where the linear dependencies such as in (a), $woman - man = queen - king$, and in (b), $strongest - stronger = loudest - louder$ can be seen. Personalized sentiment analysis benefits from the learned relations through the improved language understanding and also in other ways, for instance, with example (a), the relation between the entities can help acquire information on sentiment transferability of a user from one concept to a similar concept, and with (b), switching from comparative to superlative can give direct hints on the amount of increase to be added regarding the intensity of the sentiment. The interpretations for the use of the learned relations are conceptual and are difficult to verify explicitly.

A major drawback of the GloVe embeddings is the mishandling of the polysemous words. As discussed in Arora et al. [2018], the polysemy is not directly discernible from the embeddings, although the message resides in the linear superposition can

[20] https://sentic.net/

be recovered by the approach proposed by the authors. In this experiment, despite the inefficiency in representing polysemy, the embeddings are adopted and the other co-occurred terms in a post are taken into account by the followed recurrent neural network to compensate.



(a) man-woman

(b) comparative-superlative

Figure 7.1.: Examples of the linear substructures obtained in the vector representation by GloVe[21].

Moreover, the information conveyed in the embeddings is corpus-dependent, thus specifically, the vectors that were obtained from a Twitter corpus with 100 dimensions are taken. In Pennington et al. [2014], the authors also claimed that the GloVe word vectors outperform Word2Vec consistently under the same settings. In the refined model, GloVe is applied in the input formulations **WE**, **CW**, and **Combi** representing word-level granularity.

7.2.3. ELMo Representation

ELMo representation [Peters et al., 2018] is a character-based representation which belongs to the category of the finer-granular level. ELMo follows the structure of a deep bidirectional language model, which in particular, combines three vectors to produce the representation as in Figure 7.2. The three vectors are the raw word vectors generated from a convolutional neural network on characters, the intermediate word vectors generated from the first layer of the bidirectional language model with forward and backward pass (realized by LSTM cells), and the word vectors formed from the second layer of the bidirectional language model with the same design taking the previous intermediate word vectors as inputs. The final representation is the weighted sum of these three vectors. As a result, the representation relates to all the context words in the sentence when producing the vector for the word, i.e., a word may have

[21] https://nlp.stanford.edu/projects/glove/

different embeddings when appears in different sentences. Such representation provides a solution for handling polysemy which was previously overlooked by the traditional word embeddings as discussed in the last section.

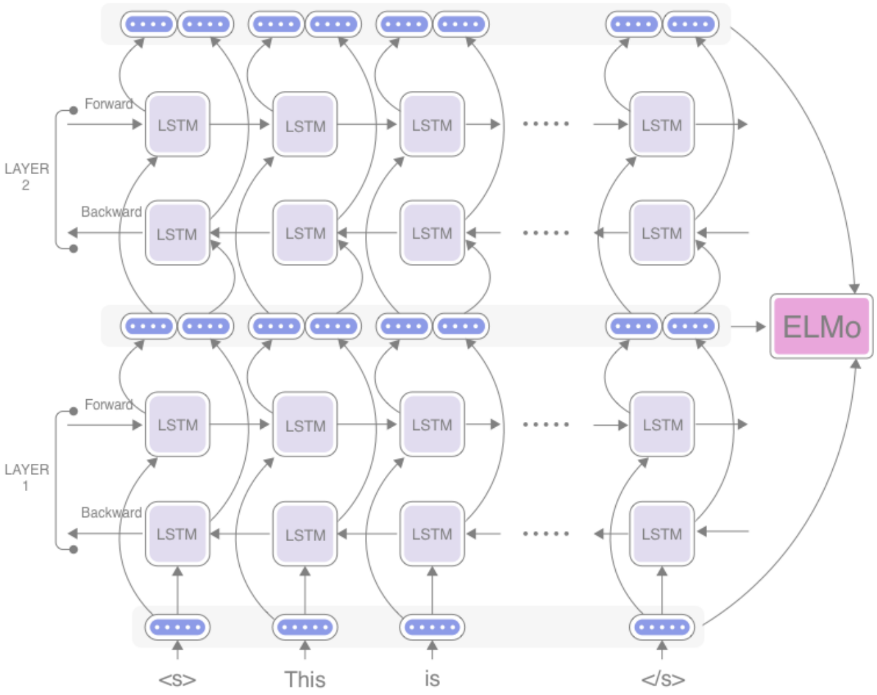


Figure 7.2.: ELMo representation combines internal word vectors of a multi-layered bidirectional language model[22].

In the refined model, the ELMo representation used in the **DC** and **Combi** input formulations is supported by TensorFlow Hub[23], which is later re-trained with other weights in the model. Specifically, a fixed mean-pooling of all contextualized word representations in a post is used to have a consistent representation for the posts, and other components are added afterwards to compose the input formulation.

7.2.4. Keras with TensorFlow Back-end

Keras[24] is a user-friendly programming interface that is compatible with three backend implementations among which TensorFlow[25] is used in the experiments. It enables sequential and functional commands that target at different needs. The Keras library

[22] http://www.realworldnlpbook.com/blog/improving-sentiment-analyzer-using-elmo.html
[23] https://tfhub.dev/google/elmo/2
[24] https://keras.io/
[25] https://www.tensorflow.org/

supports a list of conventional neural network layers focusing on convolutional and recurrent networks, with the possibility of creating customized layer to be combined with other layers if needed. There are other application platforms implementable for the experiments, however the theoretical aspects of the targeted research are central and not the evaluation of the possible platforms for the implementation. Keras is thus chosen at the time of development for the accessibility and the simplicity in realization.

7.3. Model Settings

The hyperparameters used in the models are first chosen empirically by referring to the values reported in existing works and scaling up or down according to the associated data size. After that, they are readjusted during the training process by monitoring the model performance on the validation data. Here, I report the final setting used to produce the results that will be indicated in the next two chapters.

7.3.1. Preliminary Model Setup

In the preliminary study, the separate shallow neural networks for generating embeddings contain 32 nodes at the hidden layer for topic representation and 128 nodes for concept representation. From 50,000 concepts listed in SenticNet 4, 10,045 occur in the combined dataset of Sanders and SemEval. For the recurrent network, the first two layers contain 64 cells each while the third one contains 32 cells. Dropout (= 0.4) is applied on inputs and weights during the training phase to prevent overfitting [Srivastava et al., 2014]. Also, Adam optimizer (learning rate = 0.001, Kingma and Ba [2015]) is adopted while compiling the model with categorical cross-entropy as the optimization score function (loss function). The model integrates at most 20 past tweets.

7.3.2. Advanced Model Setup

In the refined model, the concepts used in the **AR**, **CW**, and **Combi** formulations are from SenticNet 5. In **WE** and **CW**, 100 dimensional Twitter word vectors are taken from GloVe. The inputs with **AR**, **WE** and **DC** are encoded into different lengths based on the number of elements in each formulation, however they are suppressed at the embedding layer that generates a vector of length 100 at each time step in order to make fair comparisons. In **CW** and **Combi**, the input vectors fed to the recurrent layer are longer (164 and 264 respectively) because of the concatenation of representations. The dimension of user embeddings is set to 32. Each of the three recurrent layers at the recurrent block contains 100 units, and the number of time steps T is set to 20. For the selection procedure, Manhattan distance is used as the ground measurement for calculating topic similarities and the similarity threshold δ is set empirically at 0.8. At the attention block, the time unit is hour, and the values of ϵ and β are initialized at 0.01 and 0.001 respectively when using the universal Hawkes process; the initial values for the vectors a_ϵ and a_β when using user-specific processes are also vectors of 0.01 and 0.001 respectively, and the length of the vectors has to be the same with the dimension of user embeddings, which is 32. The dimension of the fully connected layer applied before the output is set to the same as the number of units in the recurrent layer. While using the refined model, the overall accuracy of the model as well as the F_1 scores for the positive and negative classes are reported.

Chapter 8

Model Comparisons

In this chapter, I compare the proposed models with manually and automatically labeled data as described in Section 7.1.2 and 7.1.3. Specifically, I evaluate the performance of the model using the input selection algorithm as introduced in Section 5.3, the different ways to associate time information as in Chapter 6.1, and the empirical and trained Hawkes processes as in Chapter 6.2. An overall evaluation of the proposed models is given in Section 12.

8.1. From the Preliminary Study to the Advanced Model

The manually labeled data are used to evaluate the performance of the preliminary model regarding the effectiveness of the assumptions introduced in Section 4.1.1. Five baseline models are chosen for the evaluation including both traditional classifiers and neural networks. Based on the findings of the preliminary study, the atomic input formulation has been revised as in Section 5.2.1 and automatically labeled data are used to evaluate the performance of the advanced model. An initial test for the advanced model is discussed in this section.

8.1.1. The Preliminary Study: Baselines

The performance of the preliminary model is compared with five baseline models. The first one utilizes the polarity values as mentioned in Section 4.1.2. The values are combined for each post, after which the result together with the number of concepts that occurred in the post are fed to a shallow fully connected network for training. This is done in order to set up a baseline that demonstrates the performance when no implicit connections of any aspect are concerned.

The neural network-based approach is compared with the traditional classifier, Support Vector Machine (SVM), which is a prominent method for sentiment-related tasks. Two SVM models are trained with different inputs using scikit-learn [Pedregosa et al., 2011]. One is a generalized model (G-SVM) trained with the presenting concepts and the associated topic (no user information attached), and the other is a personalized model (P-SVM) trained with the input of the generalized model together with the user index and public opinions. While using the radial basis function (RBF) kernel, the value for the parameters $C = 0.01$ and $\gamma = 1/N_{features}$ are set by 10-fold cross-validation on the training data.

An experiment with convolutional neural network (CNN) is also conducted. CNN is a widely used method in image processing [Krizhevsky et al., 2012; Lawrence et al., 1997], and has been found to provide good performance for NLP tasks as well. Kim [2014] uses the convolutional neural network over static and non-static representations for several sentence classification tasks. They have shown that a simple convolutional neural network is able to offer competitive results compared to other existing approaches. The structure used as a baseline in the experiment is similar to Kim [2014] with the following modifications. First, each post is represented by the concatenation of its N constituent concepts, and then a convolutional network with two convolutional layers is applied on the concept embeddings as explained in Section 4.1.4. This structure highlights the inner relationship between concepts, especially on their adjacent appearances in a message.

Finally, a generalized recurrent neural network (G-RNN) is used to compare the performance considering the dependence between the past and the current tweets when no user related information is used. The network proposed for the preliminary model is used without attaching a user index in the input sequence, and $x_{t*} = [E_{concept}\ E_{topic}]_{t*}$ is set at the input nodes. With such a network, $E_{concept}$ and E_{topic} represent the concepts and topics from a general view, thus $P_{concept}$ and P_{topic} are no longer needed. The tweets are then ordered by the creation time. With user information removed, the network mainly learns by comparing the presenting concepts and the associated topic from different time points.

8.1.2. Preliminary Model Comparison

Table 8.1 shows the performance of the preliminary model compared with the five baselines described in the preceding section. Accuracy is used as the primary evaluation metric, and macro-averaged recall is used to demonstrate the balance of the prediction over classes. Note that the performance of the preliminary model is not compared with the reported results of SemEval, because different test data are used for the evaluation as explained in Section 7.1.2.

The granular level for all the models are concept-based to enable a consistent intra-

Model	Avg. Recall	Accuracy
Sentic	44.08	37.69
G-SVM	58.47	61.13
P-SVM	58.62	61.47
CNN	53.60	54.81
G-RNN	65.87	63.82
Preliminary Model	**68.59**	**65.69**

Table 8.1.: Comparison of the performance between the preliminary model and the chosen baselines.

post representation. The Sentic model performs the worst, which is expected since it is a simple network for combining Sentic values. In the preliminary model, the Sentic values act as public opinions that are not representative on their own. They reflect a general understanding of the concepts which is neither user related nor semantically dependent.

The SVM models provide reasonable results for the given dataset. The performance of the G-SVM model is slightly below that of the G-RNN model, where the difference is predominantly caused by the trait of recurrent networks being able to consider dependencies through time. The fact that there is no significant improvement after adding user information in the P-SVM model shows us that the SVM models in their current forms are not suitable candidates for the task of analyzing individuality in sentiment.

In the work of Kim [2014], the convolutional neural network performs mapping by a sliding window over adjacent words which implies that the order of words appeared in a sentence plays a significant role, i.e., contiguous words have greater dependence. However, for concepts such an interaction is not obvious. The concept itself includes implicit connections between words, therefore this network only studies the co-occurrence of the concepts on the intra-post level. As a prerequisite, the concepts must be transformed into numerical representations, for which one-hot vectors or embeddings can be applied. To abstain from the sparsity issue pertinent to concept representation, embeddings are adopted in the experiment. However, the process of embedding can also be seen as a compression that may cause information loss. Consequently, the CNN model results in a worse performance than the SVM models.

