
THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

# STRUCTURED SENTIMENT ANALYSIS IN SOCIAL MEDIA

Abdulqader M. Almars

Master of Computer Science

*A thesis submitted for the degree of Doctor of Philosophy at The University of Queensland in 2019*

*School of Information Technology and Electrical Engineering*

# Abstract

In recent years, social media platforms, such as Twitter and Webio, have become popular sources of information on the web. These platforms contain a wealth of valuable information about user opinions, user interests, events and more. People typically use these platforms to discuss different topics, share their opinions about them and engage in question-and-answer sessions. For example, regarding smartphones, users might discuss the main aspects of a smartphone, such as the overall design, battery capacity, screen size and camera. The natural hierarchical structure of those concepts is often hidden in social media. Discovering the hidden structure can helps users understand people' preference to a certain topic at different levels of granularity, and show the reasons why they prefer this topic. Over the past decade, research on hierarchical topic models has shown considerable progress. However, these studies may not always be directly applicable to social media due to the shortness and the shallow meaning of social media messages.

There are three major challenges when dealing with social media texts. Firstly, compared with traditionally long texts, social media texts suffer from sparsity, and this issue may result in an incomprehensible and incorrect concept hierarchy. Secondly, social media contains useful information such as social opinions and information about users. Most existing methods perform a flat sentiment analysis on each extracted aspects independently, and ignore the concept hierarchy. In fact, we need to make the sentiment analysis fine-grained in order to simultaneously extract the aspects and summarise people' opinions on those discovered aspects. Thirdly, the current models only discover the concept hierarchy ignoring the community structure of users. Maintaining the consistency of user's interest on several communities according to various topics and sentiment information is a challenging problem.

In this thesis, the limitations of the existing work are addressed and effective solutions are proposed. First, in order to discover the hierarchical structure of social media content, a novel approach called the context coherence model (CCM) is proposed. It recursively top

down: (1) organizes the concepts discussed by users in social media texts; and (2) identifies the hierarchical relations among concepts. In the CCM, a new measurement called context coherence is introduced that analyses words in social media texts and determines the similarities among them. Then, the hierarchical relationship between words is determined by recursively partitioning the whole corpus into smaller parts according to the similarity results. Finally, a merging operation is performed to find similar words, group them under the same topic and remove duplicated topics. The approach is evaluated on two real-world data sets. The experiments show that the proposed approach can effectively reveal the hidden structure in social media.

Opinions are now reflected in social media on a wide range of topics: trends in pop music, fashion, politics, financial markets, natural disaster responses, sales of products and services, etc. For example, companies may want to understand the feelings of consumers towards their products or services at different levels of granularity. Therefore, the problem of hierarchical extraction is extended to consider sentiment analysis. A structured sentiment analysis (SSA) approach is proposed that summarizes users' feelings towards those concepts discovered in the tree. Given users' messages, the hierarchical clustering method is proposed to detect the top aspects interest users, based on their messages, and attaches users' attitudes to them. To perform sentiment analysis, a top-down, lexicon-based approach was designed to identify the polarity of top aspects of a topic. Finally, a simple summarization method was developed to answer questions such as: (1) What is the overall popularity of the product or service? (2) Why do people like or dislike the product or service? and (3) What are the most favourable and unfavourable aspects?

Third, modelling the interests of users is particularly important and can help organizations to understand and analyse users' behaviours and locate influential users at different granularity levels using their sentiment information. A probabilistic model, namely, the hierarchical user sentiment/topic model (HUSTM), is proposed to discover the hidden structure of topics and users while performing sentiment analysis in a unified way. In HUSTM, users who share the same topic and opinion are grouped within the same community. In this approach, the entire structure is a tree where each node is decomposed into a topic/sentiment node and a user-sentiment node. The topic/sentiment node is, in turn, a mixed distribution of words, while the user-sentiment node is a mixed distribution of users. To experimentally demonstrate the advantages of the approach, three real-world data sets were used. The results showed that, compared to other state-of-the-art techniques, the HUSTM approach can more successfully capture users' interests.

# Declaration by author

This thesis is composed of my original work, and contains no material previously published or written by another person except where due reference has been made in the text. I have clearly stated the contribution by others to jointly-authored works that I have included in my thesis.

I have clearly stated the contribution of others to my thesis as a whole, including statistical assistance, survey design, data analysis, significant technical procedures, professional editorial advice, financial support and any other original research work used or reported in my thesis. The content of my thesis is the result of work I have carried out since the commencement of my higher degree by research candidature and does not include a substantial part of work that has been submitted to qualify for the award of any other degree or diploma in any university or other tertiary institution. I have clearly stated which parts of my thesis, if any, have been submitted to qualify for another award.

I acknowledge that an electronic copy of my thesis must be lodged with the University Library and, subject to the policy and procedures of The University of Queensland, the thesis be made available for research and study in accordance with the Copyright Act 1968 unless a period of embargo has been approved by the Dean of the Graduate School.

I acknowledge that copyright of all material contained in my thesis resides with the copyright holder(s) of that material. Where appropriate I have obtained copyright permission from the copyright holder to reproduce material in this thesis and have sought permission from co-authors for any jointly authored works included in the thesis.

Abdulqader Almars

# Publications included in this thesis

**Abdulqader M. Almars**, Xue Li and Xin Zhao. Modeling user attitudes using hierarchical sentiment-topic model. Data and Knowledge Engineering, 2019.

| **Contributor** | **Statement of contribution** |
|---|---|
| Abdulqader M. Almars | Conception and design (80%)<br>Analysis and interpretation (80%)<br>Drafting and production (80%) |
| Xue Li | Conception and design (10%)<br>Analysis and interpretation (10%)<br>Drafting and production (10%) |
| Xin Zho | Conception and design (10%)<br>Analysis and interpretation (10%)<br>Drafting and production (10%) |

- **Abdulqader M. Almars**, Xue Li, Xin Zhao, I. A. Ibrahim, Weiwei Yuan, Bohan Li. Structured Sentiment Analysis. In International Conference on Advanced Data Mining and Applications ADMA 2017, Singapore, November 05 - 06, 2017, pages 695-707.

| **Contributor** | **Statement of contribution** |
|---|---|
| Abdulqader Almars | Conception and design (70%)<br>Analysis and interpretation (70%)<br>Drafting and production (65%) |
| Xue Li | Conception and design (10%)<br>Analysis and interpretation (5%)<br>Drafting and production (10%) |
| Xin Zhao | Conception and design (5%)<br>Analysis and interpretation (10%)<br>Drafting and production (10%) |
| I. A. Ibrahim | Conception and design (5%)<br>Analysis and interpretation (5%)<br>Drafting and production (5%) |
| Weiwei Yuan | Conception and design (5%)<br>Analysis and interpretation (5%)<br>Drafting and production (5%) |
| Bohan Li | Conception and design (5%)<br>Analysis and interpretation (5%)<br>Drafting and production (5%) |

- **Almars, Abdulqader** & Li, Xue & A. Ibrahim, Ibrahim & Zhao, Xin. (2018). Learning Concept Hierarchy from Short Texts Using Context Coherence: 19th International Conference, WISE 2018, Dubai, United Arab Emirates, November 12-15, 2018.

| **Contributor** | **Statement of contribution** |
|---|---|
| Abdulqader Almars | Conception and design (80%)<br>Analysis and interpretation (75%)<br>Drafting and production (70%) |
| Xue Li | Conception and design (10%)<br>Analysis and interpretation (5%)<br>Drafting and production (10%) |
| I. A. Ibrahim | Conception and design (5%)<br>Analysis and interpretation (10%)<br>Drafting and production (10%) |
| Xin Zhao | Conception and design (5%)<br>Analysis and interpretation (10%)<br>Drafting and production (10%) |

- **M. Almars, Abdulqader** A. Ibrahim, Ibrahim Zhao, Xin AL Maskari, Sanad. (2018). Evaluation Methods of Hierarchical Models: 14th International Conference, ADMA 2018, Nanjing, China, November 16-18, 2018.

| **Contributor** | **Statement of contribution** |
|---|---|
| Abdulqader Almars | Conception and design (80%)<br>Analysis and interpretation (80%)<br>Drafting and production (75%) |
| I. A. Ibrahim | Conception and design (10%)<br>Analysis and interpretation (7%)<br>Drafting and production (10%) |
| Xin Zhao | Conception and design (5%)<br>Analysis and interpretation (10%)<br>Drafting and production (10%) |
| AL Maskari, Sanad | Conception and design (5%)<br>Analysis and interpretation (3%)<br>Drafting and production (5%) |

## Submitted manuscripts included in this thesis

No manuscripts submitted for publication.

# Other Publications during candidature

### Conference Papers

- I. A. Ibrahim, **Abdulqader M. Almars**, Suresh Pokharel, Xin Zhao, and Xue Li. Interesting recommendations based on hierarchical visualizations of medical data. In Big Data Analytics for Social Computing, PAKDD'18 Workshop, Melbourne, Australia June 01-03, 2018.

- Sanad Al Maskari, I. A. Ibrahim, Xue Li, Eimad Abusham and **Abdulqader M. Almars**. Feature Extraction for Smart Sensing Using Multi-Perspectives Transformation. In The Australasian Database Conference ADC 2018 , Gold Coast, Australia May 23-25, 2018, Databases Theory and Applications, pages 236-248.

- Abdullah Albarrak, Sanad Al-Maskari, Ibrahim A. Ibrahim and **Abdulqader M. Almars**. Efficiently Mining Constrained Subsequence Patterns: 14th International Conference, ADMA 2018, Nanjing, China, November 16-18, 2018.

## Contributions by others to the thesis

My principle advisor, Prof Xue Li has provided very helpful insight towards the research problems presented in this thesis. He also helped with both reviewing and editing the published papers. Dr Xin Zho also assisted me by providing suggestions and feedback on problems formulation and solutions in this thesis. He also assisted with both the refinement of the idea and the pre-submission edition.

## Statement of parts of the thesis submitted to qualify for the award of another degree

No works submitted towards another degree have been included in this thesis.

## Research Involving Human or Animal Subjects

No animal or human subjects were involved in this research.

# Acknowledgements

In the beginning, I would like to express my deepest gratitude to my principal advisor, Prof Xue Li for his patience and support during my PhD journey. I have learned how to ask good questions, conduct concrete research and the value of establishing strong work ethics. I will always feel grateful and proud that I was one of his students.

I would like to thank my associate advisor, Dr Xin Zhao, for his suggestions and persistent guidance with my research problems. My genuine thanks go to my advisory committee member, Dr Helen Huang for the insightful feedback and support during my PhD milestones.

I thank my parents for their love and support. I thank my father Mohammed Almars for his love and advice. All my love to my mother, and thank you so much for everything you have done for me.

I would like to thank my dear wife, Fatimah Shaheen, who has been a real source of strength and happiness all the time. She stood by my side, supporting me every day during my PhD program. Thank you for being a wonderful mother for our little children, Ghazel and Mohammed.

My deep thanks and love to my colleagues in our DAS group. Special Thanks to Ibrahim, Jingwei Ma, Rocky and Suresh for helping me with my research.

Lastly, I would like to formally thank my sponsor, Taibah University in Madinah, for providing the continuous support that made this journey possible.

# Financial support

This research was supported by Saudi culture affairs and mission sector (Scholarships Program)and Taibah University.

## Keywords

Hierarchical Model,
Sentiment Analysis,
Hierarchical User-Sentiment Topic Model,
Structured Sentiment Analysis

## Australian and New Zealand Standard Research Classifications (ANZSRC)

ANZSRC code: 080107, Natural Language Processing, 100%

## Fields of Research (FoR) Classification

FoR code: 0801, Artificial Intelligence and Image Processing, 100%

# Contents

# List of Figures

# List of Tables

# List of Algorithms

CHAPTER 1

# Introduction

Social media platforms such as Twitter generate a large quantity of messages, carrying information covering a wide range of topics. Because people often express themselves spontaneously on social media, the information discussed can be associated with opinions on a variety of topics and with a number of users. For this reason, individuals and organizations may want to understand the feelings of users towards a particular topic to make informed decisions. The goal of this thesis is to understand: **What** topics that people care about, **Why** people like those topics and **Who** are the people interested in those topics. In this chapter, we give a brief introduction of the research in this thesis, including the background, motivations, research goals, challenges, contributions to the existing literature and the organization of the thesis.

## 1.1 Motivation

**Example 1: Latent structures in social media.** Social media sites such as Twitter and Webio have become the most popular methods of communication for the current generation. On the micro-blogging site Twitter, users can post texts of up to 280 characters on their profile pages. They usually share their experiences and express opinions on different topics, such as trends in pop music, fashion, politics, financial markets, natural disaster responses and sales of products and services. Figure 1.1 displays an example of the use of social media in spreading information about different topics. In smartphone-related tweets, users discuss and express opinions on the main topic (concept) of a smartphone, such as the overall design, battery capacity, screen size and camera. However, the natural hierarchical structure of those concepts and sentiment polarity are often hidden. This means the user needs to read a tweet-by-tweet conversation in order to know and understand other users' feelings towards


**Figure 1.1:** The use of social media in spreading information about different aspects.

the product. Additionally, comparing two or more products based on different aspects is a difficult task. This may require users to spend more time and effort to analyse and understand the similarities and differences between different products' features. Figure 1.2 shows an example of users discussing several features of smartphones.

**Example 2: Community structure discovery in social media.** In a microblogging platform such as Twitter, the users can express their opinions about restaurants, and can comment on different aspects, such as cleanliness, food, service and location of the restaurants. Here, *community* can be defined as groups of users that have similar opinions and commonly discussed topics with each other, and topics can be the popular themes within the community. It would be useful to automatically identify the communities and organise those communities hierarchically. By doing this, organisations could understand and analyse user' behaviour, locate influential users at different granularity levels using their sentiment information. Moreover, community structure discovery may help individuals to identify specific groups of users who create and spread rumours in social media.

**Example 3: Limitations of current methods.** Hierarchical topic models have been previously proposed to effectively extract hidden structures from traditional texts [18, 47, 45].


**Figure 1.2:** The use of Twitter platform in spreading information about several features of smartphones.

However, applying these models to social media may result in less effective performance due to the sparsity of text. In Figure 1.3, we present part of a concept hierarchy created by a hierarchical aspect-sentiment model (HASM) [45] from a smartphone data set to show the problems of incoherent concepts and an unreasonable structure. In an incoherent concept, there are some words that are not semantically related to the other words. For example, the fourth concept created by the HASM contains irrelevant words (e.g., student and boyfriend); all words on the tree should be relevant and semantically related. Another problem is that the tree shows some duplicate concepts (e.g., the children of the second and third concepts are the same as the parent concept).

Generally speaking, discovering the latent structure of specific aspects and their co-sponsoring sentiments are important from two points of view, individuals and business organizations. From the individual's viewpoint, a sentiment tree organizes aspects from general to specific. Therefore, it allows an individual to find people's attitudes and opinions about various aspects represented by the tree at different granularities. For example, someone may be interested in people's opinions about a product in general, while others may look for people's opinions on specific aspects, such as the quality of a smartphone's camera. From the point view of organizations, uncovering the hidden structure can allow them to trace public opinion on features of a product or service, and it provides them with important information to help them improve future designs and strategies.

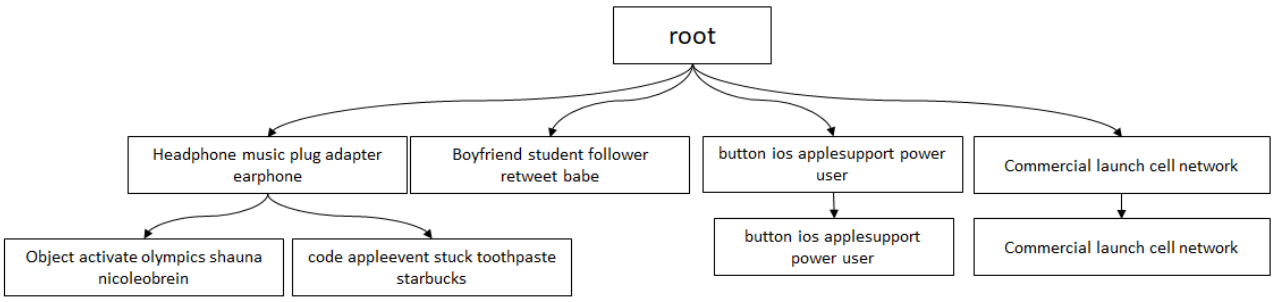



**Figure 1.3:** A part of the concept hierarchy learned from smartphone dataset using HASM Model.

## 1.2 Background

**User (Author) Community Analysis.** One of the most popular areas in data mining is user (author) community analysis. A number of models that merge the author's information into the topic model have been proposed[83, 53, 7, 91], such as the author topic model [82]. It discovers underlying topics conditioned on the authors' information, and each author is associated with a probability distribution over topics. The community-author-recipient-topic (CART) model [74] extracts communities by using the semantic content of a social network, and was one of the first attempts to integrate social links and content information for the purpose of community discovery. In comparison to these works that capture the authors' interests as a multinomial distribution over topics, the Author-Interest-Topic (AIT) model [43] introduced an additional layer between the author and topic layers that instead captured the authors' interests as a probability distribution of documents. All these models perform well for social media analysis, but they neglect the natural hierarchical structure of topics and community, and the current community analysis methods ignore the sentiment information. However, users in the same community can be further decomposed into sub-communities according to their opinions.

**Sentiment Analysis.** Sentiment analysis is also known as opinion mining [11]. *Sentiment analysis* is defined as the process of identifying the opinions people express in a piece of text as positive, negative or neutral. Several sentiment/topic models have been proposed to uncover topics with different sentiments. Lin et al. [55] introduced a flat joint sentiment/topic (JST) model, based on latent Dirichlet allocation (LDA). In JST, each sentiment polarity is associated with a mixture of topics, and all words are distributed over this mixture. In this study [42], the aspect and sentiment unification model (ASUM) was proposed as a sentence-level model that assigns all words in a sentence to the same polarity. The topic/sentiment mixture (TSM) model [63] aimed at explicitly modelling the sentiment as a language model

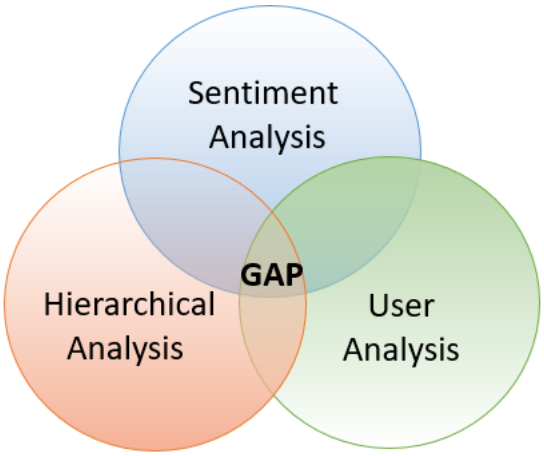



**Figure 1.4:** The Gap in the current work.

separated from topics. Indeed, TSM assumed the topic and sentiment were sampled independently, which meant the words were drawn from either topics or sentiments. In [108], the authors proposed a user sentiment/topic model (USTM) that integrated user opinions into an LDA model. All of the mentioned models extracted topics and users as a flat mixture of topics.

**Hierarchical Analysis**. Hierarchical analysis is the process of analysing a collection of documents to uncover the hidden structure they contain. The output is a hierarchy of topics (concepts) where each topic in the tree is a coherent theme, represented by either a single word of a set of words. In recent years, hierarchical topic model research has focused on identifying a hierarchical tree of topics within documents [71, 66, 107, 92, 100]. Blei et al. [15] introduced the nested Chinese restaurant process (nCRP) to hierarchically discover structures within data. The depth and number of child topics in this model are manually specified. Kim et al. [45] suggested a new model based on the nCRP, namely, the recursive Chinese restaurant process (rCRP). In the rCRP model, the nCRP model is extended to create a hierarchical tree where each document has a distribution over all the topics in the tree. Kim et al. [47] proposed a novel approach through the HASM, which applied a Bayesian non-parametric model to infer and learn the structure and sentiment in online reviews. The above-mentioned methods only extracted their topic hierarchies from the topics, without considering sentiment information or users' interests in those topics. Another disadvantage with those models is that they were proposed to be effective with larger amounts of text; applying them to social media texts can lead to an incomplete or flat tree. Figure 1.4 summarises the gap in the existing literature.

## 1.3 Research Goals

In this section, we will discuss the research goals of this thesis. At the highest level, the aim was to develop an effective hierarchical model for use with social network blog services (e.g., Twitter). The goals of this thesis were as follows:

- The main goal was to understand **what** topics people discuss in a collection of documents. More specifically, the goal was to design a framework and develop effective solutions for the discovery of hierarchical structures from social media. The approach had to be able to process large collections of short and noisy messages. More specifically, the approach had to extract different topics (aspects) and simultaneously identify the hierarchical relationships among them.

- The second goal was to design a hierarchical framework to understand **what** topics that interest people and **why** people like or dislike them. The approach focuses on identifying attitudes at various levels of granularity. The approach needs to be capable of doing a fine-grained sentiment analysis. In other words, the sentiment polarity of each aspect in the tree is performed hierarchically by including the sentiment polarity for the aspect itself and its children in the hierarchical tree.

- The third goal is to develop a framework for understanding **who** the people are who are interested in those topics discussed on social media. The approach must model users' interests and opinions on different topics in the tree simultaneously. In other words, the approach focuses on hierarchically grouping users who share the same discussed topic and opinion within the same community.

## 1.4 Research Challenges

Traditional approaches in hierarchical topic research are not designed to deal with social media texts. The frequency of words in short messages play a less discriminative role compared to traditional documents like reviews. As a result, directly applying those hierarchical models to social media will produce an unbalanced or flat tree. The challenges of this thesis were the following:

- **Topic (Concept) Extraction and Semantic Relationship Identification.** Social media messages on platforms like Twitter usually contain noise and advertisements. First,

discovering the individual aspects of a product or topic under discussion was a challenging issue. Second, with a large number of concepts discussed in very short messages, the shape of the tree was unknown (e.g., depth and width) in advance. Third, the length and shallow meaning of short messages make the semantic relationship analysis between words a difficult task.

- **Sentiment Analysis.** The current sentiment/topic models are flat models, which means they neglect the natural hierarchy of the individual aspects and the sentiment polarity. For sentiment analysis, the challenges were that we need to make sentiment analysis fine-grained in order to simultaneously discover the *hot aspects* and identify their polarities.

- **User Community Analysis.** Modelling user's interests from stoical media was a challenging task for two reasons. The first is consistency, meaning that users in the same community should be similar with respect to the topic being discussed and the opinions that they hold. Second, the hierarchical clustering in a unified way of information about topic, user and sentiment is difficult because of the sparsity and shortness of these brief messages. Further, the traditional hierarchical topic modelling method does not take into account modelling the users' interests. Therefore, we studied the problem of modelling users' interests across various topics and sentiment polarities on social media.

## 1.5 Main Contributions

Based on the research problems discussed and the challenges identified, this thesis makes the following contributions towards structured sentiment analysis on social networks.

### 1.5.1 Learning Concept Hierarchy from Short Texts Using Context Coherence

The problem that we addressed was how to extract a concept hierarchy from a given set of social media messages. The context coherence-based model (CCM), a top-down recursive model, was introduced to learn concept hierarchies from short and noisy texts by analysing the relationships between words. To achieve this, a novel measurement called *context coherence* was introduced to estimate the coverage of individual words in the whole document.

Context coherence was measured by the number of words that are related to a given word. A greater number of related words in a document implied that a word covered a wider range of aspects and that the size of the sub-hierarchy rooted in this word was relatively large. Unlike those in the existing models, the parameters of CCM (e.g., depth and width) can be automatically learned from the data.

Most current hierarchical models apply subjective methods, such as surveys, to evaluate the hierarchies they generate [5, 91]. Consequently, the results are dependent on the participants' experiences, and the preciseness and fairness of subjective evaluations can cause issues. Thus, the problem of how to evaluate the quality of hierarchical trees extracted from social media was considered. We proposed three methods to evaluate the quality of a hierarchy extracted from unstructured text. These methods reflect three important characteristics of an optimal tree: (1) *Coverage*, which reflects a topic on a high level, close to the root node, and should cover a wider range of sub-concepts than those on a lower level; (2) *Parent-child relentless*, which means the parent topic in the tree should be semantically related to its children rather than to its non-children; and (3) *Topic coherence*, where all words identified within a topic should be semantically related to the other words within that topic. We evaluated the performance of the approach with two real-world data sets. The experimental results showed that CCM can discover more prominent and coherent trees than the baseline methods.

The main points covered in this work are summarized as follows:

- A new measurement, namely, context coherence, was introduced to measure the containment relationship of words for the purposes of concept hierarchy construction.
- A new algorithm, CCM, was proposed that learns a concept hierarchy from short texts without a predefined hierarchy shape.
- Objective criteria was used to evaluate the quality of the concept hierarchy.
- Comprehensive experiments demonstrated the effectiveness of the proposed method in comparison with other approaches.

### 1.5.2 Structured Sentiment Analysis

A structured sentiment analysis (SSA) approach was introduced that incorporates hierarchy detection and sentiment analysis to automatically discover a hierarchy, as well as people's

opinions towards aspects within it, from social media texts. Combining sentiment analysis with hierarchy construction can effectively help to perform a fine-grained sentiment analysis on the concepts extracted in the tree. Structured sentiment analysis is important because it helps individuals and organizations understand people's interests in certain products and shows the reasons why they prefer them. In SSA, a top-down recursive approach was applied to extract the hierarchy and perform a fine-grained sentiment analysis. Then, the sentiment analysis was performed on only *hot aspects* that interest people. A hierarchical process was proposed for identifying the polarity of the parent node and the child node by extracting the closest opinion's words (e.g., verbs, adjectives and adverbs). Finally, a summarization approach was proposed to understand why people like those hot aspects. The main contributions of this work are summarized as follows:

- An approach to summarizing people's opinions based on an analysis of statements made in their messages was designed.
- A hierarchical sentiment approach for extracting hot aspects, discovering the relationships among them and identifying people's opinions towards them was proposed
- The approach was evaluated with three sets of real-world Twitter data. The experiment results showed that the proposed approach was effective for analysing short texts and extracting a sentiment tree.

### 1.5.3 Modelling User Attitudes with a Hierarchical Sentiment/Topic Model

A novel probabilistic model, the hierarchical user sentiment/topic model (HUSTM), was proposed for discovering the hidden structure of topics and users, while performing sentiment analysis. Modelling the attitudes or interests of users can give insight into user interests with respect to a variety of topics and help in analysing user' behaviours at any granularity level. The main goal of this study was to hierarchically model user attitudes (opinions) using different topic and sentiment information, including positive, negative and neutral. In the HUSTM, the entire structure is a tree with each node in the tree further separated into two sub-nodes: (1) the topic/sentiment node, which models word distribution over topic and sentiment (e.g., positive, negative or neutral); and (2) the user-sentiment node, which captures user attitudes using respective sentiment information. The main contributions of this work are summarized as follows:

- A unified model that discovers the hierarchical tree of topics, sentiments, and users from short texts without specifying the width and the depth of the tree was provided.
- An approach that groups users, who share the same topic and feelings, into the same community was designed.
- An approach that automatically infers the depth of the tree from stoical media was developed..
- The effectiveness of the proposed models was experimentally using three data sets. The results showed a higher-quality topical hierarchy discovered by the model when compared with other methods.