The G-RNN model works better than the convolutional network because the connections to the past as well as between topics are studied. This model intends to capture the trends in public opinions — the information of public preference towards a topic at different time is memorized and analyzed. This baseline shows the effect of **Assumption I** and **III** in the preliminary model. By adding the user index in the model, the performance is further improved (t-test with $p < 0.05$), which indicates that considering the diversity in lexical choices and an individual's relation with the public positively influence the prediction.

8.1.3. The Advanced Model with Automatically Annotated Data: An Initial Test

Given the observations from the experiments with the preliminary model, it can be believed that the role of individuality in expressing sentiment is significant for a deep understanding of user-generated text. Further, I concentrate on frequent users to discover more traits on the influence of individuality in the study. Given the data requirements in Section 7.1.1, automatically labeled data provide requested flexibility, thus is used for the evaluation of the advanced model.

In Table 8.2, as an initial test for the use of the reformed inputs, the performance of the following models is compared:

1. A model with the output layer applied directly after the recurrent network (named *RNN model*);
2. The advanced model without information decay as in Figure 6.1 (named *Attention model*, as the output layer is applied directly after the attention mechanism);
3. The advanced model with empirical Hawkes process as in Section 6.2.1 (named *eHawkes model*, as the output layer is applied after the attention mechanism is shaped by the empirical Hawkes process).

As the same as before, accuracy is used as the primary evaluation metric; however, F1-score is calculated for each class to demonstrate the balance of the prediction since only two classes are concerned. Moreover, the same input sequences are given to all the models, thus the effect of applying the attention mechanism and the Hawkes process can be evaluated.

Model	Pos. F1	Neg. F1	Accuracy
RNN Model	73.44	72.23	72.87
Attention Model	74.64	75.16	74.91
eHawkes Model	**75.34**	**76.52**	**75.96**

Table 8.2.: Comparison of the performance between the RNN model and two variants of the advanced model.

Improvements can be seen after adding the attention layer and after shaping the attention outputs with Hawkes process. The Attention model compensates the loss of focus of the RNN model with distant nodes, and the eHawkes model tightens or loosens the relations of the nodes according to the gaps between them. To implement the eHawkes model, the values of ε and β in Equation 6.3 must be set beforehand.

$\varepsilon = 0.7$ and $\beta = 0.01$ are taken which give the best performance in the experiment, and the corresponding results are shown in Table 8.2.

8.2. Effect of Input Selection

As introduced in Section 5.3.1, Euclidean distance (**ED**), Manhattan distance (**MD**), cosine similarity (**CS**), earth mover's distance (**EMD**), and word mover's distance (**WMD**) are compared when applying the input selection algorithm (Algorithm 1) in the sentiment model. The topics are represented by vectors of dimension 100 (the size of the hidden layer) while the same size is chosen for the GloVe vectors. The thresholds of the distance measures are chosen individually by comparing their performance when applied in the system. Other settings remain the same as before without considering time gaps between the posts. **Table 8.3** shows the performance of the system, where **EMDc** denotes the model in which EMD is calculated between the two centroids of the topics, and **EMDt** denotes the model in which EMD is calculated between the two sets of topics.

Model	$T = 10$			$T = 20$		
	Pos. F1	Neg. F1	Accuracy	Pos. F1	Neg. F1	Accuracy
NoSelect	73.62	74.39	74.01	74.63	75.66	75.17
ED	74.19	**75.68**	**74.96**	74.73	76.19	75.48
MD	**74.20**	74.71	74.47	74.96	**76.67**	**75.85**
CS	73.69	74.80	74.26	74.72	76.24	75.50
EMDc	73.79	75.66	74.76	74.80	76.19	75.52
EMDt	73.91	75.25	74.61	**75.08**	76.32	75.72
WMD	73.30	74.44	73.89	74.54	75.72	75.15

Table 8.3.: Performance of the system before (NoSelect model) and after implementing the selection technique with different distance measures for time steps 10 and 20.

It can be seen that Euclidean distance performs the best when T is 10, and Manhattan distance outperforms others when T is 20. It is unexpected that WMD has a performance that is not better than the NoSelect model where no selection is used. The reason can be that WMD takes external vectors while the topic embeddings for other measures are learned by considering the surrounding concepts which capture the affective information.

Aside from the WMD measure, there are no significant differences between other distance measures; however there are significant improvements between the NoSelect model and the method that provides the best results. This shows that considering the longer history by topic similarity has a positive effect on the performance of the system. It is also significant to increase the number of time steps from 10 to 20 so that more information can be related to by the RNN. Although the procedure is executed fewer times with $T = 20$, it is still effective implementing the selection method compared to the NoSelect model. Therefore, it can be concluded that the selection procedure generally improves the performance, but the choice of distance measure used in the algorithm can vary depending on the value of T. Models with smaller values of T can be more sensitive to the selection given the number of executions, while at the same time, greater improvements can be observed.

8.3. Effect of Time Gaps

In this section, the different approaches for associating time information in the personalized sentiment model are evaluated. As introduced in Chapter 6.1, the three following cases are considered:

Case I: Integrating the time information in the input as in Figure 6.2.

Case II: Integrating the time information in the attention mechanism as in Figure 6.3.

Case III: Integrating the time information with Hawkes Process as in Figure 6.4 where the universal process is adopted.

The three cases are compared with the RNN model where no attention mechanism is applied before the output layer and no (exact) time information is included in the system, and with the Basic model where the attention mechanism is applied but the time information is excluded.

Model	Pos. F1	Neg. F1	Accuracy
RNN Model	74.67	74.01	74.35
Basic Model	75.00	75.49	75.26
Case I	74.94	76.25	75.62
Case II	75.10	76.55	75.86
Case III	**75.53**	**76.70**	**76.13**

Table 8.4.: Model evaluation while using time information in different cases.

Table 8.4 shows the performance of the five models. Note that SenticNet 5 is used here in contrast to the initial test in Section 8.1.3 where SenticNet 4 is utilized. With the amount of pre-defined concepts doubled, the RNN model and the Attention model (regarded as the Basic model in this experiment) have shown increases in the performance (Table 8.2).

Comparing the F1-scores for the positive and negative classes as well as the accuracy, the RNNs model performs the worst which is followed by the Basic model. These two models do not associate time information in the model, thereby revealing the significance of including such information in the analysis. The three cases provide competitive results while Case III with Hawkes process being slightly better than the others. In fact, the improvement of Case III over the models without time information passes the t-test (with p<0.05). Case II and III add the time at the attention layer thus affect the output more directly than Case I. In Case II, the attention value is modified by the time using a multiplication, while in Case III the employed decay (or excitation) process results in better performance. However, the time encoder in Case I and II is also capable of forming a decay function, and more sophisticated structures can be explored to enhance the results. The differences among the three cases are not significant in all categories (e.g., the difference of Pos. F1 between Case I and III is significant), however the better explicitness with the Hawkes process in Case III (Section 6.1.5) motivates us for future exploration.

8.4. Universal Hawkes Process vs. User-specific Hawkes Process

Table 8.5 shows the performance of the models with the universal Hawkes process and the user-specific Hawkes process respectively when using different input formulations (introduced in Section 6.2.2). Note that the input selection algorithm is employed in this experiment. The best result is given by **Combi** formulation with user-specific Hawkes process.

Input	Universal Hawkes Process			User-specific Hawkes Process		
Formulation	Pos. F1	Neg. F1	Accuracy	Pos. F1	Neg. F1	Accuracy
AR	75.12	76.87	76.04	74.94	77.48	76.28
WE	76.06	77.06	76.58	76.61	78.41	77.55
CW	76.43	77.71	77.10	76.90	79.10	78.06
DC	76.60	79.71	78.27	77.10	80.36	78.86
Combi	77.78	80.67	79.34	78.06	82.25	80.38

Table 8.5.: Performance of the sentiment models when applying the two Hawkes processes with different input formulations.

Across different input formulations, improvements can be seen comparing the models using the universal Hawkes process and the user-specific Hawkes process. Although the increase when applying the **AR** formulation is not significant, the improvement of other formulations are significant (t test with $p < 0.05$). The lack of improvements with the **AR** formulation when learning user-specific behaviors can be caused by the sparser representation compared to the other formulations. A matching from each post to a list of concepts and negations is performed which omits information that is not present in the given list. The list of concepts provided by SenticNet 5 is more restricted than the word vectors by GloVe, and is far less flexible than the character-based representation. In addition, due to the non-canonical nature of the user-generated text, the preprocessing of the posts plays a significant role in the **AR** formulation, which also affects the performance substantially.

Improvements can also be observed when using finer granular levels which are more sensitive and representative towards user variations. Note that using the character-based **DC** formulation alone offers better performance than using the combination of word - and concept-level representations, however the ELMo representation has a more complex structure, a higher dimensional output, and it takes longer time in order to re-train the weights in the network. More specifically, as described in Section 7.2.3, although the fundamental representation level of ELMo is character-based, the (intermediate) vectors generated through the bidirectional language model contain word-level information.

In conclusion, the best solution for constructing representation for the inputs is to leverage the combined granular levels from character to word, and to concept (**Combi**). With such a representation, the system is able to analyze user-specific behaviors regarding lexical choices and the consistency of sentiment on related topics.

8.5. Model Performance: A Summary

To summarize, the performance of the models proposed for the underlined task is demonstrated in Table 8.6 for a global comparison. Although the datasets used for the preliminary and advanced models are annotated in different ways, it is obvious that the models can offer better results for binary classification compared to the three-label classification. One of the difficulties in classifying neutral class lies in the confusion of separating subjective expressions with a neutral orientation and objective expressions such as the expressions *'I neither hate it nor like it'* and *'It is Monday today'*. Moreover, the difference between a neutral class and a positive or negative class can be less distinct than the difference between a positive and a negative classes. It can be imagined that given more classes, for instance, when performing emotion analysis, making an accurate prediction can be more complex because the classes are entangled. It remains unanswered how personalization would affect the analysis when increasing the number of classes significantly. In the meanwhile, there can also be problems in obtaining a desired level of agreement during the annotation process.

Two types of comparisons are emphasized when displaying the results: One is to compare between the utilized models and the other is to compare between the associated information. Results show that when using the same granular level, the models that integrate more personalized aspects have better performance; when using the same model, finer and more comprehensive representation corresponds to better performance. Apart from making comparisons, using the same model on various amount of information can cause an unbalanced model efficiency — the model that merges in-

Annotation	Model	Input Structure	Granular Level	Evaluation Metric			
				Avg. Recall	Pos. F1	Neg. F1	Accuracy
Manual (Sanders + SemEval)	Sentic	polarity values	concept	44.08	-	-	37.69
	G-SVM	concept(s), topic	concept	58.47	-	-	61.13
	P-SVM	AR	concept	58.62	-	-	61.47
	CNN	concept(s)$_e$	concept	53.60	-	-	54.81
	G-RNN	concept(s)$_e$, topic$_e$	concept	65.87	-	-	63.82
	Preliminary	AR$_e$ (preliminary)	concept	68.59	-	-	65.69
Automatic (Sentiment140)	RNN	AR	concept	-	74.67	74.01	74.35
	Attention	AR	concept	-	75.00	75.49	75.26
	eHawkes	AR	concept	-	75.34	76.52	75.96
	uni-Hawkes	AR	concept	-	75.12	76.87	76.04
		WE	word	-	76.06	77.06	76.58
		CW	concept, word	-	76.43	77.71	77.10
		DC	character	-	76.60	79.71	78.27
		Combi	concept, word, character	-	77.78	80.67	79.34
	user-Hawkes	AR	concept	-	74.94	77.48	76.28
		WE	word	-	76.61	78.41	77.55
		CW	concept, word	-	76.90	79.10	78.06
		DC	character	-	77.10	80.36	78.86
		Combi	concept, word, character	-	78.06	82.25	80.38

-$_e$: pre-trained embeddings as input.

Table 8.6.: Overall model performance.

formation from different sources may require a more complex setup than the one that uses single-typed information. In the experiments, different settings are allowed during the embedding process and other parts are shared across different acquisition of information. Despite the possibility of exploring the optimal model setup, it is unnecessary to test all the hyperparameters within a set range given the resilience of neural networks against minor variations. While employing the same information at the input, there are a large number of model variants that can be applicable to the underlined task as well; however, I focus on the (explicit) realization of the personalization aspect and regard the proposed models as an addible and extendable compartment in larger sentiment systems. In sum, the overall model performance verifies the effect of personalization in sentiment analysis from a macroscopic view whereas in the next chapter, the effect will be examined more closely.

Chapter 9

Key Factors for Personalization

Based on the experimental results demonstrated in the last chapter, certain detailed aspects are examined to identify the key factors in personalization that influence the performance. In particular, to relate to the assumptions considered by this research, the effect of the topic-opinion relation, the user frequency, the length of the past, and the information decay is studied.

9.1. Topic-Opinion Relation

Recall that in **Assumption II**, it says *'An individual's opinion towards a topic is likely to be consistent within a period of time, and opinions on related topics are potentially influential to each other.'* In the preliminary study, the personalized framework is evaluated on the combined, manually labeled datasets without using the associated topics in order to reflect the effect of topic-opinion relations. Such an evaluation is only possible with atomic representation where the topics are explicitly included. The setup of the indicated network is the same as the one used in the preliminary study but with one difference: $x_{t*} = [E_{concept}\ P_{concept}]_*$ is set at the input nodes before the user index instead of $x_{t*} = [E_{concept}\ E_{topic}\ P_{concept}\ P_{topic}]_*$ (Section 5.1.1). The experiment shows an accuracy of 55.36 and average recall of 54.29, which is worse than the performance of the preliminary model by a large margin (Accuracy: 65.69; Avg. Recall: 68.59, Table 8.6). This result indicates the benefit of associating sentiment with topics through the components E_{topic} and P_{topic}. Concerning the time period stated in the first half of the assumption, even though the corresponding period of time in each sequence is various, the presetting of time steps limits the consideration to recent events, which is consequently dependent on the user frequency.

For the atomic representation, the topics are important components since the concept-level granularity can be over-simplified which leads to an under-represented input formulation. In the case where no affective concepts can be extracted from the post, the opinion expressed by the user on the same topic in the past can be seen as an indicator for the sentiment of the current post. Furthermore, the second half of the assumption is leveraged through the application of the topic embedding network where related topics are represented with close proximity.

9.2. User Frequency

The number of posts a user has published in the time period of the experimented corpus plays a significant role in personalization. Intuitively, the more frequently a user posts, the more information the model is able to gain from the user's past, thus the more accurate the prediction can be. Experimental results indeed confirms such an intuition. In this section, I summarize the observations made during the implementation from where manually labeled data was used with infrequent users to investigating comparably more frequent users taken from automatically labeled data.

9.2.1. Observations from a General View

The performance with respect to different user frequencies in the preliminary study is shown in Table 9.1. Majority of the users have only tweeted once, and only 51 users have tweeted more than 5 times. The user with the highest number of tweets has 113 posts. This is due to the size limitation pertinent to manual annotation and the imbalance of user frequencies in general.

# Tweets per User	# User	Accuracy
> 5	51	74.25
3, 4, 5	206	66.71
2	714	62.82

Table 9.1.: Performance of the preliminary model with users of different numbers of tweets.

I take different intervals to show in this table because a certain number of samples are needed to provide meaningful results, e.g., no user has tweeted exactly 13 times in the experimented data. The results indicate that the more a user tweets, the more accurately the model is able to predict for the user. Consequently, a high level of personalization requires adequate user data. By treating all the users who tweeted once as one user, the system achieves an accuracy of 64.61 for this ‘one user’. The setting of the input sequence for this group of users is the same with the generalized recurrent network, however they are trained together with other users so that the network is enhanced by comparing between different users. Therefore, from this observation, I am confident to claim that the overall performance will increase if most users have tweeted more than 5 times, which is very likely in a real world scenario.

9.2.2. Observations with Frequent Users

Frequent users are targeted in the development of the advanced model. As the same as before, results in Table 9.2 show that the models can predict better for the users that have posted more frequently during the test period (19 days). The models in the table follow the description in Chapter 6.1 (also as implemented in Section 8.3), which are designed to evaluate different approaches for integrating time information in the analysis. Taking the users with frequency not less than 50 or 100 as examples, the improvement of the accuracy with respect to this conclusion is significant for all the models. The accuracies for the users who have not less than 100 posts are high, because the topics of the posts can be highly related within a short period and the time gaps are comparably low. Case I offers the best result in this range indicating that it is easier for the network to find the relations from the input when associating with highly frequent users.

Model	Accuracy (all)	Accuracy	Accuracy
		user frequency$\geqslant$ 50	user frequency$\geqslant$ 100
RNN Model	74.35	77.59	89.68
Basic Model	75.26	78.52	89.62
Case I	75.62	79.36	**91.02**
Case II	75.86	78.80	89.62
Case III	**76.13**	**79.95**	90.22

Table 9.2.: Model evaluation while using all the user data, or the users with frequency not less than 50 or 100.

When applying the two Hawkes process-based models, the performance for users with different frequencies can be seen in Figure 9.1. The x-axis corresponds to the lower bound of the user frequency. The lower bound for the illustration was taken because there are different numbers of users for each frequency, and many frequencies have no users to assign to. With both models, significant growths for each input formulation can be observed while increasing the lower bound of the frequency. Note that although the **Combi** formulation gives the overall best performance (Table 8.6), it can be seen from the figure that **Combi** does not give the best results in all the cases. For instance, when the user frequency is around 80, the **WE** formulation has the best accuracy in both models. However, such an observation is also restricted by the number of frequent users in general — with only 372 posts in the test set when the user frequency is at least 100, the performance is highly dependent on the remaining

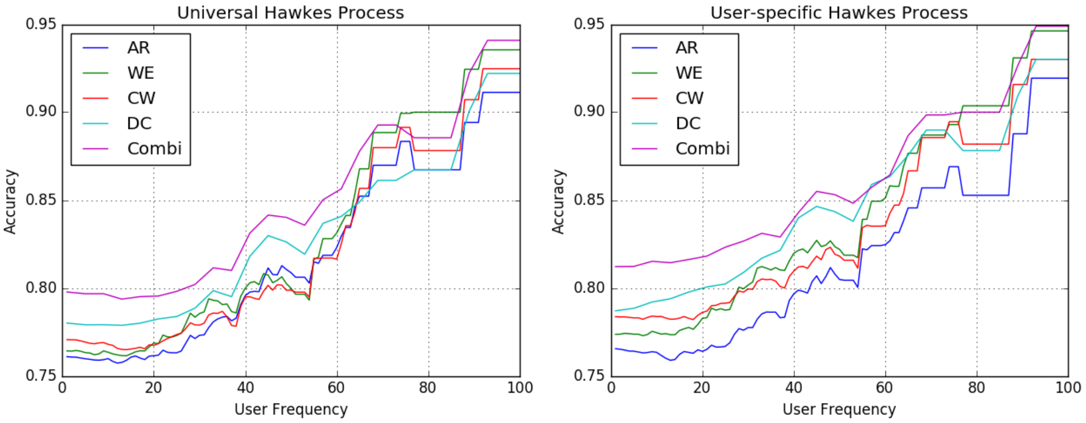


Figure 9.1.: Performance of the two Hawkes process-based models for users with different frequencies when applying different formulations.

three users. Another observation is that the **WE** formulation performs better than the **CW** formulation in higher user frequencies, but it has a lower overall performance because there are more users who have published less than 30 posts than the ones who have published more.

9.3. Length of the History

Following the experiments on user frequency, the impact of employing different numbers of historical posts is examined, where the number can be bounded by the user frequency. Observations are made for both manually and automatically labeled datasets where the latter has applied the input selection algorithm (Algorithm 1).

9.3.1. Observations from the Manually Labeled Data

Implemented on manually labeled data, an experiment is conducted by adding different numbers of past tweets in the input sequence. Given the distribution of user frequency in Table 9.1, there is no need to consider more than 20 past tweets. Recurrent networks assume that recent events have more impact, therefore more attention is given to closer nodes. Nevertheless, Table 9.3 shows that the network offers better results when relating to a longer history.

When considering only one previous post, the performance is very poor because two consecutive posts may not be related and there is not enough useful information from the previous post that can be memorized and extracted for the current one.

Number of Past Tweets	Avg. Recall	Accuracy
1	54.81	56.80
5	63.46	62.16
10	66.71	63.05
15	66.88	64.61
20	68.59	65.69

Table 9.3.: Performance of the preliminary model considering different numbers of past tweets.

When considering 10 past tweets, the system shows competitive results compared to the generalized recurrent network which considers 20 past tweets (Table 8.1). The maximum capability of this model can be further examined by a larger set of user data. This experiment shows that a rich set of user data and a network with a sufficient depth of input sequence are major influential factors of the system.

9.3.2. Observations from the Automatically Labeled Data

Figure 9.2 shows the performance of the models while using the **CW** formulation. The models are tested for T from 1 where no earlier posts are considered, to 20 after which no significant improvement can be observed due to the number of related posts a user normally published. For the case when the user history is not incorporated in the model ($T = 1, \Delta t = 0$), it can be deduced that $v' = \lambda + \epsilon\lambda'$, which leads to an accuracy of 73.87 with the universal ϵ and 74.46 with the user-specific ϵ (Equation 6.7).

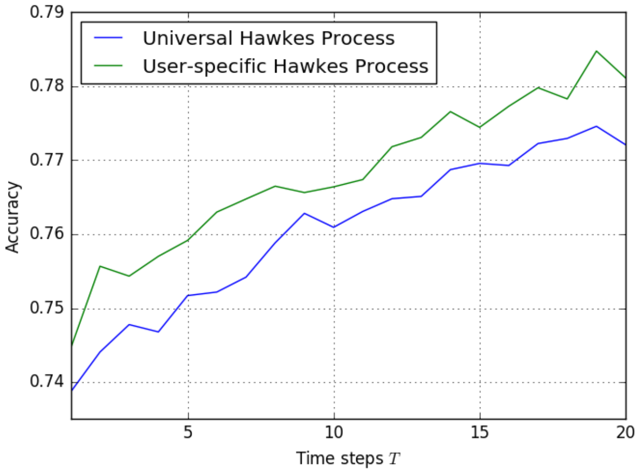


Figure 9.2.: Comparison of the two Hawkes process-based models while using different time steps.

Increases can be observed in both models when rising the number of time steps T. The increase indicates that the personalization is effective and earlier posts are valid contextual information. By using the selection procedure, the models with a smaller number of T (except when $T = 1$) take into account more related posts in the past. The increase grows slightly faster towards smaller numbers of T, which is also caused by the limitation of user frequencies in the experimented corpus. It can be foreseen that given a sufficient number of frequent users, the performance of the proposed models can be further improved.

9.4. User Behavior of Information Decay

Following the discussion in Chapter 6.2, the impact factors for the information decay are inspected while first set as constant values after which found during the training. Further, user-specific behaviors are visualized taking a number of users as examples.

9.4.1. Decay Factors

The values for the decay impact factor ε and decay rate β are first found experimentally by grid search given an empirical range. The results are illustrated in Fig 9.3. Because ε and β interact with each other on the rectified attention output, it is difficult to find a significant trend between these two values and the accuracy of the prediction. However, there is a slight tendency to offer better results with a smaller β. Comparing to the best results in Table 8.2 which are given by $\varepsilon = 0.7$ and $\beta = 0.01$, the worst result in the set range is 72.70 (accuracy) given by $\varepsilon = 0.9$ and $\beta = 0.1$ which is almost the same as the accuracy of the RNN model. Such a performance shows the importance of setting suitable parameters in the excitation function.

Based on the observations made with the empirical Hawkes process, it is likely that by setting a pre-defined range, the efficiency of using the process has not been fully unearthed. There is a possibility that the decay rate must be small in order to 'borrow' more information from the past to remedy the under-represented input. While a finer representation is used, the model can focus more on the current message compared to using coarser representations. Therefore, to refine, the two parameters are optimized together with other trainable weights through backpropagation. After the training, ε is below 0.1 and β is around 0.1 (with fluctuations inherent in neural networks) when applying the **DC** and **Combi** formulations. Indeed, when the text is well represented, the effect of the information decay can be less significant to the decision with the universal Hawkes process applied.