## 1.6 Thesis Organization

The remainder of the thesis is organized as follows. In Chapter 2, the body of literature related to the research topic is discussed. Chapter 3 discusses the first contribution to hierarchical structure detection in social media. Subsection 3.4 introduces the context coherence-based model (CCM) to learn concept hierarchies from short and noisy texts. In subsection 3.4.1, we propose a new notion called context coherence that identifies the semantic relationships of topics and discovers the hierarchical organization of those topics. Subsection 3.4.2 describes the top-down recursive algorithm to infer the concept hierarchy. Finally, subsection 3.4.3 shows the methods used to merge and group the duplicated concepts in the tree.

In Chapter 4, we discuss the SSA model and show its effectiveness in dealing with short and noisy text from social media. Discovering the hierarchical of concepts with the corresponding sentiment polarities can benefit everyone who needs to understand the current opinions on each concepts expressed in social media. Section 4.4 presents the proposed solution for the problem of SSA. In Subsection 4.4.2, we describe the process of creating the concept hierarchy. Then, subsection 4.4.3 discusses how the sentiment analysis is performed in the concepts extracted in the tree. Subsection 4.4.4 summaries the reason behind the people opinions. Chapter 5 focuses on the issue of modelling user interest in the topics discovered in the hierarchy. We used this system to automatically group users in the tree according to their interests. In subsection 5.4.1, we describe the generative process of HUSTM. Subsection 5.4.2 discusses the problem of grouping the user according to the topics of their

liking. In Chapter 6, the contributions of the research are discussed and suggestions and recommendations for future research are provided.

CHAPTER 2

# Literature Review

This chapter investigates past and current studies on topics related to the research project. The related research work is divided into four types: traditional topic models, user (author) interest analysis, sentiment analysis and hierarchical topic models. In section 2.1, flat models that extract topics from a collection of documents are introduced. In section 2.2, related research on sentiment analysis is described. Section 2.3 reviews the research on author discovery. Finally, in section 2.4, some of the related work on hierarchical topic models is described.

## 2.1 Traditional Topic Model

The traditional topic model is a type of statistical model for grouping words in order to find hidden topics within document collections. A topic contains a group of words that are semantically related and often appear together within the same context. In the literature, there are a number of topic models that have been proposed [96, 14, 12, 13]. Latent semantic analysis (LSA) [26, 25] is a popular method in the area of natural language processing (NLP). The main underlying idea of LSA is the examination of the relationships between words in a collection of documents. In LSA, words that occur in similar pieces of text are grouped within the same topic. The first step in LSA is to create a term-document matrix that describes the occurrences of words within documents. After construction of the matrix, singular value decomposition (SVD) is applied to the matrix for dimensionality reduction. In SVD, the matrix is further decomposed into the product of three other matrices $M = U \sum V^T$. where **U** and **V** are orthogonal matrices, and $\sum$ is the diagonal matrix that contains the singular values of the original matrix.

Probabilistic latent semantic analysis (PLSA) [35] is a statistical technique for the analysis

of two-mode and co-occurrence data. PLSA is proposed for solving the problems with LSA by using a generative model. The main goal of PLSA is to discover and distinguish different contexts of a word without using external knowledge. This is done in two ways. First, PLSA allows for distinguishing between words with multiple meanings. Second, PLSA groups words that share a common context under the same topic.

Latent Dirichlet allocation (LDA) [14] is a parametric probabilistic model that classifies text in a document on a particular topic. In LDA, each document is a mixture of a $k$ number of topics and each topic is represented as a mixture of words. A plate diagram of the LDA model is given in Figure 2.1. As the figure shows, the probabilistic topic model estimated by LDA consists of two matrices. The first matrix, $\Phi_k$, describes the probability or chance of assigning a particular word, when sampling a particular topic. The second matrix, $\Theta_k$, describes the probability of assigning a particular topic, when sampling a particular document. The Gibbs sampling [32] of LDA can be divided into two parts, the initialization and the sampling. In the initialization phase, LDA is recognizable in the assigning of words and documents to a random topic. Then, in the sampling phase, the data is observed and the correct topic is inferred for each word and document using:

$$P(w, z, \theta, \varphi, \beta, \alpha) \propto \left( \frac{n_{i,k} + \beta}{\sum_{r=1}^{V} n_{r,k} + \beta * V} \times \frac{n_{j,k} + \alpha}{\sum_{j=1}^{J} n_{j,k} + \alpha * K} \right) \tag{2.1.1}$$

where $n_{i,k}$ is the number of times the word i is assigned to the topic k and $n_{j,k}$ is the number of times the document j is assigned to the topic k. After a number of iterations, LDA will correctly infer the hidden topic for each document and topic. Using the equation above, LDA extracts the word-topic distribution from the first part and the topic-document distribution from the second part. The main drawback of conventional topic models is that they are parametric modes, in which the number of topics needs to be set manually[16, 97, 38].

The Chinese restaurant process (CRP) [4] is a non-parametric topic model that uses the analogy of customers seated at tables in a Chinese restaurant. CRP assumes a Chinese restaurant with an unlimited number of tables. Each table has an infinite capacity to seat customers. In the present case, a customer is the *word* and the table is a *coherent topic*. The first customer always sits at the first table. The next customer sits either at the same table as the first customer or at a new table. The decision can be calculated by the probability proportional to the number of customers already present or to an unoccupied table. The equations used to create the distribution are:

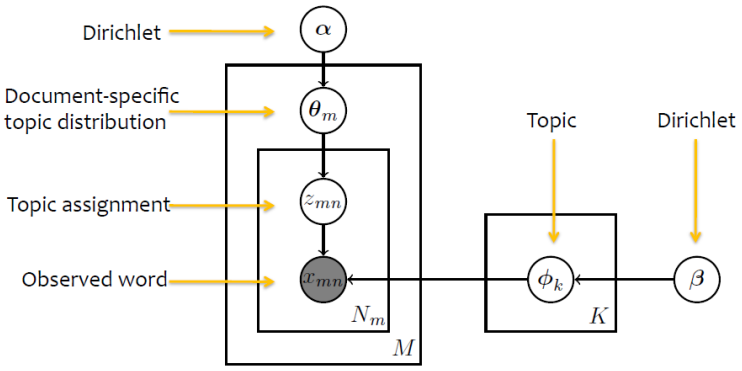



**Figure 2.1:** Latent Dirichlet Allocation [14].

$$P(occupied\ table\ i|previous\ customers) = \frac{m_i}{\alpha + m} \tag{2.1.2}$$

$$P(an\ unoccupied\ table|previous\ customers) = \frac{\alpha}{\alpha + m} \tag{2.1.3}$$

where $m_i$ is the number of customers sitting at table $i$ and $\alpha$ is a parameter. In CRP, the order in which the customers sit does not affect the final result. The main limitation of these topic models is that they generate a flat topic, so there is no relationship or structure among the discovered topics and sentiments.

With the development of social media, several research papers have made proposals for methods of handling social media content analysis in various domains, such as social community tracking [56], recommendations [78] and sentiment analysis [84, 49]. Xueqi et al. [101, 19] introduced a novel model for short text topic modelling, called the biterm topic model (BTM). The main idea of BTM is that it discovers topics by explicitly observing the word pair co-occurrence (biterm) in the corpus. In BTM, the topic is associated with a mixed distribution of word pairs. Other researchers have tried to combine short texts into large pseudo-documents to solve the word occurrence problem. They can then apply conventional topic models, such as LDA to reveal the hidden topics [98, 109, 76]. Some other studies have addressed short and sparse text in social media by self-aggregation and auxiliary word embeddings [51, 80, 39]. Chenliang et al. [51] developed a new topic model for short texts called GPU-DMM. Figure 2.2 shows the usage of word embedding to enhance topic discovery in social media. The idea behind GPU-DMM is that it extends the Dirichlet Multinomial Mixture (DMM) model by incorporating an external corpus to learn latent topic patterns and directly discover semantic relationships of learned words through the generalized *Polya urn* (GPU) model [61] in topic inferences. Although traditional topic models have been suc-

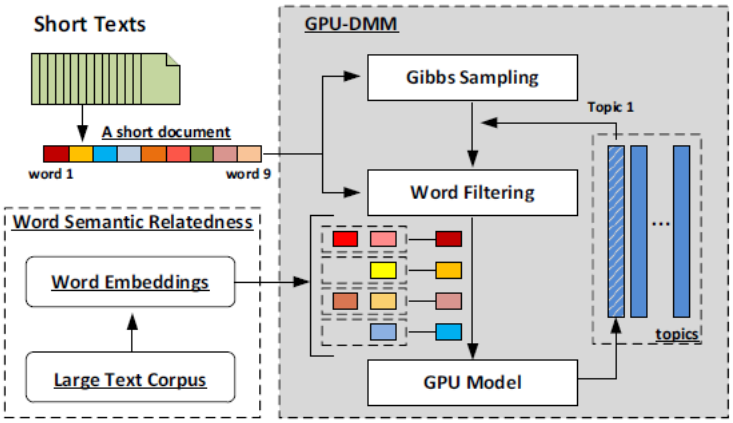



**Figure 2.2:** GPU-DMM Overview [51].

cessful in many real-world applications, the main limitation of these methods is that they only generate flat topics. Indeed, as a result, the natural hierarchical structure of a topic is neglected.

## 2.2 User (Author) Interest Analysis

This topic model is a type of statistical model for grouping words in order to find hidden topics within document collections. A popular topic model that represents documents as mixtures of topics is the LDA model, which models each topic as a distribution over words. A number of recent author topic models that merge the author's information into the topic model have been proposed [83, 41, 7, 62, 86]. Figure 2.3 shows the author layer integrated into the LDA model. The goal of the author topic model [82] is to discover underlying topics conditioned on the author's information, where each author is associated with a probability distribution over topics. As figure 2.3 shows, $a$ represents the author of a given word. $\Phi$, describes the probability or chance of assigning a particular word to a given author, generated from a symmetric Dirichlet $\beta$ prior. $\Theta$ describes the probability of assigning a particular topic for a given word. However, this author model does not provide any information about the sentiment attitudes of authors about different topics.

The CART model [74] was proposed to extract communities by using the semantic content of a social network, and it was one of the first attempts at integrating social links and content information for the purpose of community discovery. In comparison to these works that capture the authors' interests as a multinomial distribution over topics, the AIT model [43] introduced an additional layer between the author and topic layers that captured the authors' interests as a probability distribution of documents. Yan et al. [60] developed a

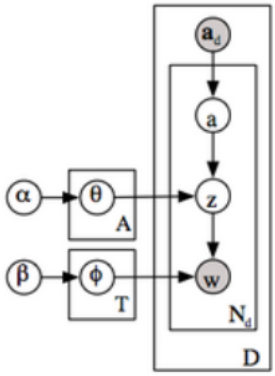



**Figure 2.3:** Author Topic Model [82].

framework called topic-link LDA that performs topic modelling and author community detection in a unified way. In [41], Shuhui et al. proposed an author topic model based collaborative filtering (ATCF) method that utilizes a user's information (e.g., textual descriptions of photos) in social media to reflect interests. All these models performed well at author interest analysis, but neglected the natural hierarchical structure of topics and sentiment information of users.

A variety of existing work is devoted to discovering a community and topic from text data [73, 31, 34]. In[105], Yin et al. proposed a community-based topic model called LCTA (latent community topic analysis) to integrate community identification into a topic model. In LCTA, text-associated graphs are used as input to discover users, who are linked to each other and share the same hidden topics. Zhou et al. [111] proposed the community profiling model COCOMP to discover communities as well as their associated topics. In [102], Yang et al. proposed a joint sentiment/topic model (STC) to simultaneously uncover communities, topics, and sentiment information.

Dynamic community discovery has also been proposed where communities are not static, but can change over time [46, 99, 50]. Li et al. [53] proposed a framework that can identify communities sharing similar topics, and capture the changes of the communities over time. Tang et al. [88, 24] proposed a novel community discovery algorithm that uses network structures. Palla et al. [72] provided a good community discovery method that detects overlapping communities to uncover the modular structure of complex systems. However, most of the existing community discovery methods identify the latent community from social networks without considering the natural hierarchy of communities, which can be of great importance and is the focus of this work. Another limitation of current models is that the interest of users in various topics is not considered.

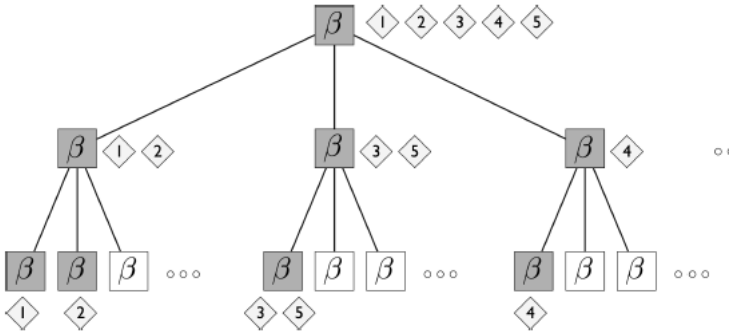


**Figure 2.4:** The hierarchy structure of nCRP.

## 2.3 Sentiment Analysis

One of the most popular areas in data mining is sentiment analysis. In data mining research, sentiment analysis is also known as opinion mining [11]. *Sentiment analysis* is defined as the process of identifying the opinions expressed in a piece of text as positive, negative or neutral. In this section, we divide the approaches to sentiment analysis into two types, traditional sentiment and joint sentiment/topic

### 2.3.1 Traditional Sentiment Approaches

One of the common ways to identify the sentiment polarities of different features of a product is with association mining rules. Hu and Li [36] proposed feature-based summaries for mining customer reviews. To achieve that, association mining rules are first used to extract the product features. Then, the opinion of each review is identified. Finally, a summary is generated of user opinions. Popescu and Etziono [79] improved the model by removing frequent nouns that are not features of the product. This technique is time-consuming because it uses the web to find product features. The lexicon-based approach was introduced to identify the sentiment polarity of features. Contained in the lexicon's set of opinion words (such as adjectives, adverbs, verbs and nouns) was the sentiment polarity (e.g., positive, negative or neutral). Hu and Li [37] used *Lexicon* to identify opinion words for two categories: *pros* and *cons*. Dictionary-based approaches were also developed that used the popular applications WordNet and SentiWordNet in order to analyse the positive and negative words using a scoring method (e.g., strongly positive, negative) [48, 107, 49, 75]. The limitation of this dictionary-based approach is that it depends on a specific domain.

### 2.3.2 Joint Sentiment Topic Approaches

The main idea of joint sentiment/topic approaches is that they discover the topics and sentiment polarities in a unified way. Each topic is divided into three polar topics: positive, negative or neutral. Each polar topic is represented as a mixed distribution over words. Several sentiment/topic models have been proposed to uncover topics with different sentiments [10, 68]. Lin et al. [55] introduced the JST model based on LDA. In JST, each sentiment polarity is associated with a mixture of topics, and all words are distributed over this mixture. In a study [42], the ASUM was proposed. It is a sentence-level model that assigns all words in a sentence to the same polarity. The TSM model [63] aimed at explicitly modelling the sentiment as a language model separate from topics. Indeed, TSM assumed the topic and sentiment were sampled independently, which meant the words were drawn from either topics or sentiments.

Similarly, Kawamae et al. [44] discovered topics and their corresponding sentiments by dividing the topics into three polar aspects (positive, negative and neutral). In this model, the topic category has probabilistic distributions over words, items and sentiment classes. The sentiment category has probabilistic distributions over words and ratings. Mukherjee et al. [67] proposed a semi-supervised approach to discovering aspect-based sentiment topics. In [67], a seeded aspect and sentiment category were used to identify the polarities of words. In [108], the authors proposed a User-sentiment Topic Model (USTM) that integrated user opinions into an LDA model. In USTM, a new layer is added to understand the interest of uses in several topics. In [103], a novel method was proposed for incorporating metadata (e.g., location, gender and age) into the topic modelling process to understand associations between metadata, topical aspects and sentiments. Subhabrata et al. [68] introduced the JAST model, extending LDA to learning different topic preferences, 'emotional' feelings about topics and writing styles. All of the models mentioned above extracted topics and sentiment as a flat mixture of topics. In real-world situations, topics have hierarchical relationships that can be discovered.

## 2.4 Hierarchical Topic Model

Hierarchical topic models have been proposed to discover hidden structures in documents, and several approaches to addressing the problems of hierarchical extraction will be discussed [27, 93, 59, 91, 92, 90]. Figure 2.7 shows the different assumptions of the hierarchi-

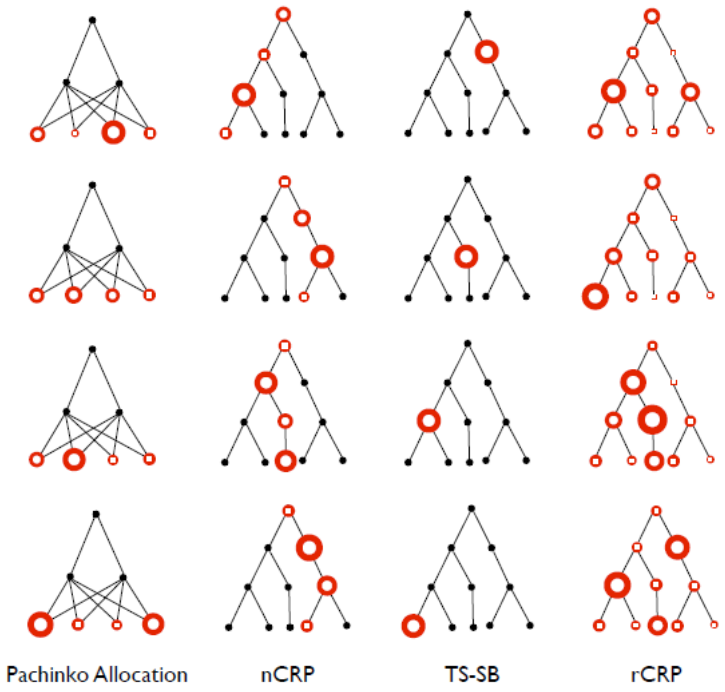



**Figure 2.5:** The different assumptions of the related models [45].

cal topic models, including the pachinko allocation model (PAM) [54, 65], the nested chinese restaurant process (nCRP), the tree-structured stick-breaking process (TS-SB) [30] and the nested chinese restaurant process (rCRP). The hierarchical PAM (hPAM) [54, 65] was developed to capture correlations between topics, and hPAM produces multiple levels of super-topics and subtopics. Each topic in the tree is a mixture of distributions over words. However, the hierarchical structure of hPAM is predetermined.

Blei et al. [15] proposed the nCRP generative probabilistic model to hierarchically learn latent structures from data. The nCRP extends CRP to represent the flat topic in a hierarchy. Figure 2.5 shows the hierarchy structure of nCRP, which assumes an infinite number of restaurants in the city in the analogy. One restaurant is considered to be the root node, and it contains an infinite number of tables. Each table in the restaurant points to another unique table in another restaurant in the next level of the tree. The restaurants are hierarchically organized into an infinitely branching tree.

The TS-SB [30] model assumes that a document is generated by a single node of the tree that has a unique topic mixture. Unlike the above methods, in nHDP [71, 66], the assumption is made that topics in a document can be generated from several paths. In both models, the depth and the number of child concepts are manually specified. In [100], Xu et al. proposed a novel knowledge-based hierarchical topic model (KHTM) that can integrate prior knowledge into topic hierarchy discovery. Moreover, Wang et al. [91, 92] proposed a novel

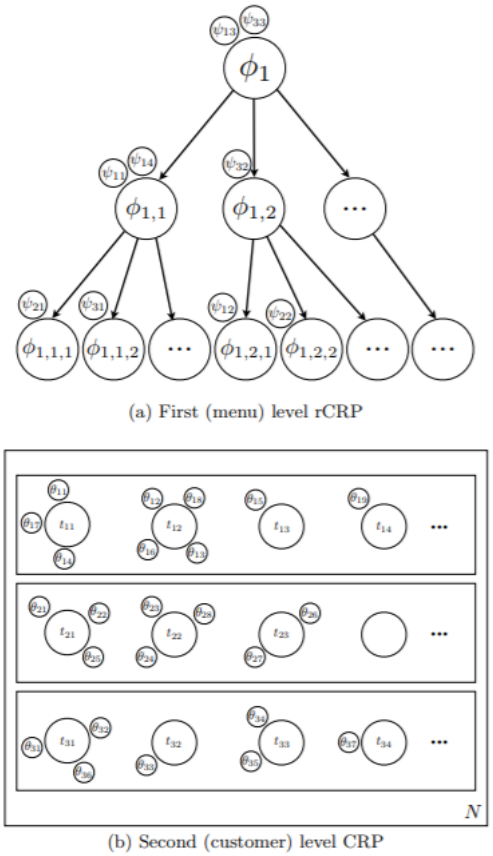



**Figure 2.6:** The first and the second assignment of topic hierarchy.

approach to recursively construct topics from a content representative document (the title). Then, a phrase mining and ranking approach is applied to rank a list of mixed-length words to represent every node in the hierarchy.

Joon et al. [45] suggested a new model, the rCRP. In rCRP, the nCRP is further extended to create a hierarchical tree in which each document has a distribution over all topics in the tree. rCRP consists of two main processes: table assignment and dish assignment. First, rCRP assumes words that are semantically related are clustered at the same table. Then, a dish is drawn in the tree for each table from the global menu. More specifically, the assignment of each word to a table is inferred by rCRP, while the semantic relationship between tables is determined by rCRP. Figure 2.6 shows the first and second assignments of topic hierarchy.

Recently, Kim et al. [47] applied hierarchical aspect-sentiment model (HASM), a more advanced model that automatically discovers the structure of aspects with corresponding sentiment polarities. Figure 2.7 is a graphical representation of HASM. In HASM, a prior tree is first generated randomly from the data using rCRP by randomly assigning each word to a node in the tree. Then, the approach uses Gibbs sampling to learn the posterior tree

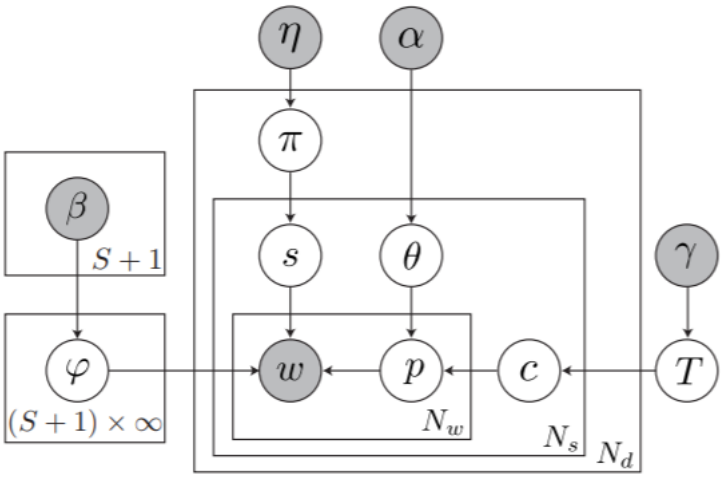



**Figure 2.7:** Graphical representation of HASM [47].

of three variables: aspect-sentiment node, the sentiment polarity of each sentence and the subjectivity of each word in the same sentence. To achieve that, HASM starts from the root node and calculates the possibilities of assigning a sentence to the current node, a child of the current node or a new node. The model uses the following equations to calculate the probability:

First, the aspect sampling of each sentence is modelled using,

$$P(w_{di}|s,p,\Phi_k,\beta) \propto \prod_{l=o}^{1}\left(\frac{n^{w,(.)}_{k,s_i\times l,-i}+\beta_{si\times l}}{\prod_w n^{w,(w)}_{k,s_{i\times l},-i}+\beta_{si\times l,w}} \times \frac{\prod_w n^{w,(w)}_{k,s_{i\times l}}+\beta_{si\times l,w}}{n^{w,(.)}_{k,s_i\times l}+\beta_{si\times l}}\right) \tag{2.4.1}$$

Second, the sentiment polarities are determined by,

$$P(s_{di}=k|w,s,p,c,\beta) \propto \left(n^{s,(k)}_{d,-i}+\eta \times P(w_{di}|s,p,\Phi_k,\beta)\right) \tag{2.4.2}$$

Finally, the subjectivity of each word is calculated using,

$$P(p_{di}=k|w,s,p,c,\beta) \propto \left(n^{p,(k)}_{d,i,-j}+\alpha \times \frac{n^{w,(v)}_{cdi,sdi\times k,-j}+\beta_{sdi\times k,v}}{\sum_{r=1}^{V} n^{w,(r)}_{cdi,sdi\times k,-j}+\beta_{sdi\times k,v}}\right) \tag{2.4.3}$$

where $n^{w,(.)}_{k,s_i\times l,-i}$ is the number of words in sentence $i$ assigned to topic $k$ and sentiment $s$; $n^{s,(k)}_{d,-i}$ is the number of $k$-polar sentences in document $d$; and $n^{p,(k)}_{d,i,-j}$ is the number of $k$-subjective words in sentence $i$ of document $d$.

In social media, people usually have a limited number of characters to use in discussing different topics (e.g., with Twitter, it is 280 characters). Also, it is common for users to use abbreviations and slang to express their feelings. The main limitation of the existing models

[9, 58, 69, 20] is that they have only been developed to deal with normal texts, and applying these hierarchical models with social media means the models will suffer from data sparsity. In other words, those models might produce incorrect and incomplete results. Another critical issue is that neither the sentiment information, nor the social communities, are considered by the existing models.

After an extensive literature search, the only work found that extracts the topic hierarchy from social media was [107]. Zhao and Li [107] developed an algorithm, based on formal concept analysis (FCA) [28], to extract *hot features* and organize them hierarchically in a tree. Next, they applied term frequency-inverse document frequency (TFIDF) to extract meaningful keywords and represent text as feature vectors. Then, external knowledge was used to filter noise and help discover the tree. These ways may be helpful in some specific domains, but not general since favourable external knowledge might not be always available. Another drawback is that the structure shape and the names of the nodes in the tree are manually specified.

In summary, despite the value of the above-mentioned studies, extracting hierarchical structures from short and noisy texts remains an open problem. This thesis aims to: (1) identify subsistent problems and challenges in hierarchical detection from social media, (2) design an effective algorithm for structure detection in social networks, (3) design approaches for sentiment analysis, and (4) propose an approach to hierarchically model the user interests express in social media.

CHAPTER 3

# Learning Concept Hierarchy From Short Texts Using Context Coherence

## 3.1 Overview

Uncovering a concept hierarchy from social media, such as tweets and instant messages, is critical for helping users quickly understand the main concepts and sub-concepts in large volumes of such texts. However, due to the sparsity of short texts in social media, existing hierarchical models fail to learn the structural relations among concepts, and lose an opportunity to discover the data more deeply. To solve this problem, a new notion called *context coherence* is introduced. *Context coherence* reflects the coverage of a word in a collection of short texts. Coverage is measured by analysing the relationships of words in complete texts. The major advantage of context coherence is that it aligns with the requirements of a concept hierarchy and can lead to a meaningful structure. Moreover, a novel non-parametric context coherence-based model (CCM) is proposed that can discover the concept hierarchy from social media texts without a pre-defined depth and width. The model was evaluated on two real-world datasets. The quantitative evaluations confirmed the high quality of the concept hierarchy discovered by the model compared with those of state-of-the-art methods.