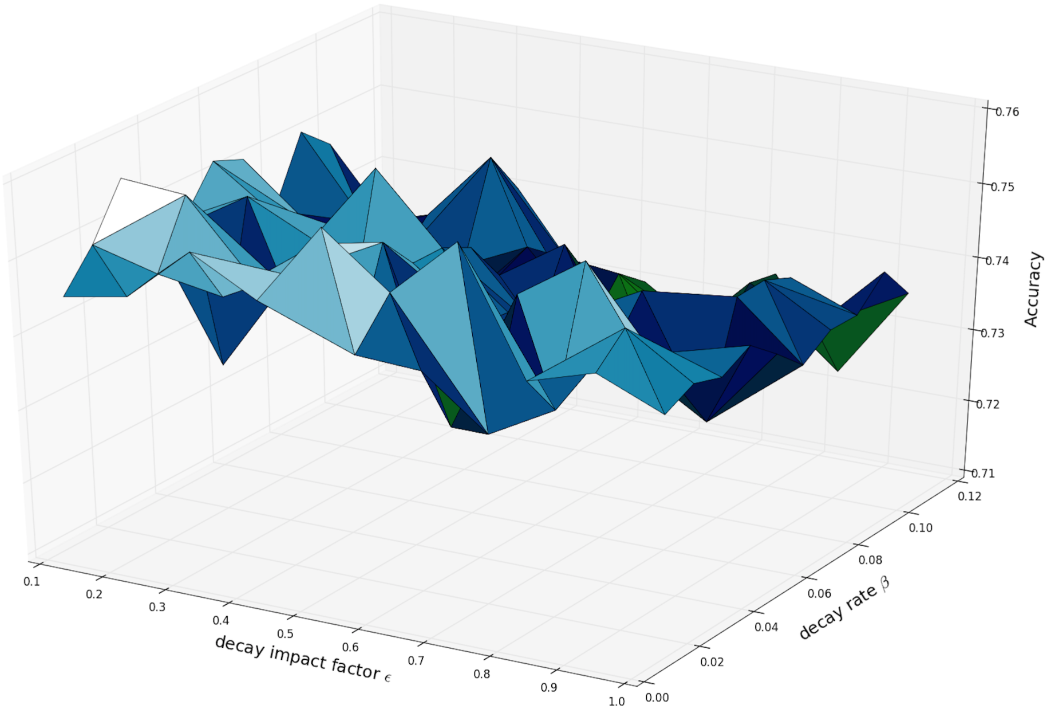


Figure 9.3.: A grid search on the parameters ε and β for the empirical Hawkes process.

9.4.2. Visualization of the User-specific Behavior

The user-specific Hawkes process is proposed to introduce the personalization in the decay behavior. As an example, Figure 9.4 illustrates a simulated process of the decay of four posts from a specific user. These four posts from the past have positive effects on the current post at t_0, and since they are from the same user, the effects decay at the same rate. At the right side of the figure, the output v' combines the original intensities of the events from different time with their intensities after the decay, where the decay is scaled with the user-specific importance factor ϵ_u.

In order to examine the user-specific Hawkes process trained during the implementation, in Figure 9.5, the intermediate calculations for the values of ϵ and β with respect to 10 random users are visualized. Each cell in the figure corresponds to the value of $a_{\epsilon i} * E_i(u)$ as in Equation 6.7 (top figure) or $a_{\beta_i} * E_i(u)$ as in Equation 6.8 (bottom figure) at dimension i for the respective user. While different results can be expected when giving different values of u, the relation across different dimensions when using the same u, i.e., between $a_{\epsilon i} * E_i(u)$ and $a_{\epsilon j} * E_j(u)$, or $a_{\beta_i} * E_i(u)$ and $a_{\beta_j} * E_j(u)$, can not be easily interpreted for the representation is implicit.

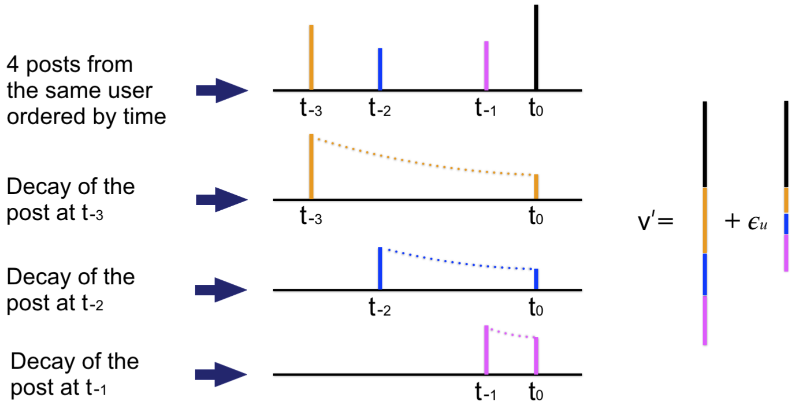


Figure 9.4.: An illustration of a simulated user-specific Hawkes process.

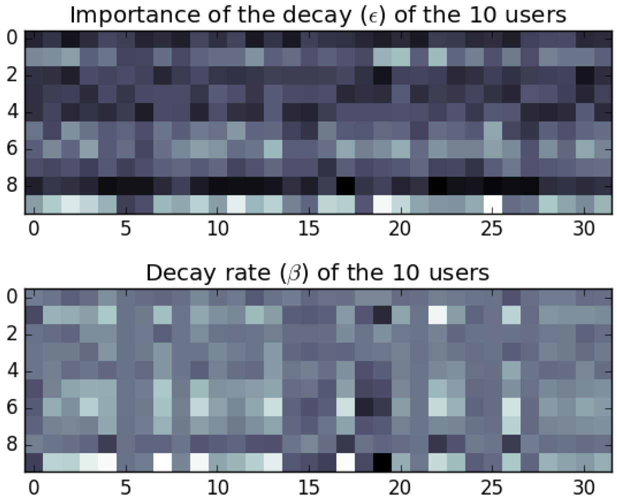


Figure 9.5.: The vectors of ϵ and β in the user-specific Hawkes process of the random 10 users.

The effect of the learned transformation vectors on the 10 users is illustrated. It can be seen that the last user in the figure (the one at the bottom line) has the greatest values for ϵ and β, which means that the decay factor has a great impact on the prediction for this user than the others but the influence from the past decays comparably fast — the user is affected a lot by recent events. In contrast, among the 10 users, the second last user is the least influenced by the past which is visualized in darker colors. In fact, the last user has posted comparably more frequently than the second last user with 109 tweets in 57 days and 44 tweets in 70 days respectively. This fact explains the reason for the fast decay of the information with respect to the last user — the time gaps between the posts are short and the decay behavior of this particular user must correspond to the pace of the publishing frequency, whereas the situation with the second last user is the opposite. The details of the two users can be found in Appendix A.2.

From this figure, it can be seen that the different decaying processes are indeed learned for different users with the vector transformation. One may argue that the behavior of the Hawkes process also depends on the time period of the experimented dataset; however, if an earlier post (outside of the training period) is highly relevant to the current one, the large value of λ_i can still prevail regardless the value of ϵ. To sum, the user-specific process reveals the dynamics in a user's sentiments throughout the user posting history and measures user behaviors individually to capture the nuance variations in the personalized modeling.

Part IV

Conclusions and Future Work

The work of this thesis is concluded in Part IV. First, the findings and conclusions of this research are highlighted in Chapter 10 where the theoretical implications that are reflected in the implementation and evaluation are also discussed. From the discussion, we see that the discoveries of the work bring significant meanings to the general research of sentiment analysis, neural networks, and NLP from a broader perspective. In Chapter 11, future work with respect to the model extensions are proposed. The extensions can be made in terms of methodology and application. Specifically, the embedding of the personalized sentiment model in an artificial companion was previously explained in Guo and Schommer [2017]. At last, Chapter 12 gives an overall summarization of the thesis.

Chapter 10

Findings and Theoretical Implications

Following the implementation and evaluation of the proposed models, I stress the main contributions of this research and make further conclusions on the theoretical implications from a broader view. In this chapter, the implications will be discussed from three aspects: the merging of various information types in sentiment analysis, the aspect of personalization in general NLP tasks, and the realization of information decay in a neural-based system.

10.1. Research Findings

This research shows that analyzing textual and contextual information related to individual differences helps understand a person's sentiment in the expressions. The work has employed three assumptions (Section 4.1.1) that are deduced from socio-psychological theories [Janis and Field, 1956; Nowak et al., 1990; Reiter and Sripada, 2002]. To leverage the assumptions, a personalized sentiment model is developed accordingly, and the design of the model has been elucidated in details in this thesis. A number of findings are discovered through the implementation of the model on Twitter data. In this section, the most significant findings of this work are summarized first from the theoretical aspect and then from the realization aspect.

From the theoretical side, the findings are prone to the type of text that is domain-independent and generated by users from social networks. In the context of personalization, the users that post frequently on the platform are the main research targets. In light of the discussion in Chapter 1 regarding the research gap, this study leads to the following conclusions:

- It is beneficial for sentiment analysis to consider the individuality in lexical choices which can be reflected in different text granules as well as in the topics that are associated with them.
- When considering single-typed representation, finer granular level can produce better results; the overall best solution for constructing input representation is to leverage the combined granular levels.
- Contextual information, such as the posting history of a user and the creation time of the posts, is valuable to the analysis and can be employed in order to improve model performance.
- The validity of a user's past information can diminish over time, and the consideration of such a phenomenon can be helpful for the modeling.
- The personalized analysis can be more advantageous for frequent users.

From the perspective of model realization, the following observations regarding the possible challenges associated with personalized sentiment modeling can be made:

- By adding user information in the input, the model can make user-specific predictions without requiring a separate adaptation for the indicated user, thus the data sparsity issue is resolved.
- The model performance is constrained by the data size and the number of historical posts considered by the system.
- After pre-selecting the historical posts that are relevant to the one being studied using topic similarity, the model can relate to the entire posting history of a user while the original setting of the model remains unchanged.

Specifically, as an extension of **Assumption II**, the study on the effect of time factor in this task has brought the following conclusions:

- While the attention mechanism offers flexibility with distant nodes by creating direct links between the output and each history, the Hawkes process that is activated inside the attention can alter the intensities of the links according to the time gaps between the past and the current posts.
- The user-specific Hawkes process is able to capture a user's behavior with respect to the information decay related to sentiment.
- The modified attention model that is shaped by the Hawkes process provides a meaningful alternative for modeling temporal sequences.

Based on these findings of this research, in the following sections, the possibility and the significance of applying the findings to other related tasks with different types of text are discussed.

10.2. Multi-aspect Information Fusion in Sentiment Analysis

The main focus during the development of the personalized model has been on performing document-level predictions for domain-independent text from social networks. A series of experiments have shown that for the text with a limited length, contextual information can be particularly valuable to the analysis. Especially, the improvement on the model performance delivered by associating the contextual information – the user, the past, and the time – has provided evidence for this observation. Further, the same observation can be reflected in the analysis of short texts in general. Moreover, the fusion of information can be comprehended as between textual and contextual information as well as among different granular levels in the text representation. The effect of both perspectives verified in this work also sheds a light on other related tasks in sentiment analysis.

10.2.1. The Role of Contextual Information with Short Texts

When conducting sentiment analysis on short texts, posts from social networks are often researched [Vo and Zhang, 2016; Wang et al., 2018a]. Without a precise definition of 'short texts', such posts represent a special type of text as discussed in Section 4.1.3, whereas there are other types that can be regarded as short texts as well. In Wu et al. [2018a], tweets and messages from Short Message Service were both used for short-text classification, however no distinction was made between the two texts. The messages created for the purpose of communication can be different from the posts on social platforms with respect to the recipient design. The tone of the messages can vary distinguishably towards different objects, while the contextual information concerning the speaker, the recipient, and the relation between them can be helpful in understanding the text and the sentiment. To build on the proposed model of this work, the past exchanges between the speaker and the recipient can serve as a contextual source that assists the analysis. In addition, time is a valid contextual information type for tracking the change of tone between them over a longer period. Thus, the sentiment model can be directly adapted by adding the recipient information at the input. Another example of short texts is news headlines as in Agarwal et al. [2016], where contextual information such as the background and the history of the publisher can be advantageous for discovering the stand of a topic, especially when comparing to news articles where the textual information alone may be sufficient for the analysis. Meanwhile, the immediateness of news makes time an important factor, and the decay of conveyed information in the news can be expected. Given these similar traits between the experimented text and the aforementioned short texts, improving the analysis of short texts by applying the contextual information in the same manner is plausible.

10.2.2. Information Fusion as a Solution

As employed in this work, information fusion can be a solution for certain challenging tasks in sentiment analysis. In Section 3.2.1, different information types and merging techniques were described, which could be explored for various tasks. When considering the fusion between textual and contextual information, aspect-level predictions with domain-specific text benefit from the solution as well, such as introduced in Section 2.2.3 where user and target information is utilized for personalized sentiment modeling on product and movie reviews. Based on the observations from the implementation, it can be inferred that information fusion is especially useful for the situation where the textual information at hand is restricted or ambiguous. For example, in the case where a user gives conflicting reviews for the same product, a decision can be made prone to the review published later by referring to the time information; in the text where the sentiment is subtle, the information of the expresser and the expresser's opinions on related topics can contribute to the analysis. Therefore, information fusion can be applied to general sentiment-related tasks as a supervision to filter out outdated information or to avoid inaccurate or biased understanding.

When considering the fusion among different granular levels in the text representation, it has been shown that finer and combined representation is more informative in personalized sentiment analysis. This merit can be interpreted from two aspects. The first one is to indicate that such a representation is more sensitive to variations in the sense that it can capture minor differences in the pattern pertaining to an expresser's lexical usage when the differences have appeared repetitively; the second one is to indicate that the representation is resilient to variations caused by mistake in which case the same pattern may not be repeated very frequently so that the final decision is not affected. These two aspects are useful in general sentiment analysis tasks where the text contains unseen words such as neologism and slang words or where the text is informally written.

10.3. Personalization in NLP

The experiments executed in this research have demonstrated the significance of personalization in sentiment analysis. For a frequent expresser, the personalized model is able to predict the sentiment of a present expression based on the analyzed behavior of the expresser in the past. In this way, the model provides a deeper language understanding that captures individualities. To further the conclusion, the potentials to breach from the analysis of sentiment towards benefiting general tasks in NLP are discussed in this section.

10.3.1. From Natural Language Understanding to Natural Language Generation

Natural language understanding is a fundamental task in NLP for its use in many applications such as text categorization and automated reasoning. It also has a strong connection to sentiment analysis in a way that improved language understanding can enhance the prediction of sentiment and the performance of a sentiment model can reveal the effectiveness of a language understanding system. According to the observations of this work, the personalization aspect following the assumptions introduced in Section 4.1.1 has shown to boost the performance of the sentiment model, which suggests the potential in strengthening the language understanding ability.

The discovered properties pertinent to individualities unearth significance in general NLP tasks. Besides performing instant predictions based on the learned knowledge from the text for the purpose of classification, the properties can be extended to the tasks that transform the learned knowledge into another sequence of text such as text summarization and question answering. As many natural language generation tasks can be seen as a consecutive step of a language understanding system, the deepened understanding brings capacity to language generation. By considering the personalization aspect, the generated text can preserve the individualities observed from the initial input with respect to the lexical choice of the expresser, the sentiment expressed in the text, and the opinions on related topics. Therefore, a sequence of text dedicated to the original expression and expresser can be generated accordingly.

10.3.2. The Potential of Personalization: An Example with Machine Translation

Machine Translation, as one of the most researched tasks in NLP, has led the advance of the development of many subtasks in the field. However, most researches are performed at a population-level where individualities are neglected. In Wintner et al. [2017], the authors have proposed the concept of personalization in machine translation and have shown that the expresser's gender conveys a special message in the text which should be preserved in the translation. This study encourages future investigations on other demographic traits that may be helpful for the translation. Michel and Neubig [2018] furthered the study to modeling the speaker explicitly by exploiting a neural-based system with a standard sequence to sequence model, where a similar situation to the work in this thesis regarding the data sparsity was realized. Their approach was designed from the perspective of domain adaptation and works by altering the bias in the output $softmax$ function. Nevertheless, the adaptation method relies on the similarity among the expressers to a certain degree, whereas the model described in this thesis intends to be free of such an assumption.

The addition of the time factor in this research brings potential to the personalization in machine translation. An example is the processing of polysemy. By fusing the expresser and the time information, the meaning of a polysemy can be deduced not only from the immediate context but also from the preferred usage of the expresser and the prevalent usage at the time of the expression. The performance of a translation system can be improved when the meanings of polysemy are correctly inferred.

10.4. Information Decay with Neural Networks

By incorporating the Hawkes process in the attention mechanism, it offers a possibility to model information decay with precise time gaps in neural networks. This revision of the attention model can benefit tasks which contain temporal sequences at the input, such as stock market analysis based on sequences of tractions and fraud predictions based on past fraudulent activities. It can be applied in combination with different network structures in the way attention model normally functions. In this work, the revised attention model is used together with a hierarchical recurrent network, whereas other possibilities can be considered with the expectation of acquiring similar learning behaviors. For instance, in the work of Vaswani et al. [2017], a model architecture named ‘Transformer’ was developed, which was based solely on the attention mechanisms. In the architecture, a positional encoding is used to compensate the lost of ordering caused by dispensing the recurrent and convolutional structures. To facilitate the information decay factor in this architecture, the revised attention model can be adapted by altering the add function with Hawkes process and the time information can be added as an auxiliary input. By doing this, the positional encoding can be discarded while the ordering of the elements as well as the time gaps between them are preserved.

In NLP, the usefulness of the additional functionality depends on the nature of the input: when the input sequences are sentences with words as the elements, the concept of time gaps does not apply since only the order of the words matters. Concerning the representation levels in the hierarchical network discussed in Section 3.1, the functionality takes effect at the higher hierarchy (the post level in Figure 3.1) which is commonly referred to as the document-level where the element, i.e., document, normally appears in a nonconsecutive manner and associates with timestamps. There can also be multiple hierarchies in the document-level, for instance, periodicals can have a number of volumes and issues with fixed or flexible publication dates, and in an issue there can be a number of articles that were created at different dates. In this case, the information decay can be considered in the analysis of the text and the revised attention model can be adopted.

Chapter 11

Model Extensions

The implications brought through the development of the proposed models have left much room for further applications. In this chapter, I discuss model extensions from the technical perspective with respect to the information representation and model design as well as from the applicational perspective with respect to the multilingual analysis and embedding in an artificial companion.

11.1. Enriched Information Representation

This thesis has shown the importance in applying a suitable input formulation on textual and contextual information. Thus, enriching the information used in the input is one of the ways to boost the model performance. Here, possibilities are discussed from the views of textual and contextual aspects respectively.

11.1.1. Embedding Phonetic Representation

Because the informal text used in this research deviates from the language standard, the representation of input text plays a significant role in improving the performance. It has been shown that combining various representations offers significant improvement in the performance. For that reason, other representation approaches should be considered. As an example, in the future work, phonetic representation can be exploited which can provide another source of information for the informal text. The posts can be transcribed into phonetic sequences, for instance, by using the International Phonetic Alphabet, in order to handle certain misspellings and to study the trend of using letters with similar pronunciations as substitutions (such as using 'gr8' for ‘great’). To be used in the model, the phonetic representation can be embedded separately and concatenated with other representations before the recurrent block. By passing the embedded information through the personalized model, the individuality concerning phonetic relations can be learned and leveraged in the analysis. Alternatively, phonetic algorithms can also be applied to encode the text, such as Soundex and Metaphone [Jordão and Rosa, 2012]; however phonetic representation may need to attend to the variation in the language being studied.

11.1.2. Exploring Social Relations

Other than considering alternatives in textual representation, different types of contextual information should be explored as well for a better understanding of individual behaviors on social platforms. For instance, social relations have been used in some literature as mentioned in Section 2.2.3 to reflex certain personality traits. In addition, by exploring the time factor, social relations can be used to help identify abnormalities in the change of sentiment, especially in the case when a user is exceptionally stimulated by other users or special events which causes untypical behaviors. In turn, such an analysis would also help predict future abnormalities when encountering similar behaviors of the influencers.

11.2. Model Variants

Extensions can be made by discovering alternatives of the model design. Different network structures and functions can be inspected for the intended task inspired by the model proposed in this work.

11.2.1. Alternatives of the Recurrent Block

In this research, recurrent networks are used as the central part of the model for relating to the information in the past, whereas other structures can be applied in the future work. While adding the time factor in the model through the Hawkes process, the order of the elements in an input is preserved such that the recurrent structure can be replaced by a convolutional or a full-attentional structure. Although in the preliminary study, the effect of using convolutional network was examined and the performance was below that of the recurrent network, it is possible to remedy the situation by leveraging the observations made in the experiments. The underperformance of the convolutional network was caused by the garbling of the order of concepts in a post while atomic representation was used, whereas a different input formulation can be helpful to restore the order of the input so that adjacent embeddings of the elements are correlated. Also, the decay behavior can be learned by adopting the revised attention model on top of the convolutional network.

Another possibility is to apply a residual network when the input is enriched and entangled or when the network is too deep [Conneau et al., 2017]. As shown, the combined representation benefits the performance the most. However, certain representation such as ELMo may be constituted by a deep structure, which can cause a lost of focus on the initial input while passing through a hierarchical network with many layers in between. By using a residual network, additional links between the input and output can be created, for example, to add a link between the user encoder and the output, and as a result, the prediction can be made prone to the information at the layer where the emphasis is desired.

11.2.2. Variants of the Hawkes Process

The Hawkes process has been applied for its ability in modeling information decay over time. As the excitation function, the exponential decay was employed in the implementation, whereas other choices such as power law function can be tested in the future work. Instead of assuming a linear relation between the background intensity and the decay effect, nonlinear processes can be considered [Zhu, 2013]; while social relations are added in the modeling, multivariate Hawkes processes can be used to merge the user-specific process with the influence from the other connected users [Liniger, 2009]. However, to be applied in a neural-based system, the possibility to train the process with backpropagation has to be discussed.

11.2.3. A Path to Explainable Deep Learning

The lack of explainability has been one of the major downsides of deep neural networks, which also makes this method arguable to many applications. The explainability corresponds to the ability of interpreting the ‘arrival’ of a particular outcome given an unseen data sample, while in the context of neural networks, it is to discover the exact feature(s) or intermediate value(s) on which a prediction relies. There are certain visualization techniques that offer some transparency on the subject. For example, heatmaps can be used to illustrate the contribution of an input made to a specific decision. Samek et al. [2016] have argued that the judgment of the quality of a heatmap can be subjective due to human intervention, therefore, they have proposed a quantitative approach to evaluate different heatmap algorithms in an objective and automated manner. It was found that the layer-wise relevance propagation algorithm provides a better explanation comparing to other tested methods. While their approach was implemented on images, Arras et al. [2017] investigated the use of the algorithm on recurrent networks for sentiment analysis and demonstrated its ability in detecting influential patterns and monitoring classifier behaviors. Similarly, the algorithm can also be adopted in the model proposed in this thesis so that different lexical patterns can be inspected for different users. In addition, since time information is used as a part of the input, the algorithm would bring insights on the interaction among the content, the time, and the sentiment of a number of posts from the same user. Thus further, the improved explainability can offer a better understanding towards user behaviors and assist on selecting suitable information and methods to carry out the analysis.

11.3. Multilinguality in Personalized Sentiment Model

It is reported by the European Commission that the majority of European people are able to speak more than one language [Eurobarometer, 2006]. Such a phenomenon motivates the research of multilingual sentiment analysis and brings significant meaning in studying language preferences in personalized models. Johnson et al. [2017] added a language identifier in the neural machine translation system to enable a multilingual translation without constructing a model for each language pair, which can be of a motivation to employ the same concept for the language adaption. However, certain modifications are needed given the difference in the nature of the two tasks. Moreover, they have stated that the language index is not for the source language but for the target language — the language of the source can be inferred by the network; however in multilingual sentiment analysis, the source language can be indicated to emphasize the preferred language use of the user when referring to certain terms (words, phrases or concepts). The acquired knowledge of such preferences can be helpful in some further applications of the analysis (see Section 11.4.1) as well as in dealing with words that have different meanings in different languages such as 'Gift' meaning *poison* in German and 'pain' meaning *bread* in French, or code-switching (Section 11.3.2).

11.3.1. Multilingual Sentiment Analysis

The challenge of multilingual sentiment analysis lies in the lack of resource of certain languages and the fact that sentiment lexicon is highly language dependent [Can et al., 2018]. While analyzing modern social text, certain information such as emoticons and emojis can assist in discovering sentiment in text [Cui et al., 2011]; however applying a machine translation mechanism before the detection of sentiment is more common [Denecke, 2008; Balahur and Turchi, 2012]. To be used as an extension of the model proposed in this work, the language index can be added to the representation so that a message x_* is represented by $\{concepts, topics, negations, user\_id, language\_id\}_*$ when using atomic representation as an example. By adding the $language\_id$, the user's preference in language choices can be learned by the network. However, such information is only valuable when there is an overlapping between the corpora of the languages with the same user. Concerning the personalized modeling in multilingual sentiment analysis, the use of the language can have imbalanced frequencies for each user. The impact on the modeling caused by this usage is to be examined.

11.3.2. Towards a Solution for Code-switching

Code-switching (or code-mixing) indicates the switching of languages within a single context [Lipski, 1978; Muysken et al., 2000]. The aspect of code-switching is rarely researched in the field of sentiment analysis given the challenges of processing the text [Çetinoğlu et al., 2016]. This aspect is reflected in the research of sentiment analysis for sentences like *'I enjoy speaking English und Deutsch'*. In personalized sentiment analysis, the information of the use of language for specific concepts is helpful for understanding a user's perspectives. When concepts are mapped to a vector space, the ones from different languages but with similar meanings are placed closer to each other. Imagine that one particular user uses one of the concepts differently caused by a language drift or the user's proficiency in the language. By analyzing the usage, this particular concept may need to be relocated in the vector space or a certain transformation may be expected when associating with this user. Potentially, the same relocation or transformation may be needed for other (related) concepts from the same language. Without specifying the used language, only the literal meaning of the previous appearance is encoded and the language-specific shift of concepts is discarded. As a solution to integrate such a concept shift in the network, a language index can be given for each concept: A message x_* is represented by $\{\{concept, language\_id\}^n, topics, negations, user\_id\}_*$. The language index is embedded together with the concept instead of being added to the entire message in order to analyze the code-switching. In this way, a joint representation for each concept and language pair $\{concept, language\_id\}$ is learned before combining with the topics, negations and user identifier.