## 3.2 Background

In recent years, social media. such as Twitter and Weibo, has become a popular form of information on the web. Short texts contain different latent concepts of a product or topic that can be hierarchically discovered. For example, in smartphone-related tweets, users discuss the main concepts of a smartphone, such as the overall design, battery capacity, screen size,

and camera. Constructing a concept hierarchy from short texts can help users understand the contents implied at different granularity levels and can facilitate many application functions, such as recommendation [107], summarization [5, 29], and sentiment analysis [49].

Hierarchical topic models have been previously proposed to effectively extract the hidden structures from traditional texts [18, 47, 45]. However, applying these models on short texts may result in less effective performance due to the sparsity of text. In Figure 3.1, we present part of a concept hierarchy created by a hierarchical aspect-sentiment model (HASM) [45] from a smartphone dataset to illustrate the problems of incoherent concepts and an unreasonable structure. In regards to incoherent concept, there are some words that are not semantically related to the other words. For example, the fourth concept created by the HASM contains irrelevant words (e.g., student and boyfriend); all words on the tree should be relevant and semantically related. Another problem is that the tree contains some duplicate concepts. For example, the children of the second and third concept are the same as the parent concept.

In social media, a few studies have addressed the problem of discovering a concept hierarchy from short texts [91, 95, 5], but these approaches do not fully exhibit the following three characteristics of an optimal tree. First, a concept on a high level, close to the root node, should cover a wider range of sub-concepts than those on a lower level. Second, a concept in a tree should be organized as parent and children concepts, where the parent concept is semantically related to its children rather than to its non-children. Third, the depth and width of the tree should be automatically inferred from the data.

To fill the gap in the current research, the Context Coherence-Based Model (CCM), a top-down recursive model, that learns concept hierarchies from short texts by analysing the relations between words is proposed. To achieve this, a novel measurement called *context coherence* to estimate the coverage of words in the whole document is introduced. *Context coherence is measured by the number of words that are related to a given a word*. A greater number of related words in a document implies that a word covers a wider range of aspects and that the size of the sub-hierarchy rooted in this word is relatively large. Unlike those in the existing models, the parameters of CCM (e.g., depth and width) can be automatically learned from the data. The hierarchy shape is inferred by the average context coherence of words in each level. Indeed, a minimum threshold is defined to limit the number of children concepts and control the depth of the tree.

Most current hierarchical models apply subjective methods, such as surveys, to evaluate the hierarchies they generate [5, 91]. Consequently, the results are dependent on the partic-

| HASM |
| --- |
| 1. headphone music plug adapter earphone<br>1.1 object activate olympic shauna nicoleobrien<br>1.2 code appleevent stuck toothpaste starbucks |
| 2. button ios applesupport power user<br>2.1 button ios applesupport power user |
| 3. commercial launch cell network<br>3.1 commercial launch cell network |
| 4. boyfriend student follower retweet babe |

**Figure 3.1:** A concept hierarchy created by HASM.

ipants' experience, and the preciseness and fairness of subjective evaluations are can cause issues. In this work, we also consider the problem of evaluating the quality of hierarchical trees extracted from social media. This task arises due to the use of subjective evaluations in the current research, such as using only a survey to test the quality of a hierarchy discovered by their models. We propose three methods to evaluate the quality of a hierarchy extracted from unstructured text. These methods are used to reflect three important characteristics of an optimal tree:(1) *Coverage*, which reflects a topic on a high level, close to the root node, and should cover a wider range of sub-concepts than those on a lower level; (2) *Parent-child relentless*, which means the parent topic in the tree should be semantically related to its children rather than to its non-children; and (3) *Topic coherence*, where all words identified within a topic should be semantically related to the other words within that topic. The main contributions of this chapter are as follows:

- A new measurement, namely, context coherence, is introduced to measure the containment relationship of words for concept hierarchy construction.
- A new algorithm, the CCM, is proposed to learn the concept hierarchy from short texts without a pre-defended hierarchy shape.
- Objective criteria was used to evaluate the quality of the concept hierarchy. Comprehensive experiments demonstrate the effectiveness of the proposed method in comparison with other methods.

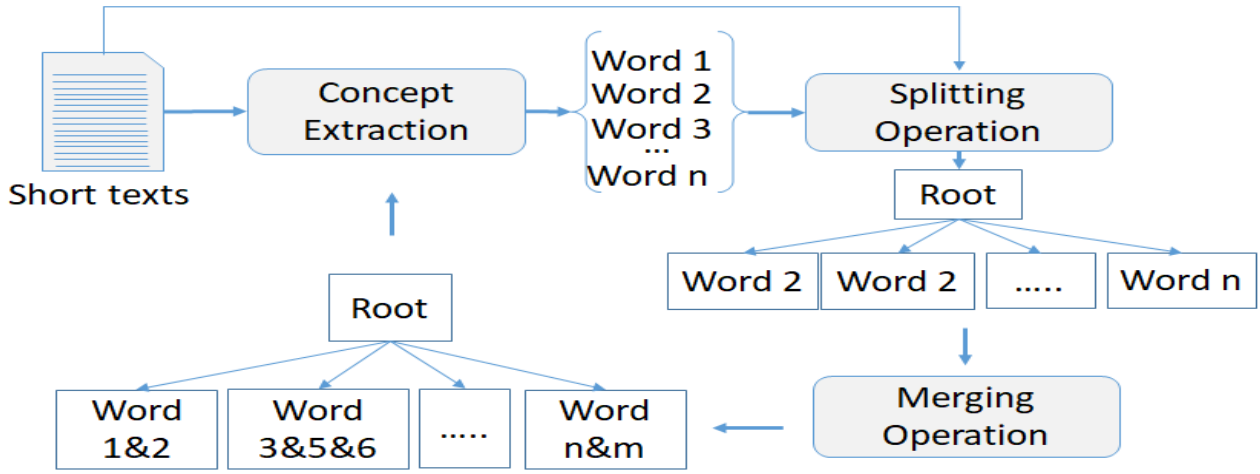



**Figure 3.2:** System Architecture.

## 3.3 Problem formulation

First, some closely related concepts are introduced below, and then the CCM problem is defined.

**Definition 1.** (Coherent Words). The basic unit of a concept hierarchy is a word. A coherent word $cw$ is one with a high coherence (coverage) score relative to other words in the vocabulary $V$.

**Definition 2.** (Concept). A concept, $t$, in a tree, $T$, is represented by either a single coherent word, $cw_i$, or as groups of coherent words, $t = \{cw_1, cw_2, cw_3...\}$, where every word $cw_1 \in t$ are refer to the same thing. A coherent word can appear in multiple concepts, though it will have a different order based on the coherence score in each concept. The number of words is decided by the merge operation (see Section 4.2).

**Definition 3.** (Concept Hierarchy ). A concept hierarchy is defined as T where each node in the tree is a concept. Every non-leaf concept, $t_i$ , has a number of children, defined as $chn^{t_i} = \left\{chn_1^{t_i}, chn_2^{t_i}, ...\right\}$. All children concepts should be semantically related to their parent concept.

**Problem 1.** Given a collection of short texts about a specific topic, $D = \{d_1, d_2, ..., d_l\}$, where $|D|$ is the length of $D$, the task is to extract a coherent concept hierarchy $T$ with unbounded depth and width. The output is a hierarchical tree of concepts, where each concept can be represented by multiple words. Words are within one concept, i.e. they are identical to the concept. For example, one concept in the tree contains three words: "picture", "pic", "photo". These three words all represent the same concept, "picture". "pic" is an abbreviation of "picture". "photo" is an alternative name for "picture". As another example, one concept contains two words "data" and "mining". These two words are used to represent a single

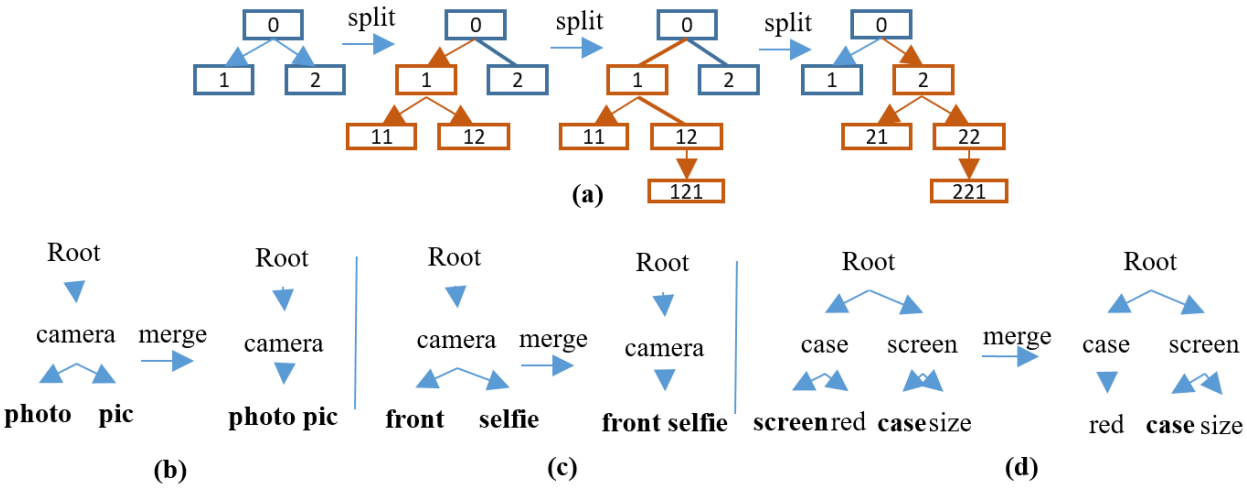



**Figure 3.3:** Split and merge operations. In the split operation (a), CCM starts from the root node and recursively creates a concept hierarchy. For the merge operation, CCM handles three cases where the concepts in the tree need to be merged: (b) synonyms merging, (c) relatedness merging, (d) duplication merging.

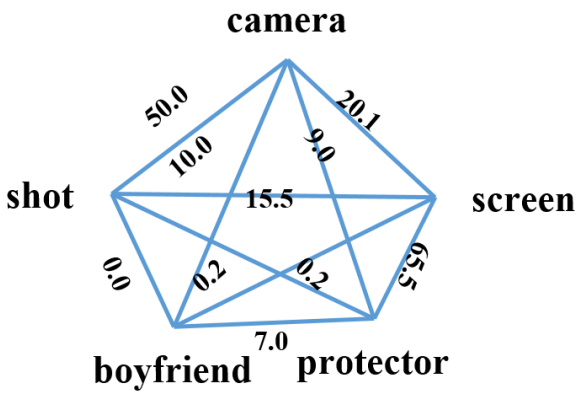
concept, "data mining" (which is a term instead of a single word).

To discover an optimal concept hierarchy from short texts, there are three important criteria:

- **Relatedness:** All root concepts of a sub-hierarchy should not only be related to their direct children, but also to all offspring. For example, the root node *iPhone* should be related to its sub-concepts *(camera, headphone, etc.)* and its sub-sub-concepts *(picture, adapter, etc.)*.

- **Coverage:** Concepts in a hierarchy should be organized from general to specific. The concepts near the root node must cover many documents, while those close to the leaf-nodes should have less coverage. For example, the parent concept "camera" in a hierarchy has more coverage than its children, *quality*, *selfie* and *front*.

- **Completeness:** A group of concepts should be combined as a single concept, if they co-occur significantly within the same contexts. For example, the concepts *picture*, *pic* and *image* should be combined into a single concept, $\{picture, pic, image\}$, because all three words refer to the same thing (i.e., *pic* is an abbreviation of *picture*, and *photo* is an alternative word for *picture*). Duplicated concepts should be removed from the tree, and similar concepts should be merged together. For example, the concepts "picture", "pic", "image" should be combined into a single concept (such as, "picture, pic, image").




| Word | Frequency | Average PMI |
|---|---|---|
| camera | 900 | 22.27 |
| boyfriend | 850 | 4.3 |
| screen | 600 | 25.32 |
| protector | 600 | 20.42 |
| shot | 500 | 16.42 |

**Figure 3.4:** A graph represents the similarity score for each word-pair. The table shows the frequency and the average context coherence score for each word.

## 3.4 Proposed Approach

Existing hierarchical models [94, 89, 104] learn concepts by observing document-level word co-occurrence, and their performance will be significantly influenced in the case of short texts. To address this problem, a novel CCM that learns concept hierarchies from short texts is proposed. Generally, the CCM automatically builds concept hierarchies in a three-step process. Figure 3.2 shows the system architecture for CCM. In the following sections, the architecture is described in detail.

### 3.4.1 Concept Extraction

Concept extraction is the main task of the CCM, where concepts are defined as either single words or groups of words. In this thesis, the notion of *context coherence* to extract concepts is introduced. The idea behind context coherence is to measure the coverage of a given word by analysing the associations among the words in the entire text. The coverage of a given word is calculated by identifying the number of words that are related to it. The relatedness reflects the similarity between the given word to other words in the text. If there is a larger number of related words, then the implication is that the word encompasses a wider range of sub-words. Figure 3.4 shows an example from the dataset. In the graph, the coherence score among all words in the text is given. The word *camera*, for example, has the highest coverage score. This indicates the word *camera* is linked to many words in the text and encompasses many sub-words. In contrast, the word *boyfriend* is related to a smaller number of sub-words. The difference between CCM and frequency-based models is that a word's

frequency implies the importance of the word with regard to the whole text in frequency-based-models. However, CCM assumes that a word is important if it encompasses a large number of words.

Given whole collections of short texts, $D$, similarity between $w_i$ and $w_j$ is first measured. Specifically, pointwise mutual information (PMI) [21] is employed to calculate the similarity of pairs as shown in Eq.3.4.1, where $P(w_i, w_j)$ indicates the probability that two words, $w_i$ and $w_j$, occur together in texts, while $P(w_i)$ and $P(w_j)$ indicate the occurrence probability of $w_i$ and $w_j$ in the texts, respectively.

$$W_{i,j} = log \frac{p(w_i, w_j)}{p(w_i)p(w_j)} \tag{3.4.1}$$

To compute the context coherence, we represent the text as a term-term matrix $M$ in which each row and column stands for all unique terms in $V$. Each cell describes the word-pair similarity score in short texts. The context coherence of a given word $w_i$ is calculated by taking the average similarity score with other words using the equation below:

$$CC(w_i) = \frac{1}{n} \sum_j W_{i,j} \tag{3.4.2}$$

where $n$ is the number of words in $D$. The model uses the average PMI for term-term analysis, because in the vocabulary of short texts, most pairs of words do not appear together frequently. That is, the PMI between most pairs of words is negative. The average PMI of a word is decided based on how many words are not related to it. This aligns with the definition of a word's context coherence. Hence, average PMI is a good approximation of context coherence.

### 3.4.2 Hierarchical Extraction

The key idea of the approach is to learn a concept hierarchy from short texts based on the coherent words identified. The hierarchical extraction function consists of two main components: a splitting process and a merging process. Splitting is performed by a recursive algorithm that is responsible for generating a hierarchical tree, while the merging process is responsible for grouping similar concepts under a new concept. A concept hierarchy is defined as a tree $T$, where each node close to root has a higher coherence score than the ones near to the leaf-nodes. Moreover, the concept hierarchy reflects the intuition that the root concept of sub-hierarchy is not only related to its direct children but also all its offspring.

For example, root node *IPhone* should be related to its sub-concepts *camera, app, ...etc* and its sub-sub-concepts *picture, release*.

**Algorithm 3.1** Splitting operation.

**Data:** $D, minS$

**Result:** Build a concept hierarchy T.

**Function** *Recursive(D)*
  **foreach** $w_1 \in V$ **do**
    **foreach** $w_2 \in V$ **do**
      **if** $w_1 \neq w_2$ **then**
        $CC(w_1, w_2) = \frac{1}{n} log \frac{p(w_1,w_2)}{p(w_1)p(w_2)}$
        $cw.add(w_1, CC(w_1, w_2))$
      **end**
    **end**
    **foreach** $t1 \in cw$ **do**
      **foreach** $t2 \in cw$ **do**
        **if** $average > minS$ ***and*** $t_1 \neq t_2$ **then**
          $t3 = merge(t1, t2)$
          Add $t3$ to $T$
          $Recursive(split(D, t_3))$
        **end**
      **end**
    **end**
    $average = \frac{1}{k} \sum_{i=0} CC(t_i)$
  **end**

### Splitting operation

The first goal of the CCM is to create a flexible tree in which each parent concept has better relatedness to its child concepts than non-child concepts. The CCM's second aim is to build a hierarchy in which concepts are general near the root and more specific near the leaves. To achieve this aim, the splitting operation takes the extracted candidate concepts as input to recursively build a tree. It recursively partitions a current concept into a number of sub-concepts. For example, if concept $t_1$ describes a *camera* and concept $t_2$ describes a *headphone*, then the whole document will be partitioned into two sub-documents. In this way,

all concepts in the same path should be semantically related and refer to the same thing. Another advantage of the model is that the shape of the tree (e.g., its depth and width) is automatically determined from the data. The number of concepts in each level is inferred by removing the child concepts whose context coherence is less than a predefined threshold. For the depth, the CCM stops the splitting process when the average coherent score of the concepts is less than the threshold. Note that not all candidate concepts are considered in the tree. Only the concepts that exceed the specified threshold are kept.

Given a document $D$ and stopping criteria $minS$, the specific recursive process of the approach can be described as follows (i.e., Algorithm 3.1). For each word, $w_i \in D$, calculate the context coherence $CC(w_i)$ to the other words in the document, using Equations (3.4.1) and (3.4.2) (Line 5). If the coherence score exceeds a predefined threshold $minS$, do one of the following: (1) Add a word as a concept, $t$ to $T$ and then split the whole document into a number of sub-concepts (sub-documents), (2) Create a new concept by merging similar concepts (Line 7). Splitting is stopped when the average coherence score, $\frac{1}{k}\sum_{i=0} CC(t_i)$, of the concepts in $L$ level is less than the stopping criteria, $minS$, where $k$ is the number of concepts created in level $L$. The process is applied again recursively for every generated sub-document to extract a concept hierarchy.

**Algorithm 3.2** Merging operation.

**Data:** $minM = 0.60\%$

**Result:** Find the least common ancestor t of $t_i$ and $t_j$.

1 $chn^{t_i} = \left\{chn_1^{t_i}, chn_2^{t_i}, ...\right\}$

2 $chn^{t_j} = \left\{chn_1^{t_j}, chn_2^{t_j}, ...\right\}$

3 $score = overlap(chn^{t_i}, chn^{t_j})$

4 **if** $score > minM$ **then**

5 | **Return** $t_3 = t_i + t_j$

6 **end**

7 **else**

8 | **Return** $t_i$

9 **end**

10 **if** $t_i \in$ ***and*** $chn^{t_j} t_j \in chn^{t_i}$ **then**

11 | Remove $t_i$ from $chn^{t_j}$

12 **end**

**Merging operation**

All concepts created by the splitting operation contain a single word. The merging operation is responsible for finding similar words and grouping them under a new concept. For the merging operation, there are three situations where concepts need to be merged. First, people often use different words to refer to the same concept (i.e., synonyms), such as *photo*, *pic* and *picture*. Those words rarely appear next to each other in the same text. The merging operation aims to find these kinds of words and combine them. Second, the CCM also tries to group words that share the same context, such as *front* and *selfie.* Third, in some situations, there are concepts that appear twice in two branches, such as *screen* $\rightarrow$ *case* and *case* $\rightarrow$ *screen*. Figure 3.3 shows the three examples of the merging operation. The CCM handles such duplications by removing one of them from a concept in the tree.

For the first two cases, the algorithm 3.2 finds the common sub-concepts of the concepts $t_i$ and $t_j$ and then merges them into a new concept $t$, either if $chn^{t_i} \subset chn^{t_j}$ or if the overlap score of two concepts exceeds the predefined threshold $minM$. In the experiment, the overlap threshold was set to 0.60. The overlap score of two concepts was calculated using a Jaccard similarity measure.

$$overlap(t_i, t_j) = \frac{|chn^{t_i} \cap chn^{t_j}|}{|chn^{t_i} \cup chn^{t_j}|} \tag{3.4.3}$$

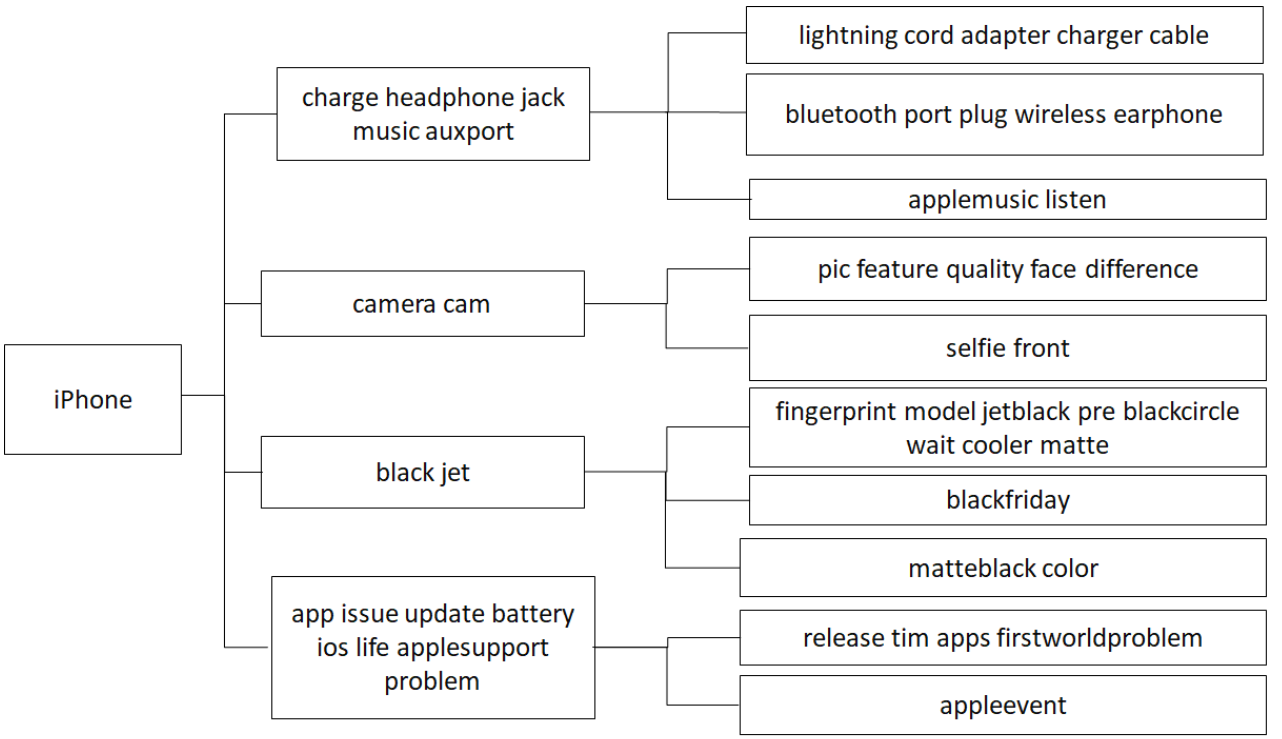
where $chn^{t_i}$ and $chn^{t_j}$ are the children of concepts $t_i$ and $t_j$. For case 3, the algorithm checks if concepts $t_i \in chn^{t_j}$ and vice versa. Then, the common concept will be deleted from one of them.

## 3.5 Experiments

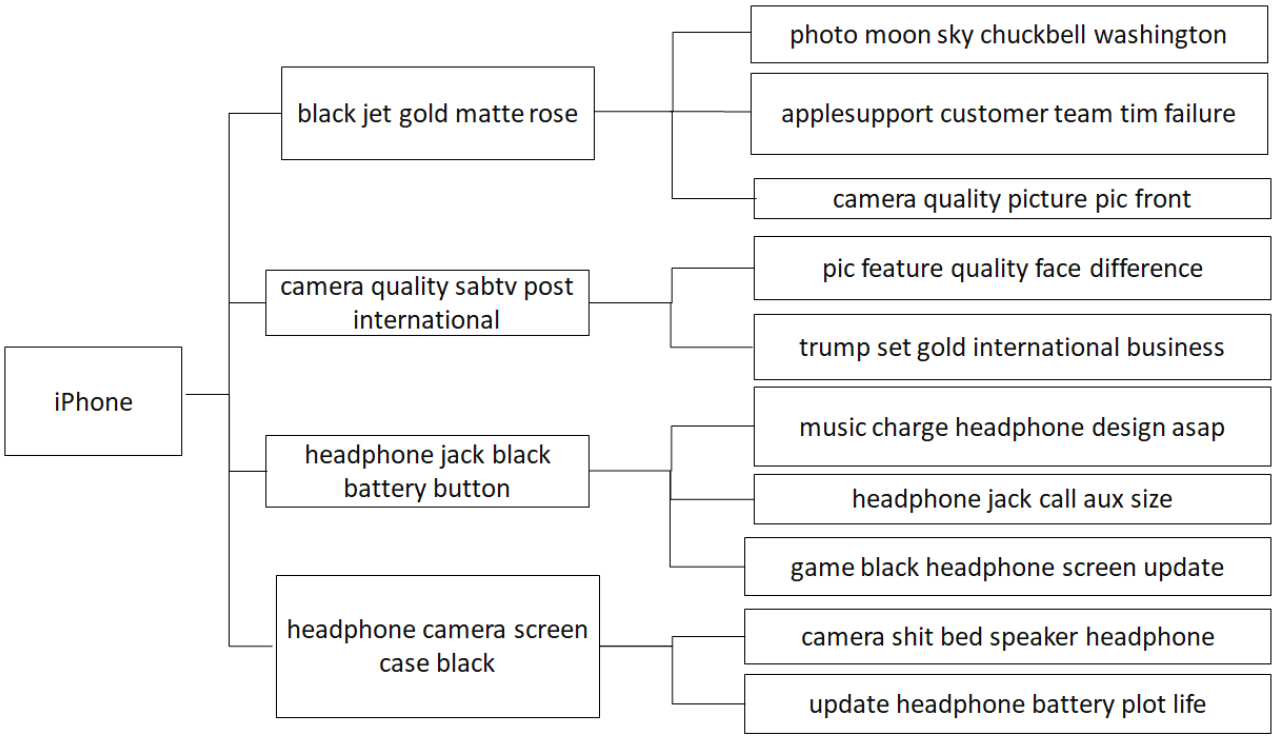
In this section, the dataset and the methods used for evaluation are introduced. Then, the experimental results are presented. The performance in terms of concept coherence, coverage and parent-child relatedness are also reported. The experimental results showed that the proposed model provides promising results.

### 3.5.1 Datasets

The method was tested on two real-world short-text corpora. In the following section, brief descriptions of each is provided.