A major challenge of this research is to find a suitable corpus for training the network. To acquire a corpus with code switching occurring in the texts published by a sufficient number of frequent users is overly demanding. Alternatives can be proposed such as to browse the database of the social platforms substantially in order to extract such texts. Moreover, different methods to generate the representation of concept and language pairs can be developed.

11.4. Further Application: Embedding of the Personalized Sentiment Model in an Artificial Companion

As one of the possible further applications, the introduced model can be integrated in companion systems to improve communication experience. The main concept of this integration is to offer a shift in sentiment modeling that is tailored by the person a companion encounters with.

11.4.1. Model Integration into a Companion

Companion technology, as a field of cross-disciplinary research, has played different roles in respect of the areas of application [Biundo et al., 2016]. For a diverse number of applications such as social robots [Paiva et al., 2014] and elderly companions [Tsiourti et al., 2016], the detection of emotions and emotional states is a primary task for an interactive communication. Emotions, or sentiments to be more general, can be analyzed based on different resources, in which text or speech is one of the most direct signals that can be easily captured. Although it is easy to adapt an existing sentiment model to a companion system, it is much more desired to modify the model to acquire a higher level of user understanding. The literal sense of the name 'companion system' implies that such an agent should be able to provide companionship for a user or a group of users, therefore the underlined system has to offer an individualized analysis.

The individuality is mainly operated by evoking stored memory, and is essential to support a long-term relationship [Ho et al., 2010; Lim, 2012]. A single personalized model is sufficient when the companion is designed to serve one user, while multiple models are desired when serving a group of users. In the latter case, an extended storage is required and is featured with an indirect comparison between users. As a better alternative, the model proposed in this work can be used upon a traditional sentiment model for multi-user scenario without creating multiple models.

11.4.2. Multi-user scenario

There exists situations where an agent is applied to serve more than one user, for example a robot that receives requests from a group of users or an introductory companion that engages itself in a business fair. These situations reveal some desired features of a companion: 1) a companion must differentiate between different users; 2) it should be able to get access to the information associated with each user, and guess a user's attitude towards a potential topic even when lacking context; 3) it should be able to compare between users; and 4) it should analyze the users' lexical and language preferences in the expressions to make user-tailored responses.

For each encounter between a companion and a user, the information required by the companion includes the user's identity, the text or speech uttered by the user, and the time flag of this encounter. Such information can formulate an exact match of the input sequence of the proposed model. To comply with the features mentioned in the last paragraph: the differentiation between users can be accomplished by external sensors and represented by the user identifier; the model is able to predict a user's sentiment on a topic if a related topic has been discussed between the companion and the user in the past, and a reference of opinion or emotion to a related topic is evoked from the memory; the comparison between users is enabled using the user identifier; furthermore, the preferences on lexical and language choices can be fulfilled as well by analyzing concept – language pairs.

11.4.3. Challenges

Although the embedding of the model seems straightforward, there remains some challenges. At each encounter, the information in the memory cells that is related to the topic of the current utterance and the associated user is activated, and unrelated information is automatically forgotten. The memory can be traced back to the first encounter which is unnecessary in certain situations. For the scenario that an excessive number of users are engaging with the companion, it may cause a significant outburst and an increased amount of computation if an offline storage is used. For such an application, an additional forgetting mechanism should be purposed. Moreover, the companion is favored with the ability of constantly learning alongside the user(s), thus a transfer learning which updates the model after each encounter should be performed. However, the labelling mechanism must be defined for the training process. A solution is to apply a semi-supervised learning that an encounter is trained with a label when there is a clear feedback from the user. Alternatively, a third party can be involved in labelling as an external observer. Another issue is to make the knowledge of the companion always up to date. For that, one can either update the embedded knowledge base manually or connect the companion with social platforms so that an automatic update is possible. An external resource of knowledge for a companion to understand new topics and to track public opinions can be helpful to avoid biased understanding from the users. Further uses of the model for such applications are yet to explore, and the performance is to be evaluated.

Chapter 12

Overall Conclusion

In this doctoral work, a *personalized sentiment model* which is able to capture users' individualities in expressing sentiment has been developed. The research targets users' posts from social networks and exploits textual and contextual information to enhance the analysis. The development of the model involves two stages: one is a preliminary study that is meant to assess the demand of studying the individualities in the defined task and the other is an advanced study that aims to further research the personalization aspects. To evaluate the effectiveness of including user information in sentiment analysis, the preliminary model is built based on three assumptions deduced from psychological theories. The assumptions reflect the individuality from different aspects and the model is designed accordingly. Concepts appearing in the text are used to represent people's lexical choices; the topic is added in the text representation to include the topic-opinion relations; public opinions are used at the input in order to find connections between individual and public opinions. A simple recurrent neural network is applied for this task which is able to relate the information of the current post with the posts in the past. Moreover, the issue of data sparsity inherent in personalized modeling is addressed by adding a user identifier in each input sequence. The preliminary model is evaluated with a combined, manually labeled Twitter dataset, and the effect of introducing user data in the model is verified by comparing to five baseline models. From the evaluation, the key factors of a personalized sentiment model are discovered. It can be concluded that the topic-opinion relation, the user frequency, and the number of the past posts considered in the network are the major factors that influence the performance of the model.

Given the positive results of the preliminary model, an enhanced model that focuses on frequent users is proposed. The enhanced model is a hierarchical network that refines the input formulation from the coarser-grained atomic representation to more sophisticated representations and applies additional embedding layers to merge different types of information. Furthermore, attention mechanism is used at the output of the recurrent network which helps the network to concentrate on related but distant posts. To further on this goal, an input selection algorithm is developed in order to choose related posts from the entire history of a user. To consider different gaps between the posts, a novel approach is introduced aiming to shape the attention output with Hawkes process. By using this approach, the attention on the related posts is boosted and the effect fades by a certain decay rate on the distance between these posts and the current post. Thus, a decay of information over time can be modeled in the network. Furthermore, user-specific Hawkes processes are proposed by adopting a factor transformation on the encoded user so that the control parameters of the process are learned taking the different user behaviors into account. This enhanced model is tested on a larger, automatically labeled dataset with users who have tweeted at least 20 times before a pre-defined timestamp, and improvements are shown after adding the attention layer with Hawkes process.

The results from these two models have brought significant meanings of applying a personalized sentiment model. It can be learned that the individualities have substantial influence on sentiment analysis and can be easily captured by the proposed models. It also shows that the representation method and the information taken at the input have a great impact on the analysis. Finer and combined representations can carry more information that is advantageous to the system. On the technical side, traditional recurrent neural networks neglect the effect of various gaps between the nodes which can be an important factor in many tasks. As shown, the Hawkes process can be combined with the attention model and recurrent networks to compensate such lack of information, and the effect of using different variants of Hawkes process is yet to be explored. The combinations with respect to the input information and the network structure can provide a deeper language understanding that is profitable for general sentiment analysis and NLP tasks.

The improvements of the preliminary model and the enhanced model have opened up new opportunities for future research. To generalize the use of the proposed models, the performance can be tested by evaluating with finer-labeled sentiments or emotions. It is also possible to use these models on existing sentiment models that do not concern user information in the prediction in order to enhance the performance. As an extension on the field of application, the personalized model can be used in an artificial companion that is adapted under a multi-user scenario to improve communication experience by offering user-tailored responses.

Part V

Supplements

Chapter A

Additional Information on the Data

A.1. Data Separation for the Sentiment140 Corpus

Sentiment140	Starting Time	2009-04-06 22:19:57
	Ending Time	2009-06-25 10:28:28
	# Training Samples	82,361
	Timestamp 1	2009-06-03 00:00:00
	# Validation Samples	16,437
	Timestamp 2	2009-06-07 00:00:00
	# Test Samples	23,202

Table A.1.: The pre-set timestamps for the data separation and the number of samples in each segment.

A.2. The Two Particular Users from the Sentiment140 Corpus

As mentioned in Section 9.4.2, for the User (User-id: 5toSucceed) with the greatest values for ϵ and β among the 10 exemplary users:

- The first post was created at 2009-04-20 00:48:25;
- The last post was created at 2009-06-15 22:59:40.

For the User (User-id: 4evaurgirl) with the lowest values for ϵ and β among the 10 exemplary users:

- The first post was created at 2009-04-17 21:15:32;
- The last post was created at 2009-06-25 09:37:27.

Bibliography

Agarwal, A., Sharma, V., Sikka, G., and Dhir, R. (2016). Opinion mining of news headlines using SentiWordNet. In *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*, pages 1–5. IEEE.

Araque, O., Corcuera-Platas, I., Sanchez-Rada, J. F., and Iglesias, C. A. (2017). Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications*, 77:236–246.

Arora, S., Li, Y., Liang, Y., Ma, T., and Risteski, A. (2018). Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6:483–495.

Arras, L., Montavon, G., Müller, K.-R., and Samek, W. (2017). Explaining recurrent neural network predictions in sentiment analysis. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 159–168.

Bacry, E., Mastromatteo, I., and Muzy, J.-F. (2015). Hawkes processes in finance. *Market Microstructure and Liquidity*, 1(01):1550005.

Baecchi, C., Uricchio, T., Bertini, M., and Del Bimbo, A. (2016). A multimodal feature learning approach for sentiment analysis of social network multimedia. *Multimedia Tools and Applications*, 75(5):2507–2525.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.*

Balahur, A. and Turchi, M. (2012). Multilingual sentiment analysis using machine translation? In *Proceedings of the 3rd workshop in computational approaches to subjectivity and sentiment analysis*, pages 52–60. Association for Computational Linguistics.

Baziotis, C., Pelekis, N., and Doulkeridis, C. (2017). Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In

Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pages 747–754.

Bespalov, D., Bai, B., Qi, Y., and Shokoufandeh, A. (2011). Sentiment classification based on supervised latent n-gram analysis. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 375–382. ACM.

Biundo, S., Höller, D., Schattenberg, B., and Bercher, P. (2016). Companion-technology: an overview. *KI-Künstliche Intelligenz*, 30(1):11–20.

Blei, D. M. and Lafferty, J. D. (2009). Topic models. In *Text Mining*, pages 101–124. Chapman and Hall/CRC.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Brody, S. and Elhadad, N. (2010). An unsupervised aspect-sentiment model for online reviews. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 804–812. Association for Computational Linguistics.

Cambria, E. (2016). Affective computing and sentiment analysis. *IEEE Intelligent Systems*, 31(2):102–107.

Cambria, E., Fu, J., Bisio, F., and Poria, S. (2015). Affectivespace 2: Enabling affective intuition for concept-level sentiment analysis. In *AAAI*, pages 508–514.

Cambria, E. and Hussain, A. (2015). *Sentic computing: a common-sense-based framework for concept-level sentiment analysis*, volume 1. Springer.

Cambria, E., Livingstone, A., and Hussain, A. (2012). The hourglass of emotions. In *Cognitive behavioural systems*, pages 144–157. Springer.

Can, E. F., Ezen-Can, A., and Can, F. (2018). Multilingual sentiment analysis: An RNN-based framework for limited data. In *Proceedings of the ACM SIGIR 2018 Workshop on Learning from Limited or Noisy Data (LND4IR'18).*

Cao, Q., Shen, H., Cen, K., Ouyang, W., and Cheng, X. (2017). Deephawkes: Bridging the gap between prediction and understanding of information cascades. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1149–1158. ACM.

Carvalho, P., Sarmento, L., Teixeira, J., and Silva, M. J. (2011). Liars and saviors in a sentiment annotated corpus of comments to political debates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 564–568. Association for Computational Linguistics.

Çetinoğlu, Ö., Schulz, S., and Vu, N. T. (2016). Challenges of computational processing of code-switching. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 1–11.

Chamlertwat, W., Bhattarakosol, P., Rungkasiri, T., and Haruechaiyasak, C. (2012). Discovering consumer insight from Twitter via sentiment analysis. *J. UCS*, 18(8):973–992.

Chassang, G. (2017). The impact of the EU general data protection regulation on scientific research. *ecancermedicalscience*, 11.

Chen, H., Sun, M., Tu, C., Lin, Y., and Liu, Z. (2016a). Neural sentiment classification with user and product attention. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1650–1659.