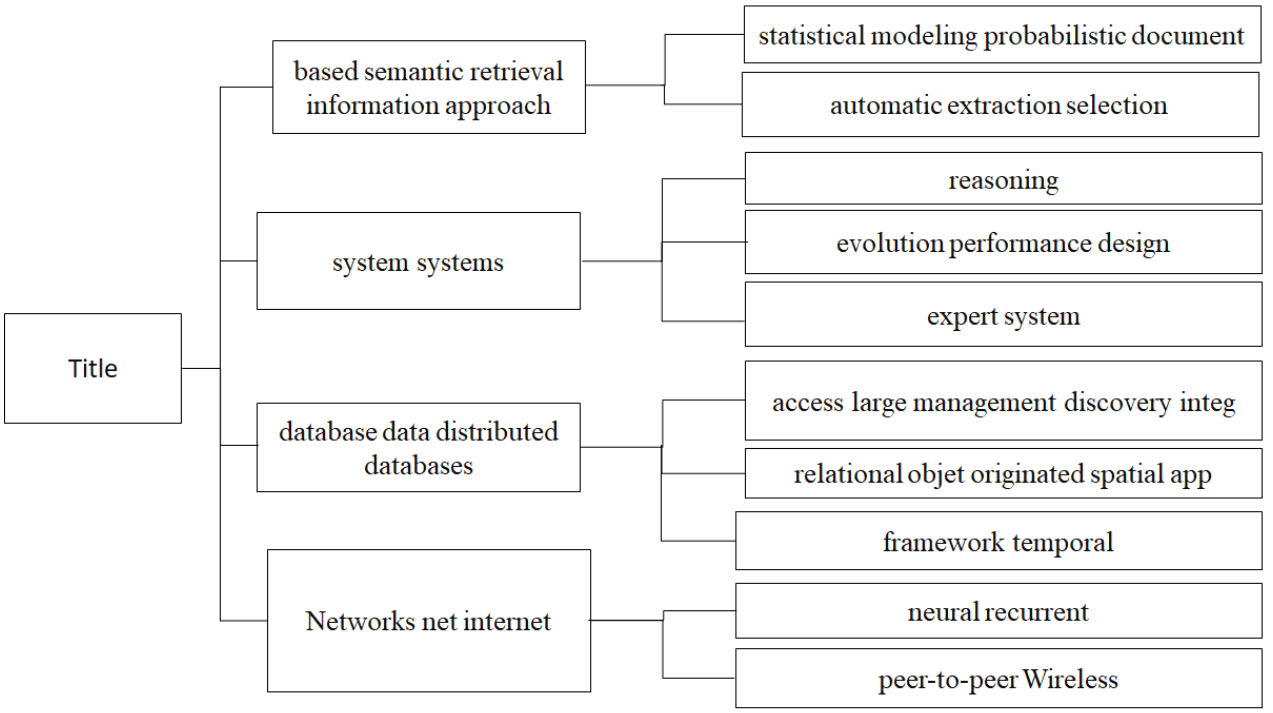



**(a)** CCM

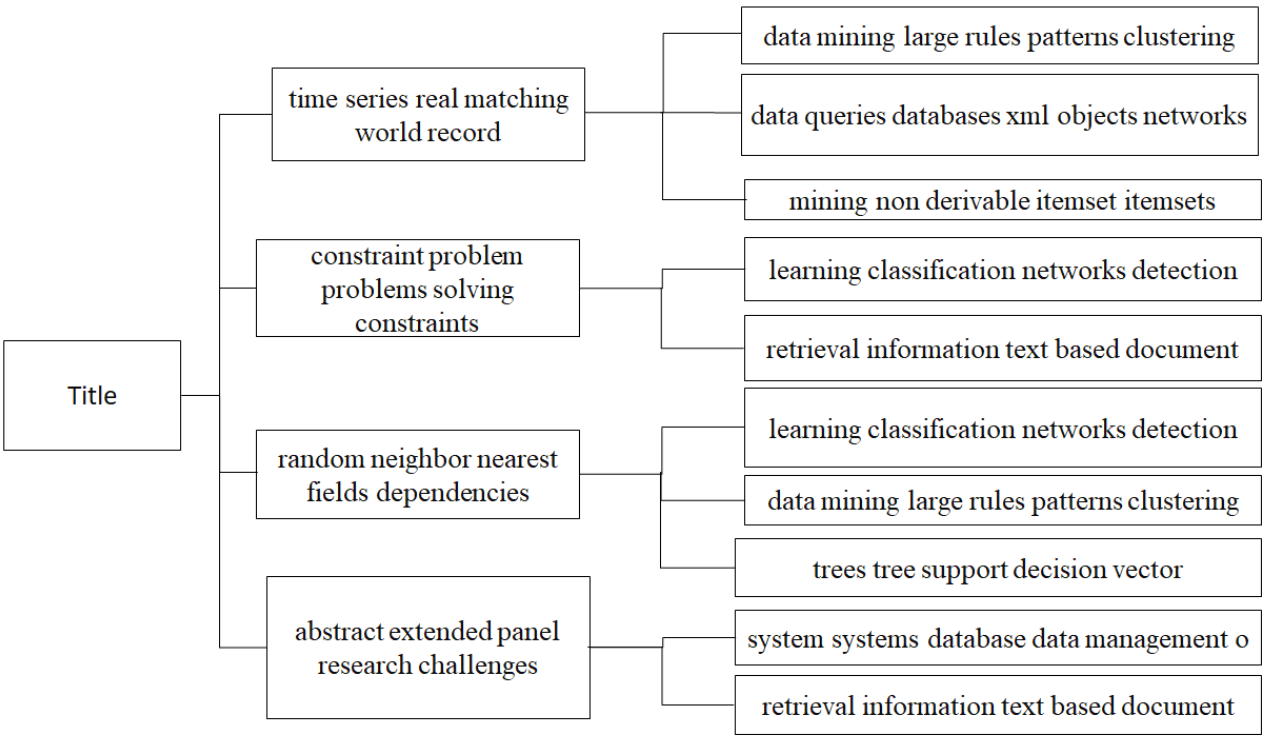



**(b)** rCRP

**Figure 3.5:** A subset of the concept hierarchy created by CCM and rCRP from the smartphone dataset. The root was defined as a first-level node, and the second and third level concepts are shown. For each method, the most relevant sub-concepts are displayed.




**(a)** CCM




**(b)** hPAM

**Figure 3.6:** A subset of the concept hierarchy created by CCM and hPAM in DBLB dataset. The root was defined as a first-level node, and the second and third level concepts are shown. For each method, the most relevant sub-concepts are displayed.

| CCM |
|---|
| 1. headphone aux music charger adapter |
| 2. charger problem cord<br>2.1 listen compliant firstworldproblem earphone<br>2.2 listen port plug aux |
| 3. applemusic music airpod<br>3.1 tim rating download universe<br>3.2 discipline price play buy |
| 4. bluetooth speaker ideal |

| rCRP |
|---|
| 1. headphone jack music charger aux |
| 2. charge headphone dilemma firstworldproblem<br>2.1 charge reservation geekuranjil nothin<br>2.2 music charge oneisenouge dram |
| 3. Life headphone battery jack<br>3.1 cpwtweet jasonkenny evelinsthought win<br>3.2 native camera experience screen |
| 4. headphone camera case singapore comment |

**Figure 3.7:** A part of the concept hierarchy obtained by CCM and rCRP on the Smartphone dataset.

- **Smartphone**. A collection of more than 68,000 distinct tweets was retrieved using the Twitter API [2]. This dataset had been used in a previous study on concept hierarchies [5]. The raw data was very noisy. Hence, the following preprocessing was performed on the dataset: (1) letters were converted to lowercase; (2) all non-alphabetic characters, stop words, and URLs were removed and (3) words with fewer than 2 characters were removed.

- **DBLP**. A collection of 33,313 titles was retrieved from a set of recently published papers in computer science in six research areas: data mining, computer vision, databases, information retrieval and artificial intelligence. This dataset has been previously used in [52, 110].

### 3.5.2 Methods for Comparison

The approach was compared with three typical models of hierarchical construction.

- **rCRP** [45]. A non-parametric hierarchical model that recursively infers the hierarchical structure of topics from discrete data. To generate a tree, its hyperparameters were tuned to find the same number of topics.

- **hPAM** [65]. A parametric hierarchical model that takes a document as input and generates a specific number of super-topics and sub-topics.

- **SSA** [5]. This is a recursive state-of-the-art hierarchical model that extracts a tree with a specified depth and width.

- **HASM** [45]. A non-parametric hierarchical aspect sentiment model that discovers aspects with the corresponding sentiment polarity. In the experiment, its hyperparameters were tuned to extract the same number of concepts as those of other methods.

- **CCM**. For the evaluation of the model , the CCM's parameters were set to approximately generate the same tree. For all methods above, its hyperparameters were tuned to discover the same number of concepts as the other methods.

### 3.5.3 Concept Hierarchy Visualisation

CCM was design to produce a high-quality hierarchical structure of concepts. Figure 3.6 showS a part of a concept hierarchical structure discovered by CCM, rCRP and hPAM. Clearly, CCM created a hierarchical tree where each parent concept is related to its direct child concepts. For instance, the child concepts *lighting cord adapter..etc* under the parent concepts *headphone jack..etc* are related. Also, CCM shows that as the depth of the tree increases the concepts become more specific. However, in rCRP and hPAM, there are some child concepts that are not related to the paired parent concepts. For example, the concepts *trump set gold ..etc* under concepts *camera quality* contain some irrelevant entries. In the following section, we quantitatively analyse the quality of the tree and present the quantitative results.

### 3.5.4 Evaluation Measures

A standard way to assess model quality is to measure held-out perplexity [45, 71]. This method is not appropriate in this case, because neither the quality of a concept hierarchy, nor the semantic quality, is considered in perplexity. The existing hierarchical models use subjective methods (e.g., surveys) to measure the goodness of the hierarchy they discover. Consequently, participants may not feel encouraged to provide accurate, honest answers.

After an extensive literature searcher, no commonly used metrics were found for measuring the goodness of a concept hierarchy constructed from short texts. There is a need for a universal method that measures the capability of a hierarchical model in discovering an optimal tree. In this thesis, three measures to quantitatively evaluate the quality are introduced, concept coherence, coverage and parent-child relatedness. First, all the top words representing a topic in the tree should be coherent. Second, a topic on the tree should be organized from general (closer to the root node) to specific (nearer the leaf node). Third, a topic in a tree should be organized as parent and children topics, where the parent topic is semantically related to its children, rather than to its non-children. These metrics are then used to compare the characteristics of a concept hierarchy constructed by the model with baseline methods.

**Table 3.1:** Average coherence score

| | **SmartPhone** | | | **DBLP** | | |
|---|---|---|---|---|---|---|
| | Level 1 | Level 2 | Level 3 | Level 1 | Level 2 | Level 3 |
| **CCM** | -1.58 | -1.99 | -2.42 | -2.91 | -3.01 | -3.10 |
| **rCRP** | -3.38 | -3.14 | -2.54 | -3.18 | -3.23 | -3.14 |
| **hPAM** | -1.84 | -2.85 | - | -2.99 | -3.05 | - |
| **HASM** | -2.60 | -2.03 | - | -3.17 | - | - |
| **SSA** | - | - | - | - | - | - |

**Quality of Concepts**

In the topic hierarchy, each topic is represented by a list of top words. The topic coherence is based on the idea that all words in this topic should be consistent with the semantic meaning of other words. For example, the topic *picture*, *pic*, *image* is coherent, because all three words refer to the same thing (i.e., *pic* is an abbreviation of *picture*, and *image* is an alternative word for picture). In order to evaluate the coherence of topics, an automated metric, namely coherence topic, proposed by Mimno et al [64] was utilised. Suppose a concept $t$ is characterized using a list $t = \{w_1^t, w_2^t, ..., w_n^t\}$ of $n$ words. The coherence score of $t$ is given by:

$$Coherence(t) = \sum_{i=2}^{n} \sum_{j=1}^{i-1} log \frac{D(w_i^t, w_j^t) + 1}{D(w_j^t)}. \tag{3.5.1}$$

where $D(w_i, w_j)$ is the number of documents containing both $w_i$ and $w_j$. $D(w_j)$ is the number of documents containing a word, $w_j$. In the experiments, the number of words in each concept was set to five. Since the model can produce concepts with less than five words, concepts that contained five words were evaluated. To evaluate the overall quality of a concept set, the average coherence score for each method was calculated. Here, only the score related to three levels of a concept hierarchy is shown. The results are illustrated in Table 3.1. A higher coherence score indicates a higher quality concept. For both datasets, the results show that the CCM achieved significant improvements compared with rCRP and hPAM. However, due to data sparsity and the shortness of the texts, the HASM failed to discover a comprehensive concept hierarchy. In the SSA mode, the concept coherence was not evaluated because the concept in the tree was represented by a single word.

**Table 3.2:** Average coverage score

| | **SmartPhone** | | | **DBLP** | | |
|---|---|---|---|---|---|---|
| | Level 1 | Level 2 | Level 3 | Level 1 | Level 2 | Level 3 |
| **CCM** | -0.80 | -0.92 | -0.98 | -0.66 | -0.84 | -0.90 |
| **rCRP** | -0.65 | -0.66 | -0.72 | -0.32 | -0.50 | -0.43 |
| **hPAM** | -0.84 | -0.72 | - | -0.64 | -0.51 | - |
| **HASM** | -0.78 | -0.83 | - | - | - | - |
| **SSA** | -0.88 | -0.94 | -0.96 | -0.72 | -0.83 | -0.87 |

**Coverage**

The second important characteristics of the concept hierarchy is the coverage of the concepts which reflects that the concepts near the root node should have a higher coverage than those close to the leaf nodes. For example, the parent topic *"battery"* in a hierarchy has better coverage than its children, *life*, *usage* and *capacity*. Given the $N$ top words of a concept, $t_z = \{w_1, w_2, w_3, ...w_N\}$, the top five words in the whole document were replaced with the first word. For example, if the top words of a topic are *picture*, *image*, *pic* and *photo*, every document containing any of these words had them replaced with the first word. It was assumed that all the words under the same concept refer to the same thing. The coverage score was calculated as follows:

$$Coverage(L) = \frac{1}{z}\sum_{z} PMI(t_z). \tag{3.5.2}$$

$$PMI(t_z) = \frac{1}{n}\sum_{j} log\frac{p(w_1^{t_z}, w_j)}{p(w_1^{t_z})p(w_j)}. \tag{3.5.3}$$

where $z$ is the number of concepts in level $L$. The default value of $N$ was set to 5 in the experiments. A higher coverage score indicates a higher quality concept. The results are illustrated in Table 3.2. For all datasets, the CCM and SSA clearly show a decrease in the coverage score when the depth of the tree increases, which indicates the concepts near the root nodes are general concepts, while those near the leaf-nodes are specific concepts. Unlike the current model, the patterns in rCRP and the hPAM were different. For example, in rCRP, the context coverage of concepts at the third level was always higher than that of the concepts at the second level, which indicates that the concepts generated by the model were not organized from general to specific. The reasons the current model outperformed the

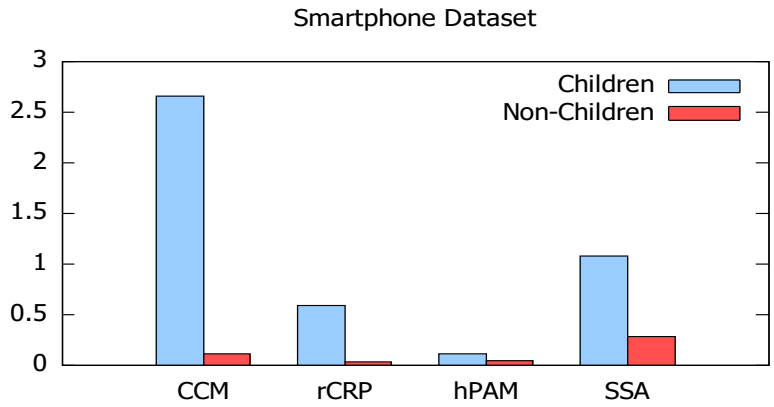



**Figure 3.8:** Parent-Child relatedness or Smartphone dataset.

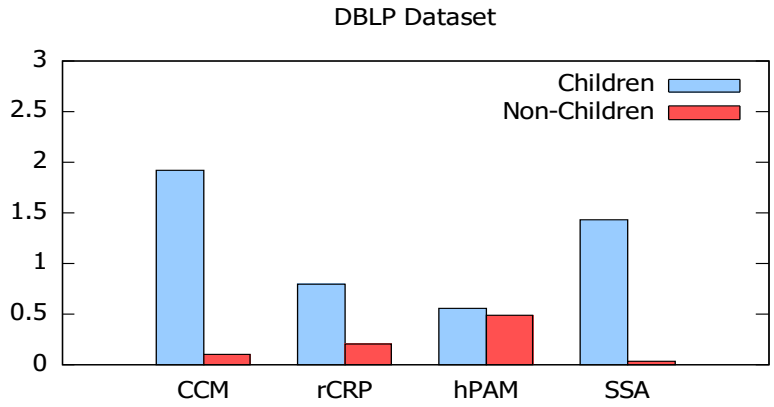



**Figure 3.9:** Parent-Child relatedness for DBLP dataset.

current baseline model was that it used pair-word relations to discover words which had better coverage and then applied the recursive approach to organize those concepts from general to specific.

### Parent-Child Relatedness

The third evaluation was meant to assess parent-child relatedness. In other words, it was assumed that parent concept $t$ should be more similar to its direct children than to the children that descend from other concepts. For example, the root node *iPhone* should be related to its sub-topics *(camera, headphone, etc.)* and its sub-sub-topics *(picture, adapter, etc.)* For simplicity, the relatedness score was only computed for a parent concept at the second level, with the children concepts at the third level. Given a concept $t$, the concept's relatedness score to its children was measured and compared to that of its non-children using Equations (3.5.4) and (3.5.5):

$$Children(t) = \frac{1}{k}\sum_{k}\frac{D(t, chn_k^t)}{D(t), D(chn_k^t)} \tag{3.5.4}$$

$$Non - Children(t) = \frac{1}{k}\sum_{z}\sum_{k}\frac{D(t, chn_k^z)}{D(t), D(chn_k^z)}, z \neq i \tag{3.5.5}$$

where $D(t, chn_k^t)$ is the number of times parent concept $t$ appears with its child concept, $chn_k^t$ and $D(t, chn_k^z)$ is the number of times parent concept $t$ appeared with its non-child concept $chn_k^{tz}$. The overall parent-child relatedness was measured by taking the average score of relatedness to children and non-children concepts for all parent concepts at the second level. Figure 3.7 and 3.8 illustrates the parent-child relatedness of four models. The higher relatedness scores indicate that a parent node was more similar to its children nodes, compared to non-children nodes at the same level. Both the CCM and SSA showed significant differences between children and non-children, while the nCRP and hPAM did not. The relatedness of the HASM was not calculated for the smartphone dataset since it produced duplicate children. In summary, for both datasets, CCM consistently outperformed the other tested methods, demonstrating that the model produced a higher quality hierarchy from the social media text. The limitation of the current state-of-the-art methods is that they capture the frequency of words to build a concept hierarchy. Due to the nature of short text, the current methods assume only highly frequent words are important provide relevant concepts in a hierarchy. Less frequent, but relevant, concepts are neglected by those models. In contrast, CCM considers both less and more frequent concepts, if they have a high coherence score. In this way, highly frequent concepts with low coherence are irrelevant concepts, while the ones with low frequency may be relevant.

## 3.6 Summary

Discovering concept hierarchies from social media is significantly critical because of the prevalence of short texts on the Internet. In this thesis, the non-parametric CCM for social media was proposed. The CCM can automatically discover a concept hierarchy by observing and analysing the relations between words in whole texts. This can be done by the proposed new measurement, *context coherence*. Context coherence was used to analyse words in social media texts and to determine the similarities among them. The semantic similarity was measured by observing and analysing the context of a given word. The results demonstrated that the approach can discover higher quality trees than previous methods. Another

advantage of the CCM is that it is simple, effective and easy to implement. In the future, the approach can be extended to automatically extract concepts with co-sponsoring sentiment polarity. Additional performance evaluation can also provide improvements on the approach. These advantages make the CCM a promising tool for social media analysis and concept hierarchy extraction.

CHAPTER 4

# Structured Sentiment Analysis

## 4.1 Overview

Extracting the latent structure of the aspects and the sentiment polarities is important as it helps customers to understand people' preference to a certain product and show the reasons why they prefer this product. However, insufficient studies have been done to effectively reveal the structure sentiment of the aspects from social media texts due to the shortness and sparsity. In this chapter, we propose a structured sentiment analysis (SSA) approach to understand the sentiments and opinions expressed by people in social media. The proposed SSA approach has three advantages: 1) automatically extracts a hierarchical tree of a product's hot aspects from short texts; 2) hierarchically analyses people's opinions on those aspects; and 3) generates a summary and evidence of the results. We evaluate our approach on popular products. The experimental results show that the proposed approach can effectively extract a sentiment tree from social media.

## 4.2 Background

Discovering the latent structure of the aspects and their cosponsoring sentiments is significantly important from two points of view - individuals and business intelligence. From the individual's view point, a sentiment tree organises aspects from general to specific. Therefore, it allows individual to find people's attitudes on various aspects represented by the tree at different granularities. For example, some may be interested in people's opinions on the product in general, while others may look for people's opinions on specific aspects, such as the quality of the camera. From the point view of business, structured sentiment tree would allow them to trace public opinion on aspects of a product and services, and provide them

with important information to help them improve future plans and strategies.

Recentelly, researchers have proposed new approaches to effectively extract the hierarchical structure from text [94, 17, 112, 22]. For example, Kim [47] and Titov [90] studied the problem by proposing a model that discovers a hierarchy from review data. However, no sufficient studies have been done to effectively reveal the hierarchical tree of the hot aspects and their corresponding polarities form social media texts. In fact, the existing approaches have only been designed to deal with traditionally long texts, such as online reviews and blogs. In general, their performance is less effective when these methods are applied to short texts in social media [77], due to both the shortness and the sparsity of the texts.

There are three major challenges when discovering hierarchical sentiment tree from social media texts. Firstly, compared with traditionally long texts, social media texts suffer from sparsity, and this issue may result in an incomprehensible and incorrect concept hierarchy. Secondly, most existing methods perform a flat sentiment analysis on each extracted aspects independently, and ignore the concept hierarchy. In fact, the sentiment polarity for an aspect should also include the polarity of its offspring. Otherwise, the polarity of this aspect may not cover all people's genuine attitudes on it. Thirdly, generating understandable and convincing summaries is challenging. People prefer to visualise the results of the structured sentiment tree in a concise and comprehensible way. More importantly, people want to know the reasons why people like and dislike those aspects represented in the tree.

In this chapter, we study the problem of extracting a sentiment tree from opinions expressed by people in social media texts. We present a structured sentiment analysis (SSA) approach which automatically extracts the hierarchical structure of hot aspects as well as the people's opinions towards those aspects. *hot aspects* can be defined as the most mentioned aspects people talk about. The input for the SSA is a collection of short messages about a particular product. A hierarchical process based on a topic model is proposed to capture the hidden relationships between aspects and extract the *hot aspects*. The outcome of the SSA is a sentiment tree where the root is the most general aspects of a product, and as the depth increases, the aspects become more specific. Each node in the tree represents the name of an aspect, along with a set of messages relevant to this aspect and its polarity.

The three challenges mentioned above can then be dealt with by using the proposed new SSA as follows.

- First, we propose a hierarchical approach to extract hot aspects and identify relationships between aspects simultaneously. The aspects on $(i-1)^{th}$ level are used to extract

the aspects on $i^{th}$ level. In this way, the weak semantic relationship between aspects preserved from the short messages can be identified.

- Second, we propose a hierarchical sentiment approach to attach people' attitudes for each nodes in the tree. To identify people' opinions, our approach performs a polarity classification for each message, followed by hierarchical statistics. On other words, the polarity of an aspect is determined by including the sentiment polarity for the aspect itself and its children in the hierarchical tree.
- Third, we proposed a summarisation approach to effectively provide a comprehensive summary and explanation for the extracted results.
- Fourth, the experiment results show that the proposed approach is effective for analysing social media texts and extracting the sentiment tree.

## 4.3 Problem Definition

Given a set of messages $M = \{m_1, m_2, ..., m_n\}$ about a specific product that a user is interested in, where $n$ is the number of messages, our task is to extract hot aspects $A = \{a_1, a_2, ...\}$ from $M$, where $a_i$ contains a subset of messages $M_i$ from $M$ talking about the $i^{th}$ aspect, and the most frequent words $TW_i = \{tw_{i1}, tw_{i2}, ...\}$ within $M_i$. The top-1 word $tw_{i1}$ is used as the name for the $i^{th}$ aspect. *Root* is one special aspect to represent the whole product. *Root* contains all messages $M$. Then, we construct a tree $T = \{t_1, t_2, ...\}$ where $t = \{i, j\}$ is a 2-tuple to indicate that aspect $a_i$ is the parent of aspect $a_j$, where $a_i, a_j \in A$. Our second task is to analyse people's opinions on those aspects discovered by tree $T$. The output is $O = \{o_1, o_2, ..., \}$, where $o_i \in [0, 1]$ is the score for the people's opinions towards aspect $a_i$. 0 means the absolute negative attitude while 1 means the absolute positive attitude.

## 4.4 Structured Sentiment Analysis

At the high level, our framework constructs a hierarchical tree of the most frequently mentioned aspects of a product with the corresponding sentiment polarity of those aspects. Figure 4.1 illustrates an architectural overview of the proposed system. The proposed system has four main components: (1) data pre-processing; (2) hierarchical extraction; (3) sentiment analysis; and (4) summary generation. The following sub-sections explain the four components in detail.

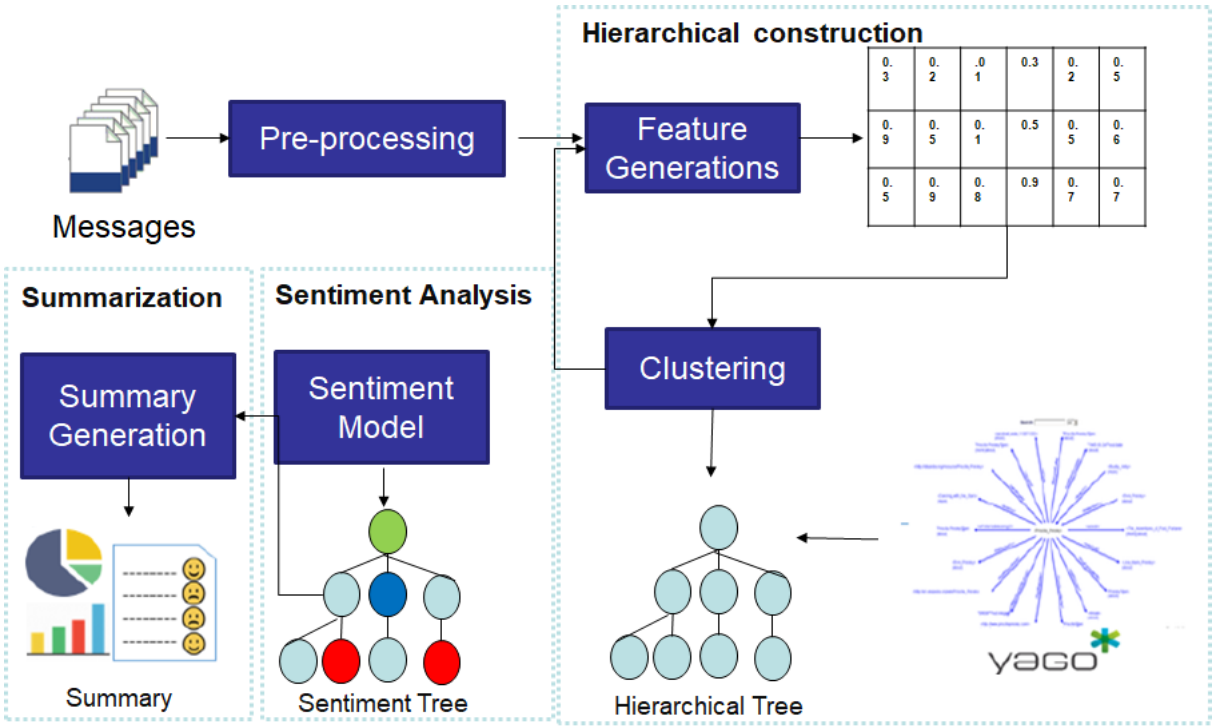



**Figure 4.1:** System Architecture of SSA.

### 4.4.1 Data Pre-processing

Data pre-processing is an important step as the quality of the hierarchy depends on the output of this step. To reduce the noise and improve the result of the SSA, we first pre-process the data to ignore common words that carry less important meaning. In this step, stop words, non-English characters and URLs are removed from the texts. Based on our observation, we noticed that most messages containing URLs are either spam or advertisements, and including them would produce irreverent information and noise to the tree. Finally, to construct the structured tree, we use the *part-of-speech tagging (POS)*[1] to extract *proper nouns*. We use only nouns and nouns phrase to extract the hot aspects since people often use *nouns* to refer to aspects of a product.

### 4.4.2 Hierarchical Extraction

The problem that we address in this section is how to construct a tree-structured representation of the *hot aspects* that most people care about from social media texts. The hierarchical tree shows the most frequent and general aspects close to the root, while the specific ones appear nearer the leaf nodes. It is important to mention that our hierarchical component can extract a hierarchical tree of hot aspects with any number of depth. The input of this compo-

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

nent is a set of messages about specific product. Our aim to find the relationships between nouns that often appear together in the same context and to extract a tree of *hot aspects*.

Our hierarchical extraction function consists of three main components: the feature generation, the recursive clustering and merge operation [33]. First, feature generation aims at extracting words (aspects) that often appear together and put them in the same node. Second, the recursive clustering is responsible for hierarchically organizing those aspects from general to specific. Third, the goal of merging operation is to filter irrelevant aspects and group the duplicated ones. In the following subsections, we discuss these steps in turn.

**Feature Generation**

The goal of this step is to transform text data into a feature representation that can best show the interests of users who post about the product' aspects. In order to find the best feature representation of the short messages, we compare four different methods (i.e, TFIDF[85], Smooth-TFIDF [70], LSA[23] and LDA[14]). For the feature generation process, the experiments perform well when using LDA technique on the short messages for feature extraction. The LDA is a Bayesian probabilistic model, which views each message as a mixture of underlying topics where each message is assigned to a set of topics via LDA. A topic model such as LDA is useful in our task to discover the hidden patterns in a text. In other words, it allows us to find terms that often appear together and are put similar words (e.g., synonyms) in the same topic. The input of the feature generation are the messages, and the number of topics is $k$ specified by the user. We calculate the weight $d_{i,k}$ of document $i$ in topic $k$ using:

$$f_{i,t,k} = \frac{n_{i,k} + \alpha_k}{\sum_{j=1}^{K} n_{i,j} + \alpha_j} \times \frac{n_{j,k} + \beta_k}{\sum_{x=1}^{K} n_{x,j} + \beta_j} \tag{4.4.1}$$

where $n_{i,k}$ is the number of times topic $k$ appears in document $i$, $n_{j,k}$ is the number of times word $j$ appears in the topic $k$. $\alpha$ *beta* are hyperparameter control the document-topic distributions topic-word distributions. The documents that talk about the same aspects are assigned to the same topic in the feature space. The output is two metrics document-topic representation $f_{i,k}$ (left part of Equation 4.4.1) topic-word representation $f_{t,k}$ (right part of Equation 4.4.1). In the next section, we only need the document-topic representation $f_{i,k}$ to cluster social media messages.

**Messages Clustering**

The problem that we address in this section is how to hierarchically group social media messages and find the hot product aspects that appear in the social media. *Hot aspects* can be defined as a list of the product' features that people care about.

We have a set of messages of messages $M = \{m_1, m_2, ..., m_n\}$ and it's feature representation $f_{i,k}$ extracted from previous step, we aim to automatically group those messaged who share the same content into clusters using cosine similarity function $C = \{c_1, c_2, ..., c_k\}$. The cosine similarity [81] function is used to cluster short text messages:

$$DocSim(M, C) = \frac{\sum_i (f_{m,i}, \times f_{c,i})}{\sqrt{\sum_j f_{m,j}^2} \times \sqrt{\sum_j f_{c,j}^2}} \tag{4.4.2}$$

where $m$ is a short message, c is a cluster centeroid, and $f_{m,i}$ is the feature representation of the message. The output of this step is a set of clusters $C = \{c_1, c_2, ..., c_k\}$. A cluster $c_i$ is a candidate of hot aspects $A$; the same as our definition for aspect $a_i$, a cluster $c_i$ contains messages $M_i$ belonging to this cluster, the most frequent words $TW_i$ from $M_i$ and the representative word $RT$ of $c_i$. In our model, only the top five words for each cluster are kept for sentiment analysis (section 4.4.3). The representative word of the cluster is represented by the most frequent word. The two examples below show the top five words for each cluster with the representative word.

**Camera**: *"camera, selfie, pic, picture, quality".*

**Audio**: *"audio, headphone, adapter, earphone, headset".*

To create the hierarchy of hot aspects, our system performs the same process again for each cluster generated on level $Lev$ to create aspects in the $Lev + 1$ level of the tree. Algorithm 4.1 shows the hierarchical extraction of SSA. Not all clusters generated by our system are useful and relevant. We add a filtering and merging to enhance the hierarchical structure of the aspects.

**Algorithm 4.1** Hierarchical Extraction for SSA.

**Data:** M : list of shot messages, k: number or nodes and Th:threshold

**Result:** Topic Hierarchy T.

$f_{i,t,k}$= transform text data to feature space using Eq.4.4.1.

**Function** *Recursive (M,k,Lev)*

  **while** $m \in M$ **do**

    *DocSim= assign the message to the most similar cluster using Eq.4.4.1.*

  **end**

  **for** *all cluster* $i \in C$ *in level* $j$ **do**

    $M_i = getm$ *messages assigned to cluster* $i$

    *getTopWords*($c_i$,5)

    *fr=getRWF*($c_i$)*= get frequency of RW of cluster i.*

    **if** $fr > Th$ **then**

      $T.add(c_i, Lev)$ *add the cluster to the tree in level Lev.*

      ***Recursive***$(M_i, k, Lev + 1)$*;*

    **end**

    **else**

      ***Stop;***

    **end**

  **end**

filter irrelevant clusters using YAGO.

merge similar cluster using Eq. 4.4.3

**return** T

### Filtering and Merging Operation

The results of the previous steps may produce an incomplete or incorrect tree. Filter steps aims to enhance the output of previous stage by keeping only *hot aspects* and removing outliers. To achieve that, we firstly assume high frequent aspects as hot aspects of a product. To tackle this problem, our algorithm can filter those outliers which are not related as hot aspects based on the term frequency. In other words, the cluster will be eliminated if its top word is lower than a specified threshold. In our experiment, we set the threshold to 0.01 on the first level. The threshold means that the frequency of its top-1 word should appear in the whole message above one percent.

Moreover, we use a YAGO ontology [87] to enhance the hierarchical representation of the aspects by providing another level if necessary. YAGO is an available public resource for all products as long as the product has a Wikipedia page. An example of an incomplete tree is when our system might produce two children, namely *jet* and *matte* under the node *black*. In this case, we use the ontology to improve the structure by adding another level to represent the node *black* under *colour*. YAGO is also useful for reorganising nodes that are incorrectly placed on the tree. In other words, if the results of our system shows *gold* under *black*, the ontology can help put the node on the correct branch. It is important to mention that our tree differs from YAGO from three perspectives. First, it is easy to obtain the physical aspects of a product from YAGO or other resources; however, some of aspects of the product are not important to people. Therefore, our system extracts only the hot aspects that people care about based on the microblog messages. Another difference in our system is that some emerging aspects about the product are mentioned by people in the microblog, but do not exist in YAGO. Third, our system provides users with people's feelings towards these aspects represented on the tree, which is not included in the YAGO ontology.

Additionally, the previous component may produce duplicate clusters. In order to reduce the number of duplicate clusters, we add a merging step which combines the redundant clusters into the same cluster if they share the same name. To achieve that, we calculate the overlap score of two clusters $C_i$ and $C_j$ using the Jaccard similarity function:

$$overlap(C_i, C_j) = \frac{|C_i \cap C_j|}{|C_i \cup C_j|} \tag{4.4.3}$$

where $C_i$ is the top words for cluster $c_i$ and $C_j$ is the top words for cluster $c_j$.

### 4.4.3 Sentiment Analysis

Combining sentiment analysis with hierarchy construction can effectively help to perform a fine grain sentiment analysis on the aspects extracted in the tree. This goal of this step is to hierarchically analyse and classify people's opinions about those extracted aspects into positive, negative and neural. Our sentiment method differs from others as it hierarchically extract the sentiment of aspects. To express an opinion on an aspect of a product, people often use opinion words *adjectives* and *verbs*. Consider the the following messages *"I love the quality of the camera"* and *"The new cell phone battery is amazing"*. The user uses the verb *love* as an opinion word to express their feeling about the *quality of the camera*. On the second example, the adjective *amazing* is used to express the user's opinion towards the aspect

*battery*. In our reseach we use opinion words to identify the polarity of aspects.

In the same message, people often mention irrelevant aspects of a product. Consider the following example: *"Iphone camera extremely impressive, Instagram linking crashing"*. Sentence-level classification approaches may classify the message as negative. While, in fact, the product aspect of the "camera" should receive a positive attitude. Therefore, in our approach, we only consider the opinion words (verbs, adjectives) towards the product's aspects to do sentiment analysis. In the above example, the irrelevant aspect, "Instagram", will be ignored. Then, just the opinion word "impressive" will be considered as it is closer than the other opinion word "crashing" to the aspect word "camera".

Our method extracts the semantic orientation of hot aspects in a hierarchical way using three distinct steps. The input to our algorithm is the hierarchical tree $T$ created from the previous stage. For sentiment analysis, we first retrieve the original messages $M_i$ for each aspect $a_i$. Then, we remove the noise, irrelevant information, from the messages as we did in section 4.4.1. After that, we tokenise the message and then perform the POS processing to assign parts of speech to words in each message. Next, all the words are stemmed to the original form by using *Lucense java API*[1]

After annotating the retrieved messages for each aspect $a_i$, the sentiment analysis phase is conducted as follows:

- Start from the leaf nodes $C_i$:
  - Get top words $TW_i$ which represent $C_i$.
- For each messages $m_i$ assigned to cluster $C_i$:
  - Search the opinion words in each message in $M_i$ that are closest to any top word in $TW_i$.
  - Identify the polarity of target using opinion lexicon and a swear list [3].
  - To deal with negation (e.g. not, no and never) , we flip the polarity returned from the lexicon.

The polarity of the message is positive or negative based on the results returned from the opinion lexicon. If the result returned from the lexicon is empty, then the swear list will be used. If the closet opinion word is found in the swear list, it will be identified as negative; otherwise, the message is classified as neutral. To deal with negations, we fillip the polarity from the opinion lexicon. For instance, the polarity of "is not good" is turned into a *negative*.

Once the sentiment polarity for each message in $M_i$ for aspect $a_i$ is extracted, our algorithm takes the sentiment analysis result to hierarchically draw the final polarity for each aspect $a_i$ in $T$. It is clear that each aspect $a_i$ is, in fact, a node in a tree. The polarity of a node is determined by including sentiment polarity for the node itself and its children. Because we split messages into aspects on each level in a hierarchical way, the direct children are enough to cover the whole branch rooted by the aspect. The polarity score $o_i$ of the aspect $a_i$ in $T$ is calculated as follows:

2

$$\tilde{P}_i = \sum_j P_j, j = i | j \in Children_i \tag{4.4.4}$$

$$\tilde{N}_i = \sum_j N_j, j = i | j \in Children_i \tag{4.4.5}$$

$$o_i = \tilde{P}_i / (\tilde{P}_i + \tilde{N}_i) \tag{4.4.6}$$

where $P_i$ is the number of positive messages from $M_i$ for aspect $a_i$, $\tilde{P}_i$ is the number of all positive messages for the aspect $a_i$ itself and all its children defined by $Children_i = \{j\}, \forall t \in T, t = \{i, j\}$. $N_i$ is the number of negative message from $M_i$ and $\tilde{N}_i$ is the number of all negative messages from itself and its children.

The final output of SSA is a tree of hot aspects of a product as well as the people's opinions about these aspects. Figure 4.2 shows the results of the tree construction and sentiment analysis.

### 4.4.4 Summarisation

Once we construct a structured sentiment tree, generating a structured summary of the sentiment tree is critical to help people better understand and interpret the sentiments about the product and its aspects. More specifically, a structured summary can help answer three questions: 1) What is the overall polarity of the product? 2) What are the most favourable and unfavourable aspects of the product? and 3) Why do people like or dislike those aspects?. For summary generation, the input of our algorithm is the sentiment tree. The output is three visualisation forms of the sentiment tree. First of all, our summary component generates a chart to show the final polarity of a product using the results of the sentiment

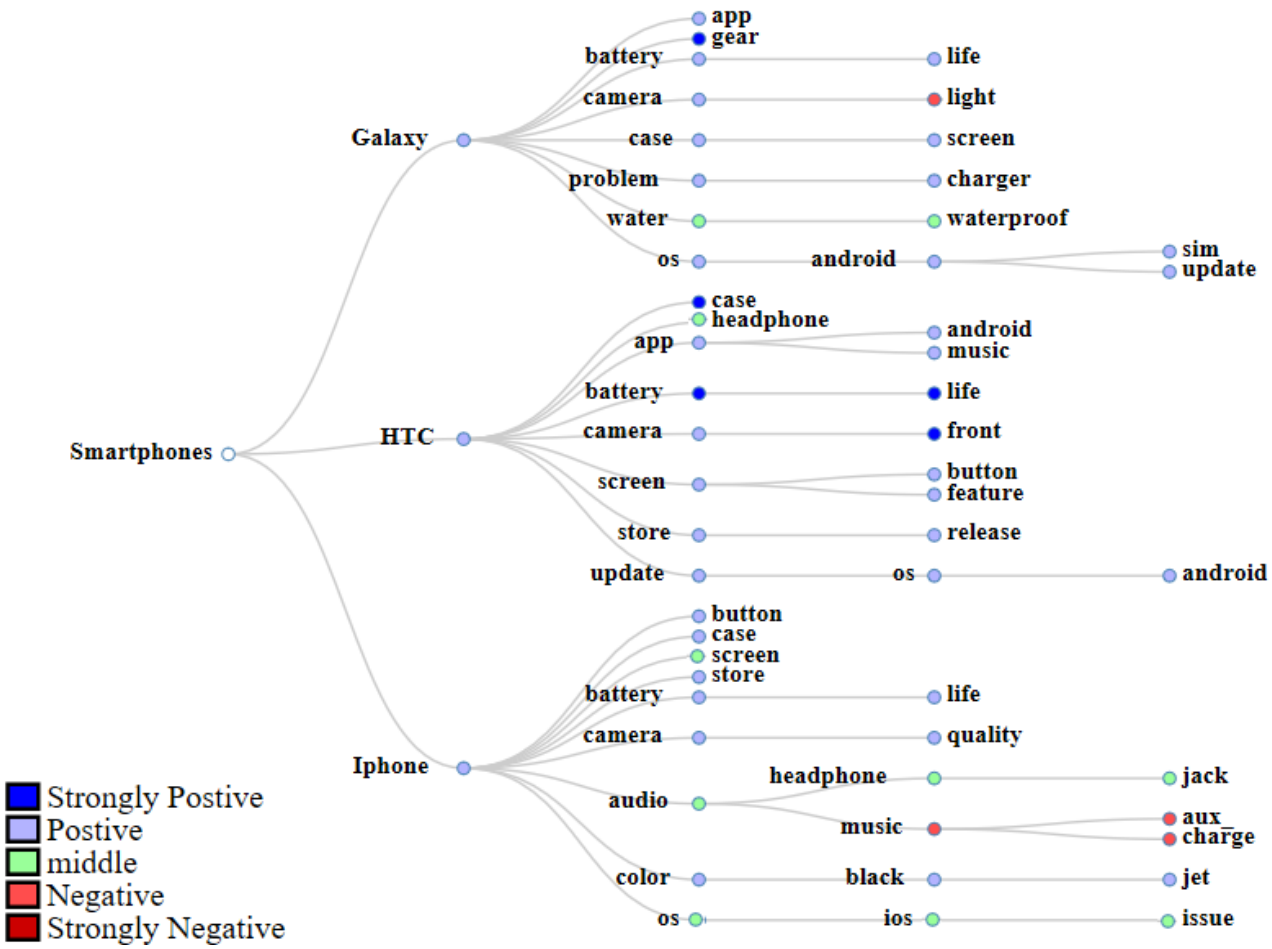



**Figure 4.2:** A structured sentiment tree of the three products.

tree. The polarity of the product is the polarity of all *Root* aspects. At any level of the tree, our system also provides an evidence of why people like or dislike certain aspects of a product by providing additional details from each message related to the aspect such as (e.g., message id, message date, text and polarity). Finally, our system discovers the top aspects that people like and dislike about a product. To achieve this, the system extracts the $X$ top $a_i$ which received the most positive and negative orientations from people.

## 4.5 Experiments and Evaluation

In this section, we first describe the settings of experiments and then demonstrate the experimental results.

### 4.5.1 Dataset and Experimental Stetting

In order to evaluate our SSA model, we crawled Twitter for three brands of cell phone by specifying the *hashtag*. The products are the *IPhone*, the *Galaxy* and the *HTC*. We selected

**Table 4.1:** Statistics used in the experiments

| Product Name | No. Tweets | No. Aspects (Nouns) |
|---|---|---|
| IPhone | 68004 | 6234 |
| Galaxy | 190494 | 6069 |
| HTC | 60895 | 2451 |

these smartphones because they are the most talked about products on Twitter. Our collection spans from the release data of the smartphone to January 2017. See Table 4.1. To extract the hierarchical tree, we set the LDA model hyperparameters to $iteration$=2000, $alpha$=0.1, $eta$=0.01, the number of topics to $k$=[25,2] ,and the threshold $ThPercentage$=[0.01,0.05]. Figure 4.2 and 4.3 shows the result of our model.

### 4.5.2 Visualization

We design SSA to produce a hierarchical structure of hot aspects such that the hierarchy can be easily summarise the aspects of the product with the corresponding opinions at any level of granularity that the user needs. Figure 4.2 shows a part of the discovered hierarchy for three smarphones. As we can see, the SSA extracts the aspects of the product from general closes to the root node to specific nears the leaf nodes. For example, the aspects Iphone $\rightarrow$ camera $\rightarrow$ quality are organised from general to specific. As advantage of SSA, the users can easily compare two or more similar products or services with different properties. This allows them them to make an informed decision to buy the product. Another advantages of our model is that it produce a simple summary and explanation why people like or dislike those aspects in the tree. In Figure 4.4, we illustrate the reason behind people opinions through listing the information about: tweet, sentiment polarity of the tweet and date. Those information can help users understand why other users is positive, negative or neural about the aspects of the product. The charts shown in Figure 4.4 displays a details summary about: 1) What is the overall polarity of the product? 2) What are the most favourable and unfavourable aspects of the product? and 3) What is the polarity of each aspect in the tree.

### 4.5.3 Hierarchy Analysis

The purpose of the evaluation is to measure the consistency and the quality of the SSA output. Since there is no prior work has been done to reveal the structure from social media

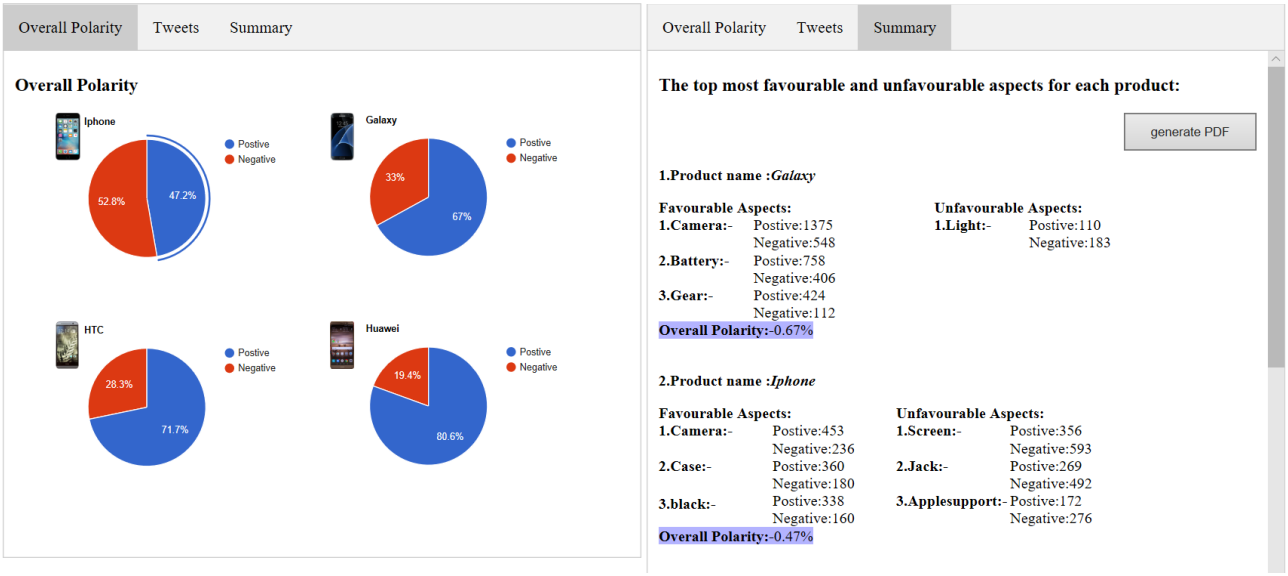



**Figure 4.3:** A details summary of the three products.

texts, a comprehensive comparison is difficult. In order to quantitatively evaluate the SSA result, we use parent-children relatedness and an online survey. First, parent-children relatedness is used to evaluate the semantic relationships between parent node and child node. Additionally, to measure the quality of the tree, we conducted an online survey. The goal of the survey was to use people's experience to evaluate three major characteristics of the hierarchical tree: node specialisation, uselessness and aspect-sentiment accuracy. We recruited 115 participants who had experience with smartphones. In this experiment, 66 were IPhone users, 34 were Galaxy users, and only 15 were HTC users, due to the differing popularities of the three brands.

### Parent-Children relatedness

An important characteristic of the hierarchical tree is parent-children relatedness, which means that parent nodes are supposed to be more similar to their direct children than others. Therefore, our goal in this section is to evaluate the relationships between the parent and its children.

In this experiment, we computed the relatedness score of SSA and compare it to two state-of-the-art methods: nCRP and rCRP. The relatedness score is not compared with HASM model since the output of this model is a flat structure. We use cosine similarity to compute the distance between the two aspects $a_i$ and $a_j$:

$$consin(\phi_i, \phi_j) = \frac{\phi_i \cdot \phi_j}{|\phi_i||\phi_j|} \tag{4.5.1}$$

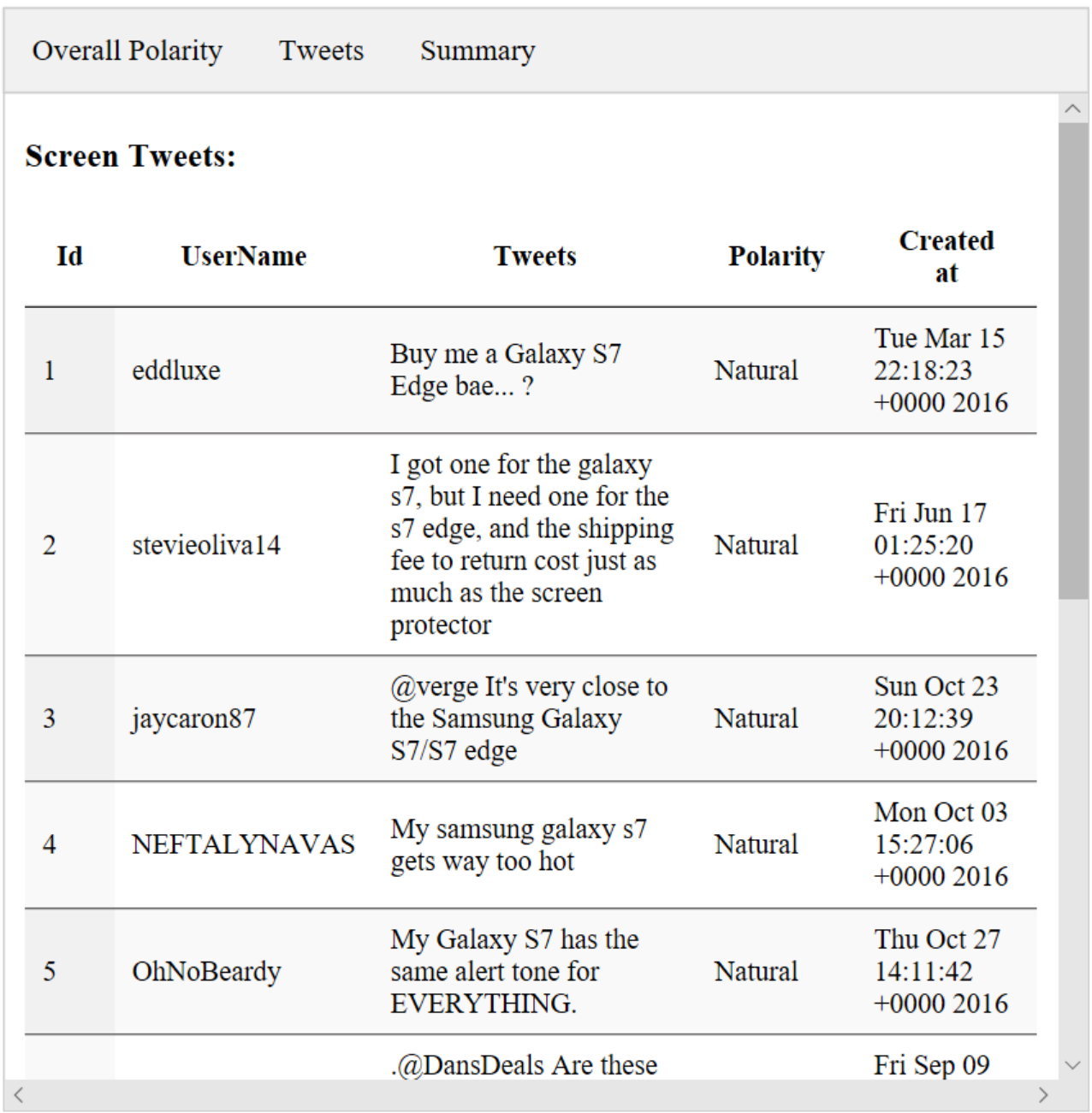



**Figure 4.4:** Information and explanation for people opinions.

where $\phi_i$ is the distribution of the word $tw_{i1}$ over all $k$ topics discovered via the LDA. The distance of aspect $a_i$ and its direct children are calculated with:

$$\triangle(i) = \frac{\sum_j consin(\phi_i, \phi_j)}{|Children_i|}, j \in Children_i \tag{4.5.2}$$

$$\nabla(i) = \frac{\sum_j consin(\phi_i, \phi_j)}{|A| - |Children_i| - 1}, j \neq i, \notin Children_i \tag{4.5.3}$$

For each parent node $\phi_i$, we compute the average cosine distance to its direct children, and then compare the average distance to non-children nodes. We average the result from children and non-children to all parent nodes for level 1. Figure 4.5 shows the parent-children relatedness score for all three phones. The comparison shows that SSA achieved better relatedness score compared with rCRP and nCRP.