Chen, P., Sun, Z., Bing, L., and Yang, W. (2017). Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 452–461.

Chen, T., Xu, R., He, Y., Xia, Y., and Wang, X. (2016b). Learning user and product distributed representations using a sequence model for sentiment analysis. *IEEE Computational Intelligence Magazine*, 11(3):34–44.

Cheng, X. and Xu, F. (2008). Fine-grained opinion topic and polarity identification. In *LREC*, pages 2710–2714.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Colleoni, E., Arvidsson, A., Hansen, L. K., and Marchesini, A. (2011). Measuring corporate reputation using sentiment analysis. In *Proceedings of the 15th international conference on corporate reputation: navigating the reputation economy.*

Conneau, A., Schwenk, H., LeCun, Y., and Barrault, L. (2017). Very deep convolutional networks for text classification. In *15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017*, pages 1107–1116. Association for Computational Linguistics (ACL).

Cui, A., Zhang, M., Liu, Y., and Ma, S. (2011). Emotion tokens: Bridging the gap among multilingual twitter sentiment analysis. In *Asia Information Retrieval Symposium*, pages 238–249. Springer.

Cui, H., Mittal, V., and Datar, M. (2006). Comparative experiments on sentiment classification for online product reviews. In *AAAI*, volume 6, page 30.

Dave, K., Lawrence, S., and Pennock, D. M. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, pages 519–528. ACM.

Dehghani, M., Gouws, S., Vinyals, O., Uszkoreit, J., and Kaiser, Ł. (2018). Universal transformers. *arXiv preprint arXiv:1807.03819*.

Denecke, K. (2008). Using SentiWordNet for multilingual sentiment analysis. In *Proceedings of the 24th IEEE International Conference on Data Engineering Workshop (ICDEW 2008)*, pages 507–512. IEEE.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Diao, Q., Qiu, M., Wu, C.-Y., Smola, A. J., Jiang, J., and Wang, C. (2014). Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 193–202. ACM.

Dos Santos, C. N. and Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. In *COLING*, pages 69–78.

Dou, Z.-Y. (2017). Capturing user and product information for document level sentiment analysis with deep memory network. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 521–526.

Eurobarometer, S. (2006). Europeans and their languages. *European Commission*.

Fang, X. and Zhan, J. (2015). Sentiment analysis using product review data. *Journal of Big Data*, 2(1):5.

Galavotti, L., Sebastiani, F., and Simi, M. (2000). Experiments on the use of feature selection and negative evidence in automated text categorization. In *International Conference on Theory and Practice of Digital Libraries*, pages 59–68. Springer.

Ghosh, S., Vinyals, O., Strope, B., Roy, S., Dean, T., and Heck, L. (2016). Contextual LSTM (CLSTM) models for large scale NLP tasks. *arXiv preprint arXiv:1602.06291*.

Go, A., Bhayani, R., and Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1(12):2009.

Gonçalves, P., Araújo, M., Benevenuto, F., and Cha, M. (2013). Comparing and combining sentiment analysis methods. In *Proceedings of the first ACM conference on Online social networks*, pages 27–38. ACM.

Gong, L., Al Boni, M., and Wang, H. (2016). Modeling social norms evolution for personalized sentiment classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 855–865.

Gong, L., Haines, B., and Wang, H. (2017). Clustered model adaption for personalized sentiment analysis. In *Proceedings of the 26th International Conference on World Wide Web*, pages 937–946. International World Wide Web Conferences Steering Committee.

Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J. (2016). LSTM: A search space Odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232.

Guo, S., Höhn, S., and Schommer, C. (2019a). Looking into the past: evaluating the effect of time gaps in a personalized sentiment model. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 1057–1060. ACM.

Guo, S., Höhn, S., and Schommer, C. (2019b). A personalized sentiment model with textual and contextual information. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL).*

Guo, S., Höhn, S., and Schommer, C. (2019c). Topic-based historical information selection for personalized sentiment analysis. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges 24-26 April 2019.*

Guo, S., Höhn, S., Xu, F., and Schommer, C. (2018a). PERSEUS: a personalization framework for sentiment categorization with recurrent neural network. In *International Conference on Agents and Artificial Intelligence, Funchal 16-18 January 2018*, page 9.

Guo, S., Höhn, S., Xu, F., and Schommer, C. (2018b). Personalized sentiment analysis and a framework with attention-based Hawkes process model. In *International Conference on Agents and Artificial Intelligence*, pages 202–222. Springer.

Guo, S. and Schommer, C. (2017). Embedding of the personalized sentiment engine PERSEUS in an artificial companion. In *2017 International Conference on Companion Technology (ICCT)*, pages 1–3. IEEE.

Gurini, D. F., Gasparetti, F., Micarelli, A., and Sansonetti, G. (2013). A sentiment-based approach to Twitter user recommendation. *RSWeb@ RecSys*, 1066.

Harris, J. R. (2010). *No two alike: Human nature and human individuality.* WW Norton & Company.

Hatzivassiloglou, V. and McKeown, K. R. (1997). Predicting the semantic orientation of adjectives. In *Proceedings of the 35th annual meeting of the association for computational linguistics and eighth conference of the european chapter of the association for computational linguistics*, pages 174–181. Association for Computational Linguistics.

Hawkes, A. G. (1971). Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90.

He, H., Wu, L., Yang, X., Yan, H., Gao, Z., Feng, Y., and Townsend, G. (2018). Dual long short-term memory networks for sub-character representation learning. In *Information Technology-New Generations*, pages 421–426. Springer.

Ho, W. C., Dautenhahn, K., Lim, M. Y., and Du Casse, K. (2010). Modelling human memory in robotic companions for personalisation and long-term adaptation in HRI. In *BICA*, pages 64–71.

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM.

Hu, X., Tang, L., Tang, J., and Liu, H. (2013). Exploiting social relations for sentiment analysis in microblogging. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 537–546. ACM.

Iskandar, B. (2017). Terrorism detection based on sentiment analysis using machine learning. *Journal of Engineering and Applied Sciences*, 12(3):691–698.

Jakob, N. and Gurevych, I. (2010). Extracting opinion targets in a single-and cross-domain setting with conditional random fields. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 1035–1045. Association for Computational Linguistics.

Janis, I. L. and Field, P. B. (1956). A behavioral assessment of persuasibility: Consistency of individual differences. *Sociometry*, 19(4):241–259.

Jia, L., Yu, C., and Meng, W. (2009). The effect of negation on sentiment analysis and retrieval effectiveness. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1827–1830. ACM.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., et al. (2017). Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Johnson, R. and Zhang, T. (2015). Effective use of word order for text categorization with convolutional neural networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 103–112.

Jordão, C. C. and Rosa, J. L. G. (2012). Metaphone-pt_BR: the phonetic importance on search and correction of textual information. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 297–305. Springer.

Joshi, A., Mishra, A., Senthamilselvan, N., and Bhattacharyya, P. (2014). Measuring sentiment annotation complexity of text. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 36–41.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Kobayashi, R. and Lambiotte, R. (2016). TiDeH: Time-dependent Hawkes process for predicting retweet dynamics. In *Tenth International AAAI Conference on Web and Social Media*.

Kouloumpis, E., Wilson, T., and Moore, J. (2011). Twitter sentiment analysis: The good the bad and the OMG! In *Fifth International AAAI conference on weblogs and social media*.

Krippendorff, K. (2011). Computing Krippendorff's alpha-reliability. *Computing*, 1.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Kusner, M., Sun, Y., Kolkin, N., and Weinberger, K. (2015). From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966.

Labille, K., Gauch, S., and Alfarhood, S. (2017). Creating domain-specific sentiment lexicons via text mining. In *Proc. Workshop Issues Sentiment Discovery Opinion Mining (WISDOM)*.

Lai, S., Xu, L., Liu, K., and Zhao, J. (2015). Recurrent convolutional neural networks for text classification. In *Twenty-ninth AAAI conference on artificial intelligence*.

Lasersohn, P. (2005). Context dependence, disagreement, and predicates of personal taste. *Linguistics and philosophy*, 28(6):643–686.

Laub, P. J., Taimre, T., and Pollett, P. K. (2015). Hawkes processes. *arXiv preprint arXiv:1507.02822*.

Lawrence, S., Giles, C. L., Tsoi, A. C., and Back, A. D. (1997). Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1):98–113.

Li, G., Chang, K., Hoi, S. C., Liu, W., and Jain, R. (2011). Collaborative online learning of user generated content. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 285–290. ACM.

Li, G., Hoi, S. C., Chang, K., and Jain, R. (2010). Micro-blogging sentiment detection by collaborative online learning. In *2010 IEEE International Conference on Data Mining*, pages 893–898. IEEE.

Lim, M. Y. (2012). Memory models for intelligent social companions. In *Human-Computer Interaction: The Agency Perspective*, pages 241–262. Springer.

Linell, P. (2009). *Rethinking language, mind, and world dialogically*. IAP.

Liniger, T. J. (2009). *Multivariate hawkes processes*. PhD thesis, ETH Zurich.

Lipski, J. (1978). Code-switching and the problem of bilingual competence. *Aspects of bilingualism*, 250:264.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Liu, B. (2015). *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press.

Liu, B., Blasch, E., Chen, Y., Shen, D., and Chen, G. (2013). Scalable sentiment classification for big data analysis using naive Bayes classifier. In *2013 IEEE international conference on big data*, pages 99–104. IEEE.

Liu, P., Qiu, X., and Huang, X. (2016). Recurrent neural network for text classification with multi-task learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2873–2879. AAAI Press.

Liu, X., He, P., Chen, W., and Gao, J. (2019). Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.

Lynn, V., Son, Y., Kulkarni, V., Balasubramanian, N., and Schwartz, H. A. (2017). Human centered NLP with user-factor adaptation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1146–1155. Association for Computational Linguistics (ACL).

Markovikj, D., Gievska, S., Kosinski, M., and Stillwell, D. (2013). Mining Facebook data for predictive personality modeling. In *Proceedings of the 7th international AAAI conference on Weblogs and Social Media (ICWSM 2013), Boston, MA, USA*, pages 23–26.

McDougall, W. (1919). *An introduction to social psychology.* Methuen & Co. Ltd. London.

Mcnamee, P. and Mayfield, J. (2004). Character n-gram tokenization for european language text retrieval. *Information retrieval*, 7(1-2):73–97.

Meena, A. and Prabhakar, T. (2007). Sentence level sentiment analysis in the presence of conjuncts using linguistic analysis. In *European Conference on Information Retrieval*, pages 573–580. Springer.

Mesnil, G., Mikolov, T., Ranzato, M., and Bengio, Y. (2014). Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews. *arXiv preprint arXiv:1412.5335*.

Michel, P. and Neubig, G. (2018). Extreme adaptation for personalized neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 312–318.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mohammad, S. (2016a). A practical guide to sentiment annotation: Challenges and solutions. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 174–179.

Mohammad, S. M. (2016b). Sentiment analysis: Detecting valence, emotions, and other affectual states from text. In *Emotion measurement*, pages 201–237. Elsevier.

Mohler, G. O., Short, M. B., Brantingham, P. J., Schoenberg, F. P., and Tita, G. E. (2011). Self-exciting point process modeling of crime. *Journal of the American Statistical Association*, 106(493):100–108.

Moraes, R., Valiati, J. F., and Neto, W. P. G. (2013). Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Systems with Applications*, 40(2):621–633.

Morency, L.-P., Mihalcea, R., and Doshi, P. (2011). Towards multimodal sentiment analysis: Harvesting opinions from the web. In *Proceedings of the 13th international conference on multimodal interfaces*, pages 169–176. ACM.

Mullen, T. and Collier, N. (2004). Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 412–418.

Munezero, M. D., Montero, C. S., Sutinen, E., and Pajunen, J. (2014). Are they different? affect, feeling, emotion, sentiment, and opinion detection in text. *IEEE transactions on affective computing*, 5(2):101–111.

Muysken, P., Díaz, C. P., Muysken, P. C., et al. (2000). *Bilingual speech: A typology of code-mixing*, volume 11. Cambridge University Press.

Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F., and Stoyanov, V. (2016). SemEval-2016 task 4: Sentiment analysis in Twitter. In *Proceedings of the 10th international workshop on semantic evaluation (semeval-2016)*, pages 1–18.

Nasukawa, T. and Yi, J. (2003). Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77. ACM.

Neil, D., Pfeiffer, M., and Liu, S.-C. (2016). Phased lstm: Accelerating recurrent network training for long or event-based sequences. In *Advances in neural information processing systems*, pages 3882–3890.

Ng, H. T., Goh, W. B., and Low, K. L. (1997). Feature selection, perceptron learning, and a usability case study for text categorization. In *SIGIR*, volume 97, pages 67–73.