**Hierarchy Quality**

We designed the SSA to construct a structured sentiment tree of hot aspects of a product. One important feature of our model is to hierarchically organise the product aspects from general to specific. To quantitatively evaluate node specialisation, we used the survey to ask the participants if the aspects were organised in a hierarchical way in the tree's structure. Figure 4.6 shows the percentage of people who agreed that the product's aspects were organised from general to specific on the tree. Overall, for all products, the results indicate that the participants were extremely satisfied with the tree's organisation.

Another important characteristic we aimed to measure was the usefulness of our automatically extracted tree, which consisted of product aspects. In our surveys, smartphone users were asked how relevant the discovered aspects were to the product on a scale of 1-5 . The rating scale was as follows: 1 (totally not relevant aspects of the given product); 2 (slightly not relevant aspects of the given product); 3 (middle); 4 (slightly relevant aspects of the given product); 5 (totally relevant aspects of the given product). Figure 4.6 shows the score for each product. For IPhone users, the results illustrate that approximately 82% agreed that the product aspects were relevant and made sense. However, for the Galaxy and HTC, about 75% of the participants felt the quality of the structure was acceptable.

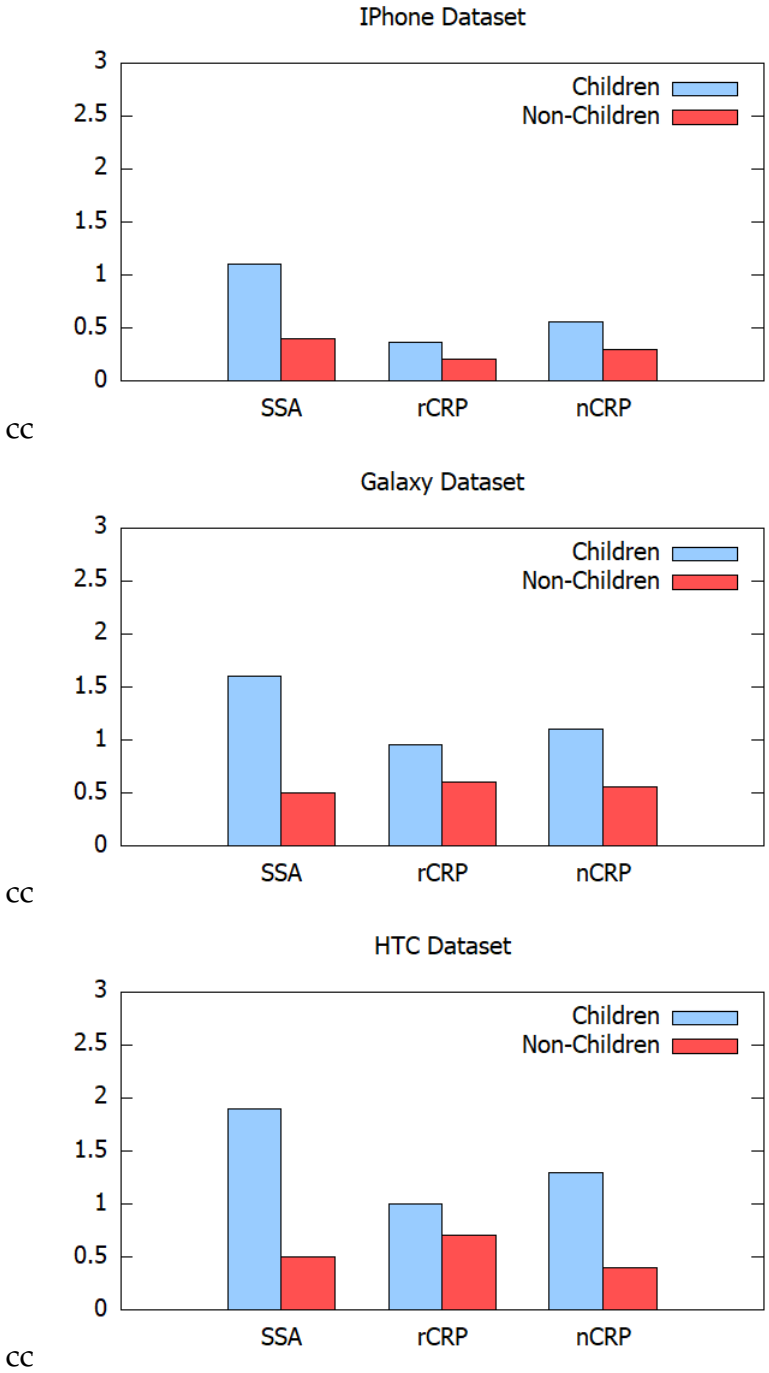



**Figure 4.5:** Parent-Children Relatedness. A high distance indicates that the parent is similar to its children. For all datasets, SSA shows the parent nodes are related to it's direct children nodes than non-children nodes. A higher score means that the parent concept are more similar. For all datasets, CCM shows higher parent-children relatedness compared with the other methods

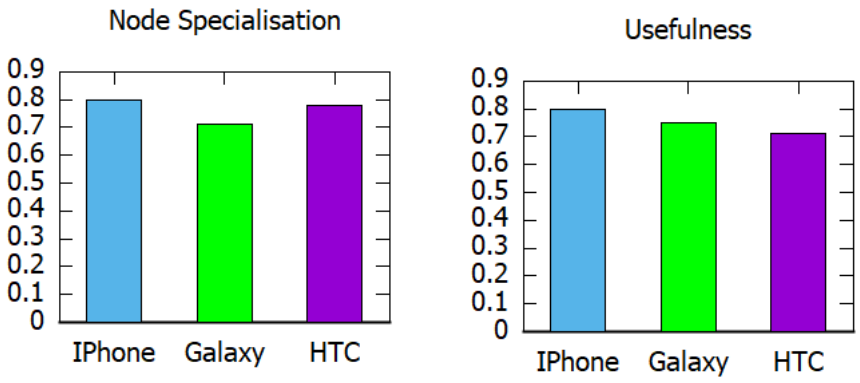



**Figure 4.6:** Hierarchy Quality.

### Aspect-Sentiment Accuracy

In addition to extracting the concept hierarchy of the hot aspects, we designed our system to show people's opinions on those aspects. To evaluate the accuracy of the sentiment analysis results, we conducted a survey to ask the participants two questions: 1) What aspects are positive about the smartphone? and 2) What aspects are negative about the smartphone? We assumed that the participants and Twitter users shared the same opinions about the hot aspects of the products given in the survey.

In this experiment, we compared the score of the respondents with the score of our model with:

$$Accuracy = \sum_{i=1} |(p_i - q_i)| \frac{l_i}{L} \tag{4.5.4}$$

where $o_i$ is the polarity score from our model for aspect $a_i$, and $q_i$ is the polarity score from the respondents for aspect $a_i$. $q_i$ is a ratio between the number of respondents who selected the aspect $a_i$ as positive and the number who selected $a_i$ as positive or negative. Since some aspects receive more responses than others for sentiment analysis, we believe the more responses an aspect receives, the higher the confidence level of people's attitudes towards this aspect. Thus, we added a weight for each aspect to calculate the final sentiment accuracy. The more responses for an aspect, the higher the weight of the aspect. $L$ is the total number of respondents for all aspects, and $l_i$ is the number of respondents who selected $a_i$ as positive or negative. Figure 4.7 shows the aspect sentiment accuracy of our model for all products. It is obvious that for all smartphones, the results indicate consistency between the sentiment result of our model and the participants' opinions.

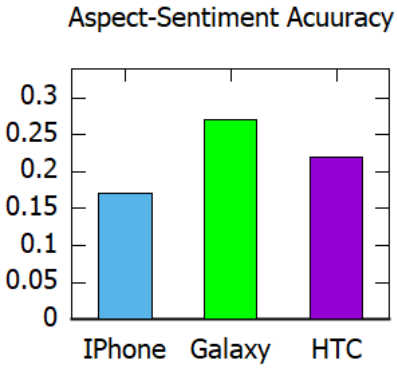



**Figure 4.7:** Aspect-Sentiment Accuracy. For all three smartphones, a lower score indicates that the results of the SSA model are more similar to the survey responses.

**Table 4.2:** Average running time in seconds.

| | **IPhone** | **Galaxy** | **HTC** |
|---|---|---|---|
| **SSA** | 410.6 | 620.5 | 460.9 |
| **HASM** | 563.6 | 741.9 | 590.1 |
| **rCRP** | 493.4 | 690.1 | 490.4 |
| **nCRP** | 483.2 | 680.6 | 479.0 |

### Computation Time

To give a comprehensive result of the performance of SSA, in this section, we report the execution time of the SSA and compare it with the other models. In this experiment, the computer that has been used had the following features: Processor Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz, 16GB Memory and Windows 10. We present the average execution time over each dataset. The execution time results are shown in Table 4.2. The results show that SSA consumes less time than the other models. The extra running time for other models comes from the Gibbs sampling. On other words, those models need a number of iteration to discover the concept hierarchy.

## 4.6 Summery

We present in this chapter a structured sentiment analysis approach (SSA) to analyse the opinions that people express about aspects of a particular product in social media. Combining sentiment analysis with hierarchy construction can effectively help to perform a fine-grained sentiment analysis on the concepts extracted in the tree. The proposed approach first discovers the structured tree of the hot aspects of a product in a hierarchical way. Then,

the corresponding sentiment analysis on those aspects identified in the tree is performed. As an advantage of SSA, it can help companies to understand the feelings of consumers towards their products or services at different levels of granularity. Finally, our approach summarises people's attitudes . The results confirm the effectiveness of our proposed approach on analysing social media messages. In future, our goal is to improve our methodology to extract more relative aspects of products. We will also improve the proposed approach to create a more specific structured hierarchical of the product.

CHAPTER 5

# Modelling User Attitudes Using Hierarchical Sentiment-Topic Model

## 5.1 Overview

Uncovering the latent structure of various hot discussed topics and corresponding sentiments from different social media user groups (e.g., Twitter) is critical for helping organizations and governments understand how users think about their services, facilities, and things that are happening around them. Although numerous research texts explore sentiment analysis on different aspects of a product, fewer works focus on why users like or dislike those products. In this chapter, a novel probabilistic model is proposed, namely the Hierarchical User Sentiment Topic Model (HUSTM), which discovers the hidden structure of topics and users and performs sentiment analysis in a unified way. The goal of the HUSTM is to hierarchically model the users' attitudes (opinions) using different topic and sentiment information, including the positive, negative, and neutral. The HUSTM (e.g., width and depth) is inferred from data in an unsupervised manner. The model is evaluated on three real-world datasets. The qualitative evaluations confirm the high quality of the hierarchy discovered by the HUSTM model in comparison to those obtained using state-of-the-art methods.

## 5.2 Background

The construction of a hierarchical tree of topics and user's interests from a social media platform is an interesting and significant problem. On social media platforms, such as Twitter and Weibo, a user often expresses opinions on various aspects of a product, such as overall

design, battery capacity, screen size, and camera. A high-quality model of user interests at different levels of granularity has many valuable applications in the areas of summarization, search and browsing. With such a model, a user could quickly compare two or more smartphones on different granularities by looking at the hierarchy. Individuals could also find users who shared identical opinions, recommending interests to them. For organizations, hierarchically modelling the attitudes or interests of users can give insight into user interests with respect to a variety of topics and help analysing user' behaviours, locating influential users at any granularity level by using their sentiment information.

In recent years, hierarchical topic model research has focused on identifying a hierarchical tree of topics from documents [71, 66, 92, 40, 106]. Blei et al. [15] introduced the nested Chinese Restaurant Process (nCRP) to hierarchically discover structures from data. The depth and the number of child-topics in this model are manually specified. Kim et al. [45] suggested a new model based on the nCRP, namely, the recursive Chinese Restaurant Process (rCRP). In the rCRP model, the nCRP model is extended further to create a hierarchical tree where each document has a distribution over all the topics in the tree. Kim et al. [47] proposed a novel approach through the Hierarchical Aspect Sentiment Model (HASM), which applied a Bayesian non-parametric model to infer and learn the structure and sentiment from online reviews. Although these models mentioned above have shown great performance in different domains, the current research dose not consider modeling a user's interest at different granularities. In fact, users who share the same opinion and topic should be hierarchically grouped together in the different group in the tree. Moreover, the current literature only identified topics or user's interests if a user mentioned such topics frequently, ignoring the sentiment trend on any given topic.

The above-mentioned methods only extracted their topic hierarchies among the topics, without considering sentiment information and the users' interests on those topics. Another disadvantage with those models is that they were proposed to deal only with long text; applying them to short texts can lead to an incomplete or flat tree. To address these problems, this chapter proposes a novel model called the Hierarchical User Sentiment Topic Model (HUSTM), which extends on the HASM by adding a user-sentiment layer that captures the users' interest topics with different sentiment information. The primary goal of the HUSTM is to discover users' attitudes and interests about different polar topics in the text hierarchy. In the HUSTM, the entire structure is a tree with each node in the tree separated further into two sub-nodes: (1) the topic-sentiment node, which models the word distribution over topic and sentiment (e.g., positive, negative, or neutral); and (2) the user-sentiment node,

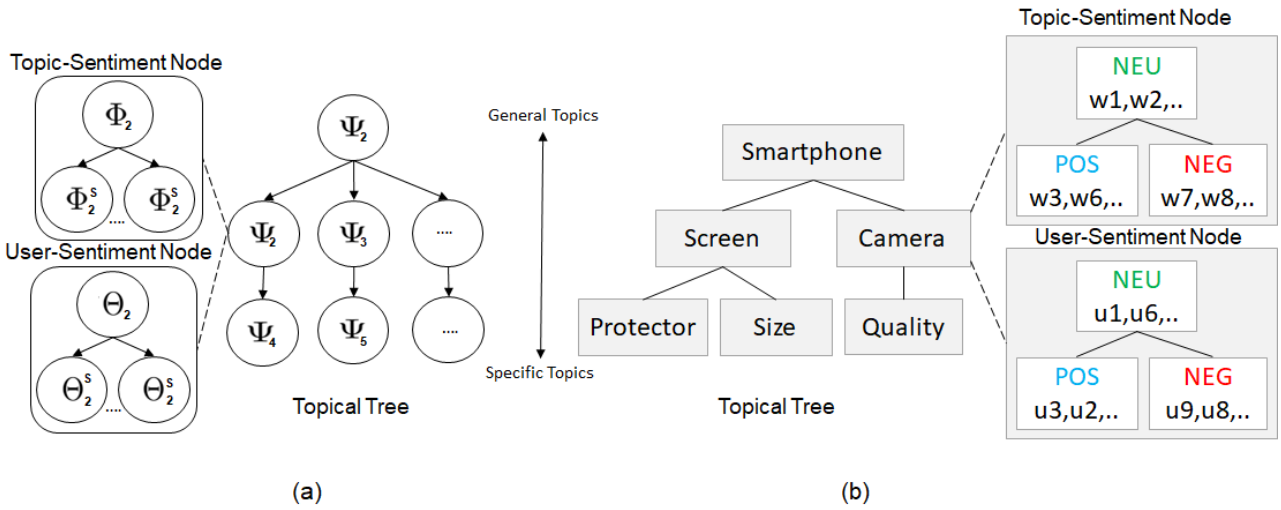



**Figure 5.1:** Topical structure of HUSTM. Each node in the tree is itself a two-level tree consisting of a topic-sentiment node and a user-sentiment node. Both nodes decompose into three sentiment polar topics: positive, negative and neutral. Each polar topic is distributed over words and users.

which captures the users' attitudes with respective sentiment information. The HUSTM incorporates the user-sentiment analysis into topic discovery to investigate the attitudes that users have towards the topics found in the tree. Figure 5.1 (b) shows a topic hierarchy run on a smartphone data set. We experimentally demonstrate the effectiveness of the proposed models in three data sets. The results show a high-quality topical hierarchy discovered by our model when compared with other methods.

The advantages of the proposed HUSTM over existing models are summarized as follows:

- It provides a unified model that discovers the hierarchical tree of topics, sentiments, and users.
- It infers the width and the depth of the tree from the data automatically.
- It discovers the topic hierarchy from both a short text and a long text by recursively modelling the words in an entire sentence.
- It allows for an estimation of the user's interest and sentiment towards topics to enhance the model accuracy.

## 5.3 Problem Definition

Below we introduce some problems of HUSTM and define the related concepts.

**Definition 1.** (Topical Tree). A topical tree is defined as $T$, where each node in the tree is a tree itself. Figure 5.1 (a) shows a magnified view to the proposed topical tree. As shown, each node $\Psi$ in the tree consists of a topic-sentiment node and a user-sentiment node. Figure 5.1 (a) shows the topical structure of HUSTM.

**Definition 2.** (Topic-Sentiment Node). A topic-sentiment node is semantically coherent them, which is represented by a multinomial distribution of the whole words in the vocabulary. Every words in the topic-sentiment node should be semantically related and refer to the same thing. Formally, a topic-sentiment node is defined as the $\Phi_k$ of the topic $k$, where each topic-sentiment node $\Phi_k$ is separated into $S$ sentiment polar topics (e.g., positive, negative and neutral) and denoted as $\Phi_{k,s}$.

**Definition 3.** (User-Sentiment Node). A user-sentiment node in the tree $T$ is defined as the $\Theta_k$ of the user topic $k$, where each user-sentiment node $\Theta_k$ is separated into $S$ sentiment polar topics, and denoted as $\Theta_{k,s}$. Each user-sentiment node is a multinomial distribution over whole users.

To help understand the above definitions, we provide the following example. In the micro-blogging site Twitter , users can post text of up to 280 characters to discuss any topic. In a topical tree, the topic-sentiment node can describe popular themes that users are interested in. The user-sentiment node is a group of users sharing a common opinion on the same topic and represents their interests. Figure 5.1 (b) shows an example of a topical hierarchy. In the figure, every node in the tree is a topic (such as camera, quality, or screen). Each topic is further decomposed into three sentiment polarities (e.g., positive, negative, or neutral) according to the sentiment polarities of the words. For instance, the positive topic includes the words *camera*, *love*, *nice* and *awesome*, while the negative topic contains *selfie*, *bad*, *quality* and *terrible*. Every words in the topic-sentiment node should be semantically coherent and refer to the same topic. For example, all words in the topic *camera*, *quality*, *front*, *cam* are related words. The users who share the same topic and opinion also are grouped in the same node.

**Problem 1.** A corpus is represented as collection of documents $N_d = \{d_1, d_2, ..., d_l\}$, where each document $d$ contains a sequence of words $N_w = \{w_1, w_2, ...w_n...\}$ and each word in a document has a unique index from a predefined vocabulary $V$. We also define $u_d$ to be a set of users who share the document $d$ and use $U_d$ to define the total number of users of the document $d$. Given the collection of documents and the user who share the same document, our task is to group the users into a hierarchy of different groups according to their opinions, and discover the coherent topics in those groups. The depth and the width of the tree is learned automatically from the data. In Section 4.2, the mechanism will be further

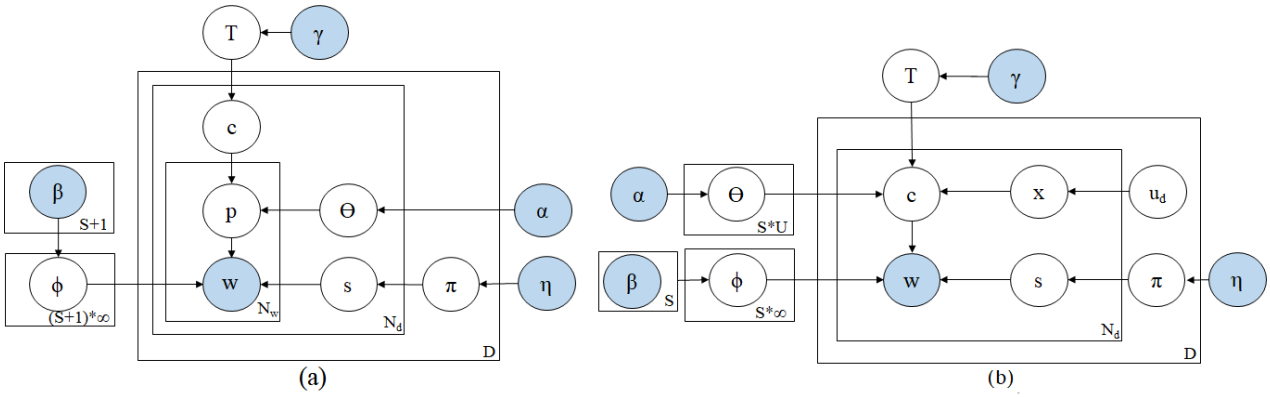



**Figure 5.2:** (a) Hierarchical Aspect-Sentiment Model (HASM). (b) The Hierarchical User Sentiment Topic Model (HUSTM).

explained.

According to [15, 5, 91, 45], to discover an optimal topic hierarchy from a text, there are three important criteria that must be considered:

- **Relatedness:** All root topics in a sub-hierarchy should not only be related to their direct children, but also to all offspring. For example, the root node *iPhone* should be related to its sub-topics *(camera, headphones)* and its sub-sub-topics *(picture, adapter)*.
- **Coverage:** All topics in a hierarchy should be organized from general to specific. The concepts near the root node must cover many documents, while those close to the leaf nodes should have lower coverage. For example, the parent topic "camera" has better coverage than its children in a hierarchy, *quality*, *selfie* and *front*.
- **Completeness:** All words in the same topic should be semantically related and refer to the same thing. For example, all words in the topic "camera, quality, picture" are related.

## 5.4 Hierarchical User-Sentiment Topic Model

Hierarchical topics models, such as the HASM, discover topic hierarchies based on document-level word co-occurrence patterns. Applying these models to short texts may produce incomprehensible and incorrect trees, since short texts possess very limited word co-occurrence information. Such models also fail to include the users' sentiment information, which is critical when extracting a more meaningful tree. To tackle these problems, the HUSTM is proposed, replacing the sentence-based layer with a word-based layer that allows the learning of a hierarchy from both short and long texts.

| Symbol | Description |
|---|---|
| $N_d$ | Number of documents . |
| $N_w$ | Number of word . |
| $U$ | Number of users . |
| $V$ | Number of unique words . |
| $S$ | Number of sentiment labels . |
| $x_{di}$ | Users associated with token i in document $d$. |
| $s_{di}$ | Sentiment label associated with token i in document $d$. |
| $w_{di}$ | The $i_{th}$ word token in document $d$. |
| $c$ | Topic-Sentiment node . |
| $T$ | Prior Tree . |
| $m_n$ | Number of words assigned to topic $n$. |
| $M_n$ | Number of words assigned to topic $n$ and its all subnodes. |
| $\pi$ | Document sentiment distribution. |
| $\Phi$ | Topic and sentiment polar topic . |
| $\Theta$ | User and sentiment polar topic. |
| $\alpha, \beta, \gamma, \eta$ | The fixed parameters of Dirichlet distribution priors. |

### 5.4.1 Generative Process

The key idea of the HUSTM is to discover users' interests about different polar topics from short texts and long texts using the hierarchy structure. More specifically, the HUSTM integrates the users' sentiment information to enhance its tree structure and detect the users' attitudes at different granularities. Each topic obtained by the HUSTM has a sentiment label and users have portability distribution over all sentiment polar topics in the tree. The hierarchy structure provides information about the topics that a group of users care about. Furthermore, the HUSTM not only captures users interest's (e.g., whether a topic is liked or disliked) but also shows the words that users give to describe their opinions.

The HUSTM is hierarchical generative mode. Each word in a document is associated with three latent variables: a user, a topic, and a sentiment label. In this model, each user is associated with a multinomial distribution over topics and sentiments $\Theta_{k,s,u}$, and each word is associated with a multinomial distribution over topics and sentiments $\Phi_{k,s,w}$. Conditioned on the users' distributions over topics and sentiment, the generative process for each docu-

ment can be summarized as follows: to begin, a prior tree is created randomly using rCRP from the documents. Then, for each document, a sentiment label is chosen from the sentiment distribution. Next, a user is sampled at random for each word token in the document. After that, a topic is chosen from the tree for each word associated with the user of that word. Finally, the words are sampled and conditioned on both their topic and sentiment label.

To automatically infer the depth and the width of the tree, we adopt the rCRP as a prior tree. The prior tree is defined as a tree generated before data observation. The shape of the prior tree T can be defined with an unbounded width and depth. After observing the data, the width and depth of the tree $T$ will be learned. The hyperparameter $\gamma$ controls the structure of the tree. A higher value of $\gamma$ increases the number of children nodes for each parent node in the tree, while a lower value produces a narrow and shallow prior tree. The formal procedure of Figure 5.2 (b) is explained as follows:

- Draw a tree $T$ with unbounded depth and width from rCRP prior $T \sim rCRP(\gamma)$.
- For each document $d$ in $D$, sample a sentiment distribution $\pi \sim Dirchlet(\eta)$
- For each word $i$ in document $N_d$
  - Sample a user $x_{di} \sim Uniform(u_d)$.
  - Sample a topic-sentiment node $c_{di} \sim T$.
  - Sample a sentiment $s \sim Multinomial(\pi)$.
  - Sample a word $w_{di} \sim Multinomial(\phi_{c,s})$.

For a better understanding of the uniqueness of the HUSTM, we offer a comparison between the HUSTM and the HASM. Figure 5.2 shows the model structure of the HUSTM in comparison to the HASM, and the relevant notations are listed in Table 5.1. The HASM is a non-parametric model for discovering a topic-sentiment hierarchy from online review data. As can be seen in Figure 5.2 (a), the HASM is a sentence-based model, which means that its sentiment analysis is achieved on a sentence-level rather than word-level, and that all of the words in a sentence are assigned to the same topic. Hence, when the documents are short, the HASM will fail to discover the hierarchy structure due to the limited number of words in the short text.

Replacing the HASM sentence layer with a word layer helps the HUSTM to effectively discover hierarchies from short and long texts. As described above, all words in a document

will be distributed across the whole tree randomly. Then, as described in Section 4.2, we will recursively observe the data to learn the correct topics. The modelling occurrence of a single word can reveal the topics better, enhancing the learning of topics. In our experiment, we demonstrate the advantages of our model compared with the other methods processing a short text.

**Algorithm 5.1** Sampling $k_{ix}$ by recursive algorithm.

**Function** *Recursive* ($\Psi_k$)

$next_k \sim m_k \times P(w_{di}|x_{di}, s, \Psi_k, \beta, \alpha)$

$next_k \sim M_k \times P(w_{di}|x_{di}, s, \Psi_k, \beta, \alpha)$

$next_k \sim \gamma \times P(w_{di}|x_{di}, s, \Psi_k, \beta, \alpha)$

**if** $next_k = \Psi_k$ **then**

  **return** $k$

**else if** $next_k = child\ of\ \Psi_k$ **then**

  $\Psi_c = next_k$

  **return** Recursive($\Psi_c$)

**else if** $next_k = new\ child\ of\ \Psi_k$ **then**

  $\Psi_n = next_k$

  **return** Recursive($\Psi_n$)

The total probability of the word, sentiment, and topic assignments of the entire corpus is:

$$\mathcal{L} = P(T|\gamma) \prod_{k=1}^{\infty} \prod_{s=1}^{S} P(\Phi_{s,k}|\beta_s) \prod_{d}^{D} P(\pi_d|\eta) \prod_{i}^{N_d} \prod_{x=1}^{U} P(w_{d,i,k}|s, c, \Phi) P(c_{di}|x_{di}, T) P(x_{di}|u_d) \tag{5.4.1}$$

### 5.4.2 Model Inference and Parameter Estimation

We use Gibbs sampling, as it provides a simple method for parameter estimation and posterior sampling. More specifically, the aim is to estimate the *posterior* of the tree and the posterior of only two groups needs to be inferred, topic and user sampling and sentiment sampling. Other variables are integrated out and do not need sampling.

**Topic and User Sampling.**

The algorithm starts by randomly assigning words to random topics, sentiments and users, based on the set of users in the document. The output of the random assignment is the *prior tree*. After generating the prior tree, our task is to observe the data and learn the structure of the tree. On other words, we aim at assigning each word and user to the correct node in the tree. To achieve that, we start from the root node of the prior tree and recursively move along the path. Each node on the tree contains statistics about the sentiment polarity of the words and the attitudes of the users. Algorithm 5.1 illustrates the recursive process for HUSTM. Formally speaking, for each word $i$ and user $x$ assigned to node $k$ in the tree, starting from the root node, we compute one of the following possibilities:

- P( Select the current node $\Psi_k$ ) $\propto m_k \times P(w_{di}|x_{di}, s, \Psi_k, \beta, \alpha)$
- P(Select a child $c$ of the current node $\Psi_k$ ) $\propto M_c \times P(w_{di}|x_{di}, s, \Psi_k, \beta, \alpha)$
- P(Create a new child under the current node $\Psi_k$ ) $\propto \gamma \times P(w_{di}|x_{di}, s, \Psi_k, \beta, \alpha)$

$$P(w_{di}|x_{di}, s, \Psi_k, \beta, \alpha) \propto \left( \frac{n_{i,k,s} + \beta}{\sum_{r=1}^{V} n_{w_r,k,s} + \beta * V} \times \frac{n_{j,k,s} + \alpha}{\sum_{u=1}^{U} n_{x_u,k,s} + \alpha * K} \right) \tag{5.4.2}$$

where $w_{di} = i$ and $x_{di} = j$ represent the assignments of the $i_{th}$ word in the document $d$ to the topic $k$ and the user $j$, respectively. The $n_{i,k,s}$ is the number of times the word $i$ is assigned to the topic $k$ and the sentiment label $s$. The $n_{j,k,s}$ is the number of times the user $j$ is assigned to the topic $k$ and the sentiment label $s$. We draw the topic and user for each word $i$ in the document $d$ as a block, conditioned on all other variables. The recursive process stops for each word and user if a node is chosen by the first or the third possibilities. Then, we can estimate the word topic distribution $\Phi_{k,w}$ and user topic distribution $\Theta_{k,u}$ by:

$$\Phi_k = \frac{n_{i,k,s} + \beta}{\sum_{r=1}^{V} n_{w_r,k,s} + \beta * V} \tag{5.4.3}$$

$$\Theta_k = \frac{n_{j,k,s} + \alpha}{\sum_{u=1}^{U} n_{x_u,k,s} + \alpha * K} \tag{5.4.4}$$

**Algorithm 5.2** Gibbs sampling algorithm for HUSTM.

**Data:** Prior Tree: randomly initialisation $\Phi$ and $\Theta$

**Result:** Topic Hierarchy T.

```
while not finished do
    for all documents d ∈ [1, D] do
        for all users x ∈ [1, U_d] do
            for all words i ∈ [1, N_d] do
                for the current assignment for word w_di and user x_di:
                decrement counts and sums recursively
                start from the root topic:
                sample a topic recursively using Eq.2:
                P(w_di | x_di, s, Φ_k, β, α)
                sample a sentiment label s using Eq.3:
                P(w_di = i, x_di = j | c, s, β)
                increment counts and sums recursively
            end
        end
    end
end
```

**Sentiment Sampling.**

Subsequently, the sentiment polarity of each word and user is sampled simultaneously. The probability of assigning the label $s$ for the word $i$ and the user $x$ in the document $d$ is:

$$P(w_{di} = i, x_{di} = j | c, s, \beta) \propto \left( \frac{n_{d,c,s} + \eta}{n_d + \eta * 3} \times \frac{n_{i,c,s} + \beta_s^i}{\sum_{r=1}^{V} n_{w_r,c,s} + \hat{\beta}_s} \right) \tag{5.4.5}$$

where $n_{d,c,s}$ is the number of words in the document $d$ that is assigned to a topic-sentiment node $c$. The $n_d$ is the total number of words in the document $d$. The $n_{i,c,s}$ is the number of times the word $i$ appears in the topic-sentiment node $c$. The $\eta$ hyperparameter controls the sentiment distribution for each document. After a number of iterations, the words and users will be assigned to the correct topic and sentiment. The final output is the posterior tree which is defined as a tree generated after data observation.

Note that the polar topic of the user depends on the sentiment polarity of the word. For


**Figure 5.3:** Sample output from HUSTM run on Smartphone dataset.

each word, the HUSTM identifies its polarities from the sentiment distribution of words in the document. To discriminate between different sentiment polarities, the sentiment lexicons from previous studies are incorporated as seed words [55]. In particular, the neutral, positive and negative priors are first assigned at 0.01 for all the seed words in the documents. The weights of a word in the Dirichlet priors for neutral, positive and negative topics are 0.01, 2.0 and 0.001, respectively. Algorithm 5.2 shows the sampling process of HUSTM.

## 5.5 Experiments

The experimental evaluation of the proposed model is performed according to the following three aspects: the coverage, the parent-child relatedness, and the topic-sentiment consistency. These experimental results show that our proposed model provides promising results.

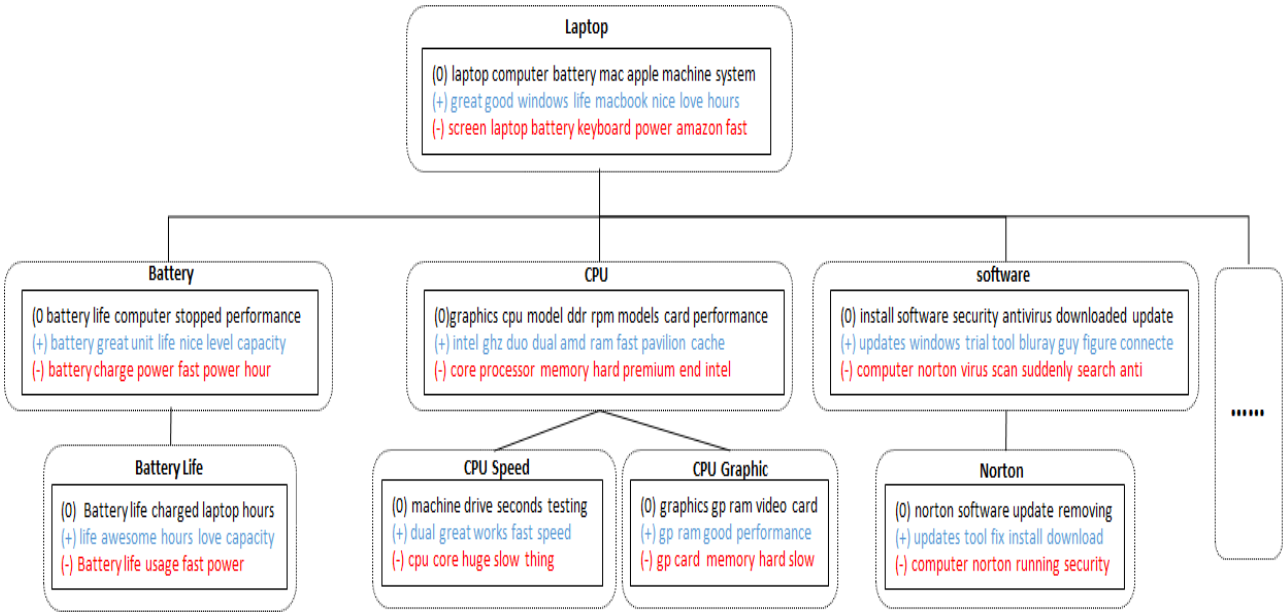



**Figure 5.4:** Sample output from HUSTM run on Laptop dataset.

### 5.5.1 Datasets

We conducted experiments on three real-world data sets. For the long texts, experiments were conducted on two data sets collected from Amazon.com reviews: LAPTOPS and DIGITALSLRS. The LAPTOPS data set contained 10,014 documents and users. The DIGITALSLRS data set contained 20,862 documents and 20,000 users. These data sets were both used in a previous study on hierarchies extraction [42]. To show the robustness of our model in comparison to state-of-art methods, another experiment was conducted on short texts. The publicly available data set called Smartphone is used, which contained more than 68,000 distinct tweets and 17,800 users crawled from Twitter [5, 6, 8]. For all data sets, the documents were pre-processed by lowercasing all words, removing all stop-words and filtering out all words that appeared too rarely in the documents (i.e., appeared less than five times in the entire corpus).

### 5.5.2 Methods for Comparison

Our approach was compared with three typical models of hierarchical construction.

- **rCRP** [45]. A non-parametric generative model that hierarchically discovers the structure of topics from data. The rCRP assumes each document to be generated over the entire topic hierarchy.

- **nCRP** [15]. A hierarchical model that constructs a tree-structured hierarchy of topics.

This model enables each document and topic to be generated by a single path from the root node to a leaf node.

- **HASM** [47]. A non-parametric hierarchical model that discovers a hierarchy with unbounded depth and width. The model accepts review data as input and generates a topic tree with the corresponding sentiment polarities.

- **Experiment Setup**. For the Dirichlet hyperparameters, we set $\alpha = 1.01$, $\beta = 0.1$, $\eta = 0.5$, $\gamma = 0.1$, and $iteration = 500$. For all other models, their hyperparameters are tuned to generate an approximately identical number of topics. Figures 5.3 and 5.4 show parts of the topical trees discovered by the HUSTM model on the Smartphone and LAPTOPS data sets, respectively.

### 5.5.3 Hierarchy Visualization

We designed the HUSTM to produce a high-quality hierarchical structure. As can be seen in Figures 5.3 and 5.4, each topic in the hierarchy corresponds to three sentiment polarities. Words such as *beautiful*, *nice* and *good* were classified as positive topics, while *scratches*, *damn* and *smash* were considered negative. Moreover, the HUSTM clearly outputs a hierarchical tree where each parent topic was related to its direct child topics. For example, the child topics *screen, colour, quality etc.* under the parent topics *screen, protector, colour etc.* were related. Another advantage of the HUSTM was that it clustered users to different groups in the hierarchy according to the topics discussed and sentiment polarities. Figure 5.5 shows a sample of the topics discussed by various users. For instance, the group of users *ripatici*, *kellymensa*, *eddyc42* etc. is negative about the topic âĂIJheadphoneâĂİ, while the group *dakotayounger*, *JackBoeth*, *GBMendy* etc. is positive about it. Figure 5.5 illustrates an example of texts used by users to share their opinions about the topics.

### 5.5.4 Evaluation Methods

In this chapter, we quantitatively evaluated the quality of the tree in terms of coverage, parent-child relatedness, and topic-sentiment consistency. These metrics were then used to compare the validity of a topic hierarchy constructed by the HUSTM with those obtained via the other methods.

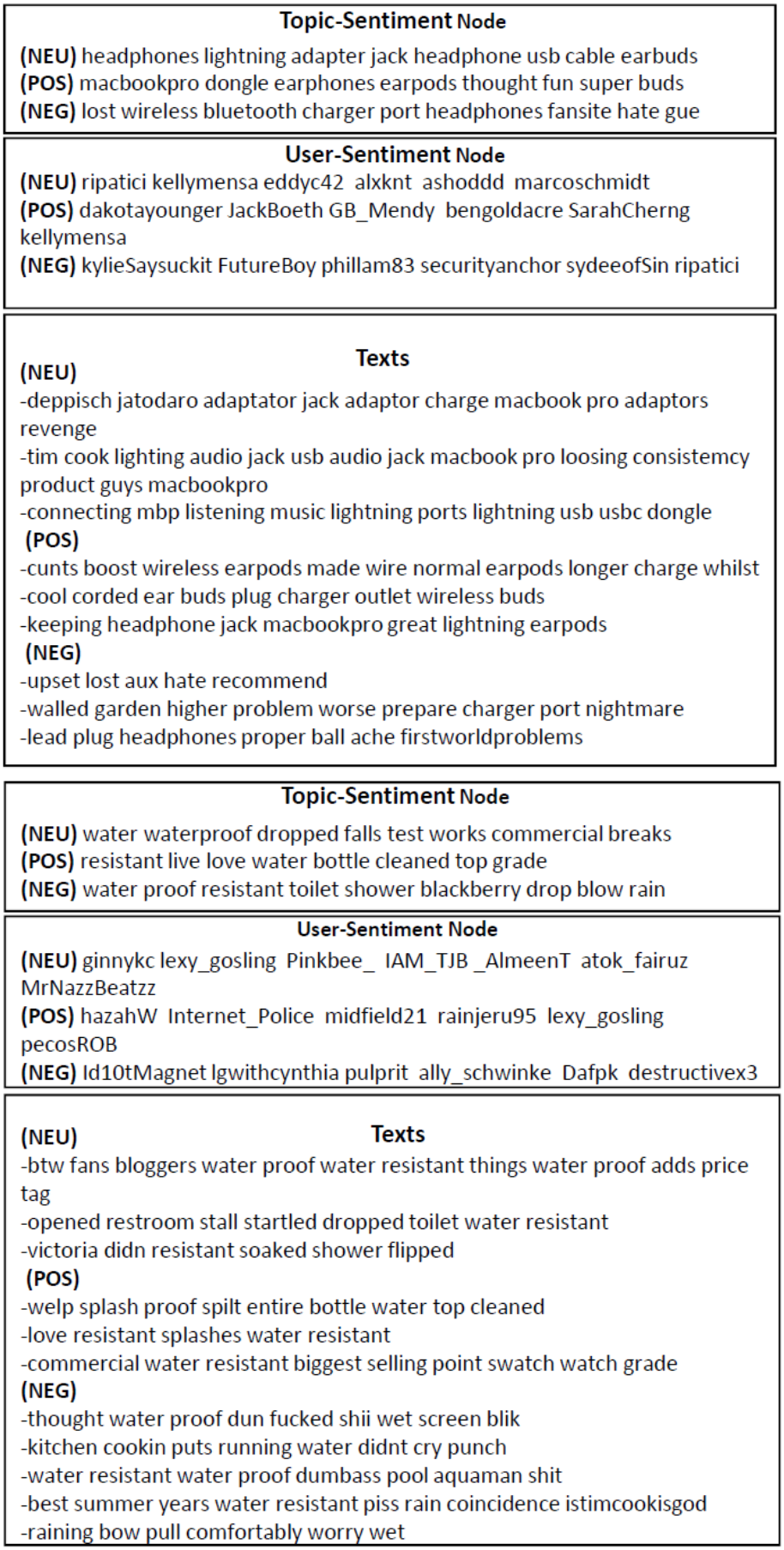



**Figure 5.5:** Example of user's interest on different topics .

**Table 5.1:** Average coverage score

| | LAPTOPS | | | DIGITALSLRS | | | SMARTPHONE | | |
|---|---|---|---|---|---|---|---|---|---|
| | Level 1 | Level 1 | Level 3 | Level 1 | Level 1 | Level 3 | Level 1 | Level 2 | Level 3 |
| **HUSTM** | 0.205 | 0.627 | 0.938 | 0.211 | 0.621 | 0.929 | 0.561 | 0.757 | 0.932 |
| **HASM** | 0.210 | 0.624 | 0.933 | 0.139 | 0.406 | 0.935 | 0.751 | 0.836 | - |
| **rCRP** | 0.288 | 0.696 | 0.839 | 0.234 | 0.646 | 0.822 | 0.684 | 0.712 | 0.783 |
| **nCRP** | 0.267 | 0.464 | 0.703 | 0.224 | 0.352 | 0.550 | 0.645 | 0.712 | 0.792 |

**Evaluation measure**

Cosine similarity was used in previous studies [47, 45] as a way of measuring the quality of a hierarchy. In our experiment, cosine similarity was used to compute the distance between the topic and sentiment.

$$ConSim(\Phi_{k_1,s}, \Phi_{k_2,s}) = 1 - \frac{\Phi_{k_1,s}.\Phi_{k_2,s}}{||\Phi_{k_1,s}||||\Phi_{k_2,s}||}. \tag{5.5.1}$$

**Coverage**

Measure coverage is used in this chapter to evaluate whether the topic hierarchy discovered by the HUSTM is organized from general to specific. Indeed, the coverage should reflect that the topics discovered near the root discuss general topics, like *battery*, *CPU* and *Software*. As the level increases, the topics should become more specific, such as *battery life*, *CPU speed* and *Norton*. The coverage score of the topic-sentiment nodes $\Phi_{k,s}$ at level $L$ is calculated as follows:

$$Coverage(\phi, \Phi_{k,s}, L) = \frac{1}{S}\sum_{s}^{S} ConSim(\phi, \Phi_{k,s}). \tag{5.5.2}$$

Similar to [45, 47], the root node $\phi$ is selected as a reference for measuring the coverage score. The average score is calculated for all topic-sentiment nodes at level $L$ as $\sum_{k=1}^{k} Coverage(\phi, \Phi_{k,s}, L)$. The results of the coverage score measuring are shown in Table 5.1. In long texts, the coverage score for all models increases as the level increases, which means that our assumptions are correctly reflected by the model. In short texts, the results clearly demonstrate a decrease in the coverage score as the level of the tree increases for all models. It is evident that the HUSTM outperforms the baseline method. In short texts, the HASM failed to discover a hierarchy with more than two depths. The reason for this is that

**Table 5.2:** Average coverage score

| | LAPTOPS | | | DIGITALSLRS | | | SMARTPHONE | | |
|---|---|---|---|---|---|---|---|---|---|
| | Level 1 | Level 2 | Level 3 | Level 1 | Level 2 | Level 3 | Level 1 | Level 2 | Level 3 |
| **HUSTM** | 0.205 | 0.627 | 0.938 | 0.211 | 0.621 | 0.929 | 0.561 | 0.757 | 0.932 |
| **HASM** | 0.210 | 0.624 | 0.933 | 0.139 | 0.406 | 0.935 | 0.751 | 0.836 | - |
| **rCRP** | 0.288 | 0.696 | 0.839 | 0.234 | 0.646 | 0.822 | 0.684 | 0.712 | 0.783 |
| **nCRP** | 0.267 | 0.464 | 0.703 | 0.224 | 0.352 | 0.550 | 0.645 | 0.712 | 0.792 |

the HASM tried to model every sentence independently, which is not effective with short texts.

**Evaluation measure**

Cosine similarity was used in previous studies [47, 45] as a way of measuring the quality of a hierarchy. In our experiment, cosine similarity was used to compute the distance between the topic and sentiment.

$$ConSim(\Phi_{k_1,s}, \Phi_{k_2,s}) = 1 - \frac{\Phi_{k_1,s}.\Phi_{k_2,s}}{||\Phi_{k_1,s}||||\Phi_{k_2,s}||}. \tag{5.5.3}$$

**Coverage**

Measure coverage is used in this chapter to evaluate whether the topic hierarchy discovered by the HUSTM is organized from general to specific. Indeed, the coverage should reflect that the topics discovered near the root discuss general topics, like *battery*, *CPU* and *Software*. As the level increases, the topics should become more specific, such as *battery life*, *CPU speed* and *Norton*. The coverage score of the topic-sentiment nodes $\Phi_{k,s}$ at level $L$ is calculated as follows:

$$Coverage(\phi, \Phi_{k,s}, L) = \frac{1}{S}\sum_{s}^{S} ConSim(\phi, \Phi_{k,s}). \tag{5.5.4}$$

Similar to [45, 47], the root node $\phi$ is selected as a reference for measuring the coverage score. The average score is calculated for all topic-sentiment nodes at level $L$ as $\sum_{k=1}^{k} Coverage(\phi, \Phi_{k,s}, L)$. The results of the coverage score measuring are shown in Table 5.1. In long texts, the coverage score for all models increases as the level increases, which means that our assumptions are correctly reflected by the model. In short texts, the results also demonstrate an increase in the coverage score as the level of the tree increases for all

models. In short texts, the HASM failed to discover a hierarchy with more than two depths. The reason for this is that the HASM tried to model every sentence independently, which is not effective with short texts. It is evident that the HUSTM outperforms the baseline method.

**Parent-Child Relatedness**

The goal of second evaluation was to analyse the semantic relatedness between the parent and child topics. It was assumed that parent topics should have more similarities with their direct child topics than with the child topics of other parent topics. For example, Figure 5.3 shows the parent topic *screen protector colour smashed, etc.* should be more similar to its direct child topics *screen, broken, protector, glass, etc.* and *colour, bright, quality, colour , etc.* than to non-child topics. In our experiment, we only calculated the relatedness score for a parent topic at a second level with its direct child topics at a third level. Given a parent node $\Phi_k$, we compute the parent-child relatedness score to its direct child node $\Phi_c$ and compare it to its non-child node $\hat{\Phi}_c$ by using the following equations :

$$\Delta(\Phi_k, \Phi_c) = \frac{1}{S}\sum_{s}^{S} 1 - \frac{\Phi_{k,s}.\Phi_{c,s}}{||\Phi_{k,s}||||\Phi_{c,s}||} \tag{5.5.5}$$

$$Children(\Phi_k, \Phi_c) = \frac{1}{n}\sum_{N} \Delta(\Phi_k, \Phi_{c_n}) \tag{5.5.6}$$

$$Non - Children(\Phi_k, \hat{\Phi_c}) = \frac{1}{n}\sum_{n} \Delta(\Phi_k, \hat{\Phi_{c_n}}) \tag{5.5.7}$$

We took the relatedness average to all children and non-children at the second level. Figure 5.6 illustrates the parent-child relatedness of four models. A lower value meant that a parent topic was more semantically related to its direct child topics than to its non-child topics. The HUSTM, HASM, and rCRP models showed significant differences in the semantics for direct child and non-child nodes. This means that the direct child topics were semantically related to their parent topics. In contrast, the nCRP shows different pattern in the relatedness score for child topics compared to non-child topics. The relatedness of the HASM was not calculated for the short texts, since it was only a two-level tree. The reason for this was that, in the Smartphone data set, the number of words in every sentence was limited. Hence, modelling every sentence independently led to a flat tree. In contrast, since the HUSTM modelled every word in the document, the results show its effectiveness dealing with short text.