Nguyen, L. T., Wu, P., Chan, W., Peng, W., and Zhang, Y. (2012). Predicting collective sentiment dynamics from time-series social media. In *Proceedings of the first international workshop on issues of sentiment discovery and opinion mining*, page 6. ACM.

Nowak, A., Szamrej, J., and Latané, B. (1990). From private attitude to public opinion: A dynamic theory of social impact. *Psychological review*, 97(3):362.

Ogata, Y. (1998). Space-time point-process models for earthquake occurrences. *Annals of the Institute of Statistical Mathematics*, 50(2):379–402.

Oliveira, N., Cortez, P., and Areal, N. (2016). Stock market sentiment lexicon acquisition using microblogging data and statistical measures. *Decision Support Systems*, 85:62–73.

Paiva, A., Leite, I., and Ribeiro, T. (2014). Emotion modeling for social robots. *The Oxford handbook of affective computing*, pages 296–308.

Pak, A. and Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, pages 1320–1326.

Pang, B. and Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, page 271. Association for Computational Linguistics.

Pang, B., Lee, L., et al. (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Pele, O. and Werman, M. (2009). Fast and robust earth mover's distances. In *2009 IEEE 12th International Conference on Computer Vision*, pages 460–467. IEEE.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Plank, B. (2016). What to do about non-standard (or non-canonical) language in NLP. *Bochumer Linguistische Arbeitsberichte*, page 13.

Plank, B., Alonso, H. M., and Søgaard, A. (2015). Non-canonical language is not harder to annotate than canonical language. In *Proceedings of The 9th Linguistic Annotation Workshop*, pages 148–151.

Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., and Manandhar, S. (2014). SemEval-2014 task 4: Aspect based sentiment analysis. *Proceedings of SemEval*, pages 27–35.

Poria, S., Cambria, E., Bajpai, R., and Hussain, A. (2017). A review of affective computing: From unimodal analysis to multimodal fusion. *Information Fusion*, 37:98–125.

Poria, S., Cambria, E., and Gelbukh, A. (2016a). Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, 108:42–49.

Poria, S., Cambria, E., Howard, N., Huang, G.-B., and Hussain, A. (2016b). Fusing audio, visual and textual clues for sentiment analysis from multimodal content. *Neurocomputing*, 174:50–59.

Poria, S., Cambria, E., Winterstein, G., and Huang, G.-B. (2014). Sentic patterns: Dependency-based rules for concept-level sentiment analysis. *Knowledge-Based Systems*, 69:45–63.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf.*

Ravi, K. and Ravi, V. (2015). A survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowledge-Based Systems*, 89:14–46.

Reitan, J., Faret, J., Gambäck, B., and Bungum, L. (2015). Negation scope detection for Twitter sentiment analysis. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 99–108.

Reiter, E. and Sripada, S. (2002). Human variation and lexical choice. *Computational Linguistics*, 28(4):545–553.

Ren, F. and Wu, Y. (2013). Predicting user-topic opinions in Twitter with social and topical context. *IEEE Transactions on affective computing*, 4(4):412–424.

Rosa, R. L., Rodríguez, D. Z., and Bressan, G. (2015). Music recommendation system based on user's sentiments extracted from social networks. In *2015 IEEE International Conference on Consumer Electronics (ICCE)*, pages 383–384. IEEE.

Russell, J. A. (1980). A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161.

Russell, J. A. and Barrett, L. F. (1999). Core affect, prototypical emotional episodes, and other things called emotion: dissecting the elephant. *Journal of personality and social psychology*, 76(5):805.

Sacks, H., Schegloff, E. A., and Jefferson, G. (1978). A simplest systematics for the organization of turn-taking for conversation. In *Studies in the organization of conversational interaction*, pages 7–55. Elsevier.

Sakaki, S., Miura, Y., Ma, X., Hattori, K., and Ohkuma, T. (2014). Twitter user gender inference using combined analysis of text and image processing. In *Proceedings of the Third Workshop on Vision and Language*, pages 54–61.

Samek, W., Binder, A., Montavon, G., Lapuschkin, S., and Müller, K.-R. (2016). Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673.

Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260. ACM.

Schimmack, U. and Grob, A. (2000). Dimensional models of core affect: A quantitative comparison by means of structural equation modeling. *European Journal of Personality*, 14(4):325–345.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61:85–117.

Schommer, C., Kampas, D., and Bersan, R. (2013). A prospect on how to find the polarity of a financial news by keeping an objective standpoint. *Proceedings ICAART 2013*.

Schouten, K. and Frasincar, F. (2015). Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):813–830.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

Severyn, A. and Moschitti, A. (2015). Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 959–962. ACM.

Shen, T., Zhou, T., Long, G., Jiang, J., Pan, S., and Zhang, C. (2018). DiSAN: directional self-attention network for RNN/CNN-free language understanding. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Shouse, E. (2005). Feeling, emotion, affect. *M/c journal*, 8(6):26.

Singh, V. K., Piryani, R., Uddin, A., and Waila, P. (2013). Sentiment analysis of movie reviews: A new feature-based heuristic for aspect-level sentiment classification. In *2013 International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s)*, pages 712–717. IEEE.

Song, K., Chen, L., Gao, W., Feng, S., Wang, D., and Zhang, C. (2016). PerSentiment: a personalized sentiment classification system for microblog users. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 255–258. International World Wide Web Conferences Steering Committee.

Song, K., Feng, S., Gao, W., Wang, D., Yu, G., and Wong, K.-F. (2015). Personalized sentiment classification based on latent individuality of microblog users. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Stets, J. E. (2006). Emotions and sentiments. In *Handbook of social psychology*, pages 309–335. Springer.

Stieglitz, S. and Dang-Xuan, L. (2013). Emotions and information diffusion in social media?sentiment of microblogs and sharing behavior. *Journal of management information systems*, 29(4):217–248.

Surowiecki, J. (2005). *The wisdom of crowds.* Anchor.

Taboada, M. (2016). Sentiment analysis: an overview from linguistics. *Annual Review of Linguistics*, 2:325–347.

Tan, C., Lee, L., Tang, J., Jiang, L., Zhou, M., and Li, P. (2011). User-level sentiment analysis incorporating social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1397–1405. ACM.

Tan, S., Cheng, X., Wang, Y., and Xu, H. (2009). Adapting naive Bayes to domain adaptation for sentiment analysis. In *European Conference on Information Retrieval*, pages 337–349. Springer.

Tang, D., Qin, B., and Liu, T. (2015a). Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1422–1432.

Tang, D., Qin, B., and Liu, T. (2015b). Learning semantic representations of users and products for document level sentiment classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1014–1023.

Tang, D., Qin, B., and Liu, T. (2016). Aspect level sentiment classification with deep memory network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 214–224.

Tang, D., Qin, B., Liu, T., and Yang, Y. (2015c). User modeling with neural network for review rating prediction. In *Twenty-Fourth International Joint Conference on Artificial Intelligence.*

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., and Qin, B. (2014). Learning sentiment-specific word embedding for Twitter sentiment classification. In *ACL (1)*, pages 1555–1565.

Toh, Z. and Wang, W. (2014). DLIREC: Aspect term extraction and term polarity classification system. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 235–240.

Tsiourti, C., Ben-Moussa, M., Quintas, J., Loke, B., Jochem, I., Albuquerque Lopes, J., et al. (2016). A virtual assistive companion for older adults: design implications for a real-world application. In *SAI Intelligent Systems Conference*, pages 556–566.

Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Vo, D. T. and Zhang, Y. (2016). Don?t count, predict! an automatic approach to learning sentiment lexicons for short text. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 219–224.

Wang, G., Sun, J., Ma, J., Xu, K., and Gu, J. (2014). Sentiment classification: The contribution of ensemble learning. *Decision support systems*, 57:77–93.

Wang, H., Can, D., Kazemzadeh, A., Bar, F., and Narayanan, S. (2012). A system for real-time Twitter sentiment analysis of 2012 U.S. presidential election cycle. In *Proceedings of the ACL 2012 system demonstrations*, pages 115–120. Association for Computational Linguistics.

Wang, J.-H., Liu, T.-W., Luo, X., and Wang, L. (2018a). An LSTM approach to short text sentiment classification with word embeddings. In *Proceedings of the 30th Conference on Computational Linguistics and Speech Processing (ROCLING 2018)*, pages 214–223.

Wang, W., Feng, S., Gao, W., Wang, D., and Zhang, Y. (2018b). Personalized microblog sentiment classification via adversarial cross-lingual multi-task learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 338–348.

Wang, X., Zhang, C., Ji, Y., Sun, L., Wu, L., and Bao, Z. (2013). A depression detection model based on sentiment analysis in micro-blog social network. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 201–213. Springer.

Wang, Y., Huang, M., Zhao, L., et al. (2016). Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615.

Whitelaw, C., Garg, N., and Argamon, S. (2005). Using appraisal groups for sentiment analysis. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 625–631. ACM.

Wiebe, J., Wilson, T., and Bell, M. (2001). Identifying collocations for recognizing opinions. In *Proceedings of the ACL-01 Workshop on Collocation: Computational Extraction, Analysis, and Exploitation*, pages 24–31.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing.*

Wintner, S., Mirkin, S., Specia, L., Rabinovich, E., and Patel, R. N. (2017). Personalized machine translation: Preserving original author traits. In *EACL (1)*, pages 1074–1084.

Wu, F. and Huang, Y. (2016). Personalized microblog sentiment classification via multi-task learning. In *Thirtieth AAAI Conference on Artificial Intelligence.*

Wu, L., Morstatter, F., and Liu, H. (2018a). SlangSD: building, expanding and using a sentiment dictionary of slang words for short-text sentiment classification. *Language Resources and Evaluation*, 52(3):839–852.

Wu, Z., Dai, X.-Y., Yin, C., Huang, S., and Chen, J. (2018b). Improving review representations with user attention and product attention for sentiment classification. In *Thirty-Second AAAI Conference on Artificial Intelligence.*

Xiao, S., Yan, J., Yang, X., Zha, H., and Chu, S. M. (2017). Modeling the intensity function of point process via recurrent neural networks. In *Thirty-First AAAI Conference on Artificial Intelligence.*

Xu, J., Chen, D., Qiu, X., and Huang, X. (2016). Cached long short-term memory neural networks for document-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1660–1669.

Xue, W. and Li, T. (2018). Aspect based sentiment analysis with gated convolutional networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2514–2523.

Yang, D., Zhang, D., Yu, Z., and Wang, Z. (2013). A sentiment-enhanced personalized location recommendation system. In *Proceedings of the 24th ACM conference on hypertext and social media*, pages 119–128. ACM.

Yang, Y. and Eisenstein, J. (2015). Putting things in context: Community-specific embedding projections for sentiment analysis. *arXiv preprint arXiv:1511.06052*, 4(3).

Yang, Y. and Eisenstein, J. (2017). Overcoming language variation in sentiment analysis with social attention. *Transactions of the Association for Computational Linguistics*, 5:295–307.

Yang, Y. and Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *Icml*, volume 97, page 35.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Yao, K., Zweig, G., Hwang, M.-Y., Shi, Y., and Yu, D. (2013). Recurrent neural networks for language understanding. In *Interspeech*, pages 2524–2528.

Yu, J., Zha, Z.-J., Wang, M., and Chua, T.-S. (2011). Aspect ranking: identifying important product aspects from online consumer reviews. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1496–1505. Association for Computational Linguistics.

Yu, X., Wei, X., and Lin, X. (2010). Algorithms of BBS opinion leader mining based on sentiment analysis. In *International Conference on Web Information Systems and Mining*, pages 360–369. Springer.

Zaremba, W. and Sutskever, I. (2014). Learning to execute. *arXiv preprint arXiv:1410.4615*.

Zhang, L., Wang, S., and Liu, B. (2018). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253.

Zhao, K., Cong, G., Yuan, Q., and Zhu, K. Q. (2015a). SAR: A sentiment-aspect-region model for user preference analysis in geo-tagged reviews. In *2015 IEEE 31st International Conference on Data Engineering*, pages 675–686. IEEE.

Zhao, Q., Erdogdu, M. A., He, H. Y., Rajaraman, A., and Leskovec, J. (2015b). SEISMIC: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1513–1522. ACM.

Zhao, Z., Lu, H., Cai, D., He, X., and Zhuang, Y. (2017). Microblog sentiment classification via recurrent random walk network learning. In *IJCAI*, volume 17, pages 3532–3538.

Zhou, Q., Xu, Z., and Yen, N. Y. (2019). User sentiment analysis based on social network information and its application in consumer reconstruction intention. *Computers in Human Behavior*, 100:177–183.

Zhu, L. (2013). *Nonlinear Hawkes Processes*. PhD thesis, Citeseer.