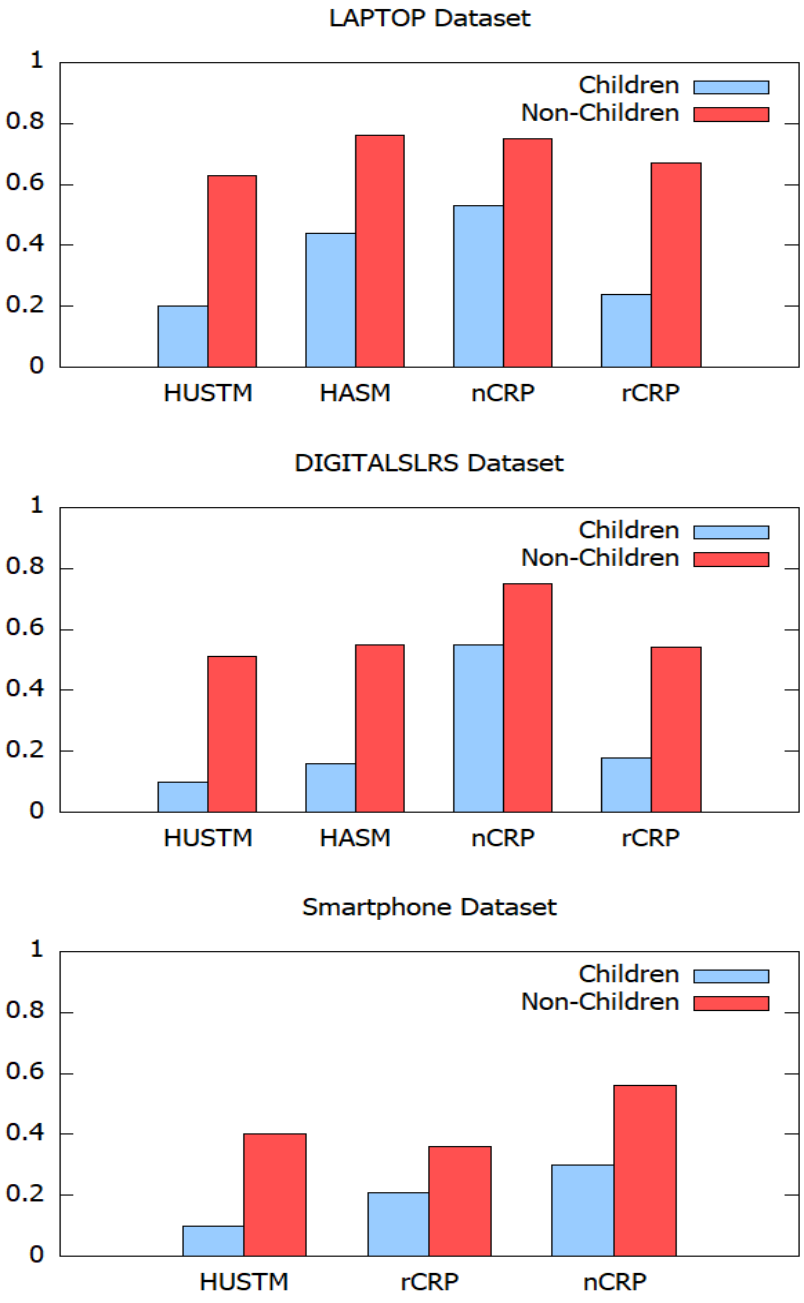



**Figure 5.6:** Parent-Child relatedness.

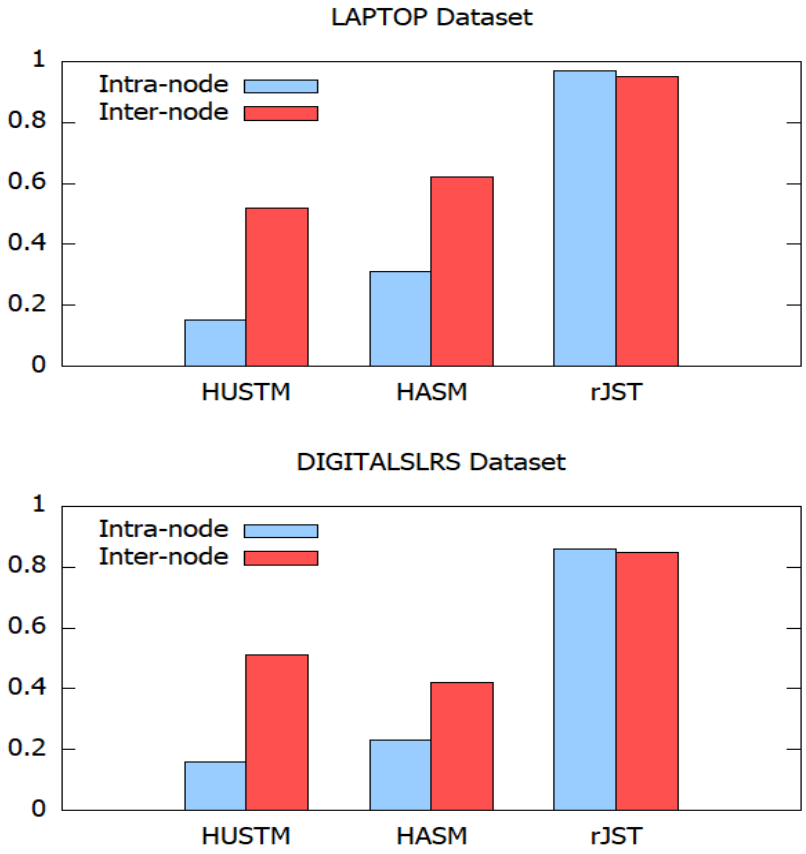



**Figure 5.7:** Topic Sentiment Consistency.

### Topic-Sentiment Consistency

The third evaluation aimed at measuring the consistency of the topic-sentiment nodes. In the HUSTM, each topic-sentiment node in the tree was decomposed into three topics with the different sentiment polarities. Our goal was to measure the consistency of the intra-topic node and compare it to the inter-topic node. For a topic-sentiment node $\Phi_k$ at level $L$ and $\Phi_{c,l}$, the consistency of $\Phi_t$ is calculated at level $L$ as follows:

$$IntraNode : ConSim(\Phi_i, \Phi_j), \Phi_i \in \Phi_K, \Phi_j \in \Phi_k \tag{5.5.8}$$

$$InterNode : ConSim(\Phi_i, \Phi_j), \Phi_i \in \Phi_K, \Phi_j \in \Phi_c \tag{5.5.9}$$

We took the average consistency for every $\phi_K$ at level $L$. The overall topic-sentiment consistency was calculated by taking the average score of every node at level 2. In this experiment, we compared the HUSTM with the HASM and reverse Joint Sentiment Topic (rJST) model [57] due to the output of those models is the same as HUSTM. The consistency score is not calculated for rCRP and nCRP since every node in the tree is only represented by a single topic. Figure 5.7 shows the comparison results of HUSTM compared with HASM and

**Table 5.3:** Average running time in seconds.

| | LAPTOPS | DIGITALSLRS | SMARTPHONE |
|---|---|---|---|
| **HUSTM** | 795.4 | 913.8 | 640.4 |
| **HASM** | 704.9 | 834.7 | 563.6 |
| **rCRP** | 648.6 | 794.1 | 493.4 |
| **nCRP** | 632.3 | 785.7 | 483.2 |

rJST at level 2. The results show that the HUSTM and HASM achieved a lower intra-node distance than an inter-node distance, while the rJST results demonstrated high distances for both intra-nodes and inter-nodes. The comparison shows that the HUSTM achieved better topic-sentiment consistency.

#### Computation Time

In this section, we report the running time of the HUSTM and compare it with the existing models. In this experiment, the computer that has been used had the following specifications: Processor Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz, 16GB Memory and Windows 10. The average running time for the HUSTM, compared with the other methods, is shown in Table 5.2. As can be seen, the average running time for the HUSTM is comparable with the other models. **Note that** the HUSTM was proposed to identify three important factors (topic, user and sentiment) in a unified way. In rCRP and nCRP, the output is only a hierarchy of topics with neither the sentiment information nor the user's interest being considered. Even though the HUSTM consumes slightly more time, the quality of the hierarchy discovered by it is better than the others.

## 5.6 Summery

This chapter presents a hierarchical user sentiment topic model (HUSTM), which can discover the hierarchy of the topic and user data while performing a sentiment analysis simultaneously. The primary advantage of the HUSTM is that it allows modelling of the users' sentiment information in the model. It offers a general and effective model for answering questions about topics that the majority of users care about and why users like or dislike those topics. Experiments were conducted to evaluate the quality and consistency of the HUSTM, based on three data sets. The results demonstrated that the HUSTM was able to

achieve a high-quality hierarchy in comparison to the results generated by other existing models. In the future, we would like to carry out experiments on a much larger scale and to evaluate the quality of the model on data sets from different domains.

CHAPTER 6

# Conclusions and Future Work

In this chapter, the thesis is concluded by reflecting on the work's novel contributions. Then, potential future work is briefly discussed related to the research.

## 6.1 Summary

In this thesis, several novel models have been presented, along with effective approaches for solving the problems of hierarchical and sentiment mining in social media texts. In doing so, three tasks were accomplished: (1) three subsistent problems in structure and sentiment analyses were identified, (2) a novel and effective solution to address these problems was proposed, and (3) the effectiveness of the proposed methods were experimentally demonstrated.

In Chapter 1, the research background and the significance of the research for structured sentiment analysis was presented. Then, a real-world example to show the limitations of the current research was presented. Finally, the research problems and challenges were discussed, as well as the main contributions of this thesis.

In Chapter 2, the existing work related to the research topic was discussed. The related work was then divided into three sections. First, the traditional topic models were presented and their limitations were illustrated. Next, the user and sentiment models were discussed. Then, a brief summary of sentiment analysis was presented. Finally, the existing hierarchical models were introduced and their limitations were discussed.

In Chapter 3, an approach called *CCM* was proposed for the hierarchical construction of concepts from short texts. The approach had three stages. First, concept extraction was performed to reveal and identify the semantic relatedness between concepts. In particular, a new notion called *context coherence* was introduced. *Context coherence* is a measurement

used to identify concepts and find the semantic meanings among them. The coverage of a given word was calculated by identifying the number of words that were related to it. Then, a top-down splitting algorithm was proposed to hierarchically organize the concepts into specific nodes. The approach required no prior knowledge of the depth and width of the tree because the depth and width of the tree are automatically inferred from the data. Finally, a merging algorithm was proposed to group similar concepts and remove redundant ones. In order to evaluate and compare the CCM model with other methods, three subjective evaluation methods were proposed to measure the appropriateness of the concept hierarchy, quality of concept, coverage, and parent-children relatedness. The approach was evaluated with two real-world data sets. The results showed that CCM achieved better results in discovering the concept hierarchy from short text compared to other tested state-of-the-art techniques.

In Chapter 4, a review of how people typically share their experiences with different aspects of a topic on social media and that organizations may want to know and understand people's opinions on their product was provided. How an SSA approach works to hierarchically summarize people's feelings as shared in social media was shown. In other words, SSA can automatically extract *hot aspects* (the most frequently mentioned aspects that interest people) as well as opinions about those aspects. At a high level, the proposed approach consists of four components. First, data pre-processing is performed to reduce noise and improve the results of SSA. Second, a hierarchical model is used to construct a tree of the *hot aspects*. Third, a top-down approach performs the sentiment analysis on the extracted aspects. Given the users' messages, the polarities of aspects are determined hierarchically. In other words, the prolixity of a parent aspect is calculated by including the polarity of itself and its children. Finally, to help with understanding and interpreting the results, a simple summarizing algorithm is used to generate a summary and explanation of the tree. The structured algorithm can help (1) show the polarity of aspects at different levels of granularity, (2) explain why people like or dislike the aspects, (3) illustrate the most favourable and unfavourable aspects, and (4) compare two or more products at various levels. In order to evaluate the approach, data from extracted from Twitter for tweets about three smartphone devices. Then, SSA was applied to extract the structured sentiment tree. The experimental results confirmed the effectiveness and efficiency of the approach in discovering and analysing the sentiment polarity of those aspects of the smartphones.

In Chapter 5, a novel probabilistic model, called HUSTM, was introduced to discover the topics, user interests and sentiment analysis in a unified manner. The primary goal

of HUSTM was to model and cluster users' interests using different topic and sentiment information. In HUSTM, the entire structure is a tree where each node is decomposed into two sub-nodes: (1) the topic/sentiment node, which reflects the topic of interest to that specific community; and (2) the user-sentiment node, which models the users' attitudes with respect to the sentiment information. To automatically infer the depth and width of the tree, rCRP was adopted as a prior tree, and the data was observed to modify and infer the posterior tree. The model was experimentally evaluated using three data sets, and the results showed a high-quality hierarchy had been discovered by the model.

## 6.2 Future Work

Many real-world challenges remain unsolved. In this section, three future work directions that have potential for further investigation are briefly introduced.

## 6.3 Cross-Domain Structure Analysis

There are a number of web pages that contain structured information, such as Wikipedia, YAGO and data.gov. It is possible to use data fusion techniques to integrate the structured information from those sites to build a huge structure that contains information from different domains. Future work would greatly benefit from cross-domain analysis. However, this task is challenging since different sources of data (e.g., different structures) have to be considered. Another challenge is how to keep discovering interesting information and keep continuously in the extracted structure.

## 6.4 Demographic Structure Analysis

In Chapter 5, an approach that clusters users' interests using different topic and sentiment information was proposed. In the real world, there is other critical demographic information that needs to be considered, such as age, gender and location. For instance, if certain information is organized as follows: age (e.g., "15-20", "21-30", "31-50"), gender ("male" or "female") and location (e.g., "Brisbane", "Melbourne" and "Sydney"), then one of the possible groups would be "male" aged "21-30" living in "Brisbane". Such information about users can help companies to better understand the preferences of different groups and improve their

future plans, accordingly. Hence, it would be helpful to introduce an approach that incorporates this demographic information in the structured sentiment tree in order to identify relationships between demographic information, concepts and sentiment.

# References


[1] Apache lucene. URL https://lucene.apache.org/core/.

[2] Twitter api. URL https://developer.twitter.com/en/docs.html.

[3] Opinion lexicon. URL https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html.

[4] D. J. Aldous. Exchangeability and related topics. In P. L. Hennequin, editor, *École d'Été de Probabilités de Saint-Flour XIII — 1983*, pages 1–198, Berlin, Heidelberg, 1985. Springer Berlin Heidelberg. ISBN 978-3-540-39316-0.

[5] A. Almars, X. Li, X. Zhao, I. A. Ibrahim, W. Yuan, and B. Li. Structured sentiment analysis. In *Advanced Data Mining and Applications*, pages 695–707. Springer International Publishing, 2017.

[6] A. Almars, I. A. Ibrahim, S. A. Maskari, and X. Zhao. Evaluation methods of hierarchical models. In *Advanced Data Mining and Applications*, pages 455–464. Springer International Publishing, 2018.

[7] A. Almars, X. Li, I. A. Ibrahim, and X. Zhao. *Learning Concept Hierarchy from Short Texts Using Context Coherence: 19th International Conference, Dubai, United Arab Emirates, November 12-15, 2018, Proceedings, Part I*, pages 319–329. 11 2018.

[8] A. Almars, X. Li, and X. Zhao. Modeling user attitudes using hierarchical sentiment-topic model. *Data Knowledge Engineering*, 2019. ISSN 0169-023X. doi: https://doi.org/10.1016/j.datak.2019.01.005. URL http://www.sciencedirect.com/science/article/pii/S0169023X18304828.

[9] V. Anoop, S. Asharaf, and P. Deepak. Unsupervised concept hierarchy learning: A topic modeling guided approach. *Procedia Computer Science*, 89:386 – 394, 2016. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2016.06.086. Twelfth International

Conference on Communication Networks, ICCN 2016, August 19âĂŞ 21, 2016, Bangalore, India Twelfth International Conference on Data Mining and Warehousing, ICDMW 2016, August 19-21, 2016, Bangalore, India Twelfth International Conference on Image and Signal Processing, ICISP 2016, August 19-21, 2016, Bangalore, India.

[10] M. F. A. Bashri and R. Kusumaningrum. Sentiment analysis using latent dirichlet allocation and topic polarity wordcloud visualization. *2017 5th International Conference on Information and Communication Technology (ICoIC7)*, pages 1–5, 2017.

[11] H. Binali, V. Potdar, and C. Wu. A state of the art opinion mining and its application domains. In *2009 IEEE International Conference on Industrial Technology*, pages 1–6, Feb 2009. doi: 10.1109/ICIT.2009.4939640.

[12] D. M. Blei. Probabilistic topic models. *Commun. ACM*, 55(4):77–84, Apr. 2012. ISSN 0001-0782. doi: 10.1145/2133806.2133826. URL `http://doi.acm.org/10.1145/2133806.2133826`.

[13] D. M. Blei and J. D. Lafferty. Correlated topic models. In *Proceedings of the 18th International Conference on Neural Information Processing Systems*, NIPS'05, pages 147–154, Cambridge, MA, USA, 2005. MIT Press.

[14] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[15] D. M. Blei, T. L. Griffiths, and M. I. Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *J. ACM*, 2010.

[16] J. L. Boyd-graber and D. M. Blei. Syntactic topic models. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 185–192. Curran Associates, Inc., 2009. URL `http://papers.nips.cc/paper/3398-syntactic-topic-models.pdf`.

[17] J. Chen, J. Zhu, J. Lu, and S. Liu. Scalable training of hierarchical topic models. *Proc. VLDB Endow.*, 11(7):826–839, Mar. 2018. ISSN 2150-8097. doi: 10.14778/3192965.3192972. URL `https://doi.org/10.14778/3192965.3192972`.

[18] P. Chen, N. L. Zhang, T. Liu, L. K. M. Poon, and Z. Chen. Latent tree models for hierarchical topic detection. *CoRR*, 2016.

[19] X. Cheng, X. Yan, Y. Lan, and J. Guo. Btm: Topic modeling over short texts. *IEEE Transactions on Knowledge and Data Engineering*, 26(12):2928–2941, Dec 2014. ISSN 1041-4347. doi: 10.1109/TKDE.2014.2313872.

[20] J.-T. Chien and Y.-L. Chang. Hierarchical theme and topic model for summarization. *2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2013.

[21] K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16(1):22–29, Mar. 1990. ISSN 0891-2017.

[22] M. Danilevsky, C. Wang, F. Tao, S. Nguyen, G. Chen, N. Desai, L. Wang, and J. Han. Amethyst: A system for mining and exploring topical hierarchies of heterogeneous data. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 1458–1461, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2174-7. doi: 10.1145/2487575.2487716. URL `http://doi.acm.org/10.1145/2487575.2487716`.

[23] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6): 391, 1990.

[24] N. Du, B. Wang, and B. Wu. Overlapping community structure detection in networks. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 1371–1372, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-991-3. doi: 10.1145/1458082.1458285. URL `http://doi.acm.org/10.1145/1458082.1458285`.

[25] S. T. Dumais. Latent semantic analysis. *Annual Review of Information Science and Technology*, 38(1):188–230, 2004. doi: 10.1002/aris.1440380105. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/aris.1440380105`.

[26] P. W. Foltz. Latent semantic analysis for text-based research. *Behavior Research Methods, Instruments, & Computers*, 28(2):197–202, Jun 1996. ISSN 1532-5970. doi: 10.3758/BF03204765. URL `https://doi.org/10.3758/BF03204765`.

[27] B. C. M. Fung, K. Wang, and M. Ester. Hierarchical document clustering using frequent itemsets. In *Proceedings of the Third SIAM International Conference on Data Mining, San Francisco, CA, USA, May 1-3, 2003*, pages 59–70, 2003.

[28] B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag, Berlin, Heidelberg, 1st edition, 1997. ISBN 3540627715.

[29] S. Gerani, G. Carenini, and R. T. Ng. Modeling content and structure for abstractive review summarization. *Computer Speech Language*, 53:302 – 331, 2019. ISSN 0885-2308.

[30] Z. Ghahramani, M. I. Jordan, and R. P. Adams. Tree-structured stick breaking for hierarchical data. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 19–27. Curran Associates, Inc., 2010. URL http://papers.nips.cc/paper/4108-tree-structured-stick-breaking-for-hierarchical-data.pdf.

[31] H. Gómez-Adorno, Y. Alemán, D. V. Ayala, M. A. Sánchez-Pérez, D. Pinto, and G. Sidorov. Author clustering using hierarchical clustering analysis. In *CLEF*, 2017.

[32] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235, 2004. ISSN 0027-8424. doi: 10.1073/pnas.0307752101. URL https://www.pnas.org/content/101/suppl_1/5228.

[33] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[34] Y. He, C. Wang, and C. Jiang. Discovering canonical correlations between topical and topological information in document networks. *IEEE Transactions on Knowledge and Data Engineering*, 30(3):460–473, March 2018. ISSN 1041-4347. doi: 10.1109/TKDE.2017.2767599.

[35] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1):177–196, Jan 2001. ISSN 1573-0565. doi: 10.1023/A:1007617005950. URL https://doi.org/10.1023/A:1007617005950.

[36] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA, 2004. ACM. ISBN 1-58113-888-1. doi: 10.1145/1014052.1014073. URL http://doi.acm.org/10.1145/1014052.1014073.

[37] M. Hu and B. Liu. Opinion extraction and summarization on the web. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, AAAI'06, pages 1621–

1624. AAAI Press, 2006. ISBN 978-1-57735-281-5. URL http://dl.acm.org/citation.cfm?id=1597348.1597456.

[38] X. Hu, H. Wang, and P. Li. Online biterm topic model based short text stream classification using short text expansion and concept drifting detection. *Pattern Recognition Letters*, 116, 10 2018. doi: 10.1016/j.patrec.2018.10.018.

[39] J. Huang, M. Peng, and H. Wang. Topic detection from large scale of microblog stream with high utility pattern clustering. In *Proceedings of the 8th Workshop on Ph.D. Workshop in Information and Knowledge Management*, PIKM '15, pages 3–10. ACM, 2015. ISBN 978-1-4503-3782-3.

[40] S. Jameel, W. Lam, and L. Bing. Nonparametric topic modeling using chinese restaurant franchise with buddy customers. In A. Hanbury, G. Kazai, A. Rauber, and N. Fuhr, editors, *Advances in Information Retrieval*, pages 648–659, Cham, 2015. Springer International Publishing. ISBN 978-3-319-16354-3.

[41] S. Jiang, X. Qian, J. Shen, Y. Fu, and T. Mei. Author topic model-based collaborative filtering for personalized poi recommendations. *IEEE Transactions on Multimedia*, 17: 907–918, 2015.

[42] Y. Jo and A. H. Oh. Aspect and sentiment unification model for online review analysis. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 815–824, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0493-1.

[43] N. Kawamae. Author interest topic model. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 887–888, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0153-4.

[44] N. Kawamae. Hierarchical approach to sentiment analysis. In *2012 IEEE Sixth International Conference on Semantic Computing*, pages 138–145, Sep. 2012. doi: 10.1109/ICSC.2012.62.

[45] J. H. Kim, D. Kim, S. Kim, and A. Oh. Modeling topic hierarchies with the recursive chinese restaurant process. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 783–792. ACM, 2012. ISBN 978-1-4503-1156-4.

[46] M.-S. Kim and J. Han. A particle-and-density based evolutionary clustering method for dynamic networks. *Proc. VLDB Endow.*, 2(1):622–633, Aug. 2009. ISSN 2150-8097. doi: 10.14778/1687627.1687698. URL https://doi.org/10.14778/1687627.1687698.

[47] S. Kim, J. Zhang, Z. Chen, A. Oh, and S. Liu. A hierarchical aspect-sentiment model for online reviews. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, AAAI'13, pages 526–533. AAAI Press, 2013.

[48] S.-M. Kim and E. Hovy. Determining the sentiment of opinions. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics. doi: 10.3115/1220355.1220555. URL https://doi.org/10.3115/1220355.1220555.

[49] E. Kontopoulos, C. Berberidis, T. Dergiades, and N. Bassiliades. Ontology-based sentiment analysis of twitter posts. *Expert Systems with Applications*, 40(10):4065 – 4074, 2013. ISSN 0957-4174.

[50] A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato. Finding statistically significant communities in networks. *PLOS ONE*, 6(4):1–18, 04 2011. doi: 10.1371/journal.pone.0018961. URL https://doi.org/10.1371/journal.pone.0018961.

[51] C. Li, H. Wang, Z. Zhang, A. Sun, and Z. Ma. Topic modeling for short texts with auxiliary word embeddings. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 165–174, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4069-4. doi: 10.1145/2911451.2911499. URL http://doi.acm.org/10.1145/2911451.2911499.

[52] D. Li, Y. Ding, C. Sugimoto, B. He, J. Tang, E. Yan, N. Lin, Z. Qin, and T. Dong. Modeling topic and community structure in social tagging: The ttr-lda-community model. *Journal of the American Society for Information Science and Technology*, 62(9):1849–1866. doi: 10.1002/asi.21581. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.21581.

[53] D. Li, Y. Ding, X. Shuai, J. Bollen, J. Tang, S. Chen, J. Zhu, and G. Rocha. Adding community and dynamic to topic models. *Journal of Informetrics*, 6(2):237 – 253, 2012. ISSN 1751-1577.

[54] W. Li and A. McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 577–584. ACM, 2006. ISBN 1-59593-383-2.

[55] C. Lin and Y. He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 375–384. ACM, 2009. ISBN 978-1-60558-512-3.

[56] C. Lin, B. Zhao, Q. Mei, and J. Han. Pet: A statistical model for popular events tracking in social communities. In *KDD'10 - Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data*, pages 929–938, 9 2010. ISBN 9781450300551. doi: 10.1145/1835804.1835922.

[57] C. Lin, Y. He, R. Everson, and S. Ruger. Weakly supervised joint sentiment-topic detection from text. *IEEE Transactions on Knowledge and Data Engineering*, 24(6):1134–1145, 2012.

[58] T. Liu, N. L. Zhang, and P. Chen. Hierarchical latent tree analysis for topic detection. In T. Calders, F. Esposito, E. Hüllermeier, and R. Meo, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 256–272, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[59] X. Liu, Y. Song, S. Liu, and H. Wang. Automatic taxonomy construction from keywords. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 1433–1441, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1462-6.

[60] Y. Liu, A. Niculescu-Mizil, and W. Gryc. Topic-link lda: Joint models of topic and author community. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 665–672, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1.

[61] H. Mahmoud. Polya urn models. In *Chapman Hall/CRC Texts in Statistical Science*, ICML '06, 2008.

[62] A. McCallum, X. Wang, and A. Corrada-Emmanuel. Topic and role discovery in social networks with experiments on enron and academic email. *J. Artif. Int. Res.*, 30(1):249–272, Oct. 2007. ISSN 1076-9757.

[63] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: Modeling facets and opinions in weblogs. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-654-7.

[64] D. Mimno and A. McCallum. Expertise modeling for matching papers with reviewers. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 500–509, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-609-7. doi: 10.1145/1281192.1281247. URL `http://doi.acm.org/10.1145/1281192.1281247`.

[65] D. Mimno, W. Li, and A. McCallum. Mixtures of hierarchical topics with pachinko allocation. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 633–640, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-793-3.

[66] R. Mourad, C. Sinoquet, N. L. Zhang, T. Liu, and P. Leray. A survey on latent tree models and applications. *J. Artif. Int. Res.*, 2013.

[67] A. Mukherjee and B. Liu. Aspect extraction through semi-supervised modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 339–348, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=2390524.2390572`.

[68] S. Mukherjee, G. Basu, and S. Joshi. Joint author sentiment topic model. In *Proceedings of the 2014 SIAM International Conference on Data Mining, Philadelphia, Pennsylvania, USA, April 24-26, 2014*, pages 370–378, 2014.

[69] V.-A. Nguyen, J. L. Ying, P. Resnik, and J. Chang. Learning a concept hierarchy from multi-labeled documents. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3671–3679. Curran Associates, Inc., 2014.

[70] X. Ni, X. Quan, Z. Lu, L. Wenyin, and B. Hua. Short text clustering by finding core terms. *Knowledge and Information Systems*, 27(3):345–365, Jun 2011. ISSN 0219-3116. doi: 10.1007/s10115-010-0299-7. URL `https://doi.org/10.1007/s10115-010-0299-7`.

[71] J. Paisley, C. Wang, D. M. Blei, and M. I. Jordan. Nested hierarchical dirichlet processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):256–270, Feb 2015. ISSN 0162-8828. doi: 10.1109/TPAMI.2014.2318728.

[72] G. Palla, I. DerÃl'nyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, June 2005. ISSN 0028-0836. URL http://dx.doi.org/10.1038/nature03607.

[73] S. Papadopoulos, Y. Kompatsiaris, A. Vakali, and P. Spyridonos. Community detection in social media. *Data Mining and Knowledge Discovery*, 24(3):515–554, May 2012.

[74] N. Pathak, C. Delong, A. Banerjee, and K. Erickson. Social topic models for community extraction. 2008.

[75] I. PeÃśalver-Martinez, F. Garcia-Sanchez, R. Valencia-Garcia, M. ÃĄngel RodrÃ■guez-GarcÃ■a, V. Moreno, A. Fraga, and J. L. SÃąnchez-Cervantes. Feature-based opinion mining through ontologies. *Expert Systems with Applications*, 41(13): 5995 – 6008, 2014. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2014.03.022. URL http://www.sciencedirect.com/science/article/pii/S0957417414001511.

[76] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 91–100, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-085-2. doi: 10.1145/1367497.1367510. URL http://doi.acm.org/10.1145/1367497.1367510.

[77] X.-H. Phan, C.-T. Nguyen, D.-T. Le, L.-M. Nguyen, S. Horiguchi, and Q.-T. Ha. A hidden topic-based framework toward building applications with short web documents. *IEEE Transactions on Knowledge and Data Engineering*, 23(7):961–976, 2011.

[78] O. Phelan, K. McCarthy, and B. Smyth. Using twitter to recommend real-time topical news. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 385–388, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-435-5. doi: 10.1145/1639714.1639794. URL http://doi.acm.org/10.1145/1639714.1639794.

[79] A.-M. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 339–346, Stroudsburg, PA,

USA, 2005. Association for Computational Linguistics. doi: 10.3115/1220575.1220618. URL https://doi.org/10.3115/1220575.1220618.

[80] X. Quan, C. Kit, Y. Ge, and S. J. Pan. Short and sparse text topic modeling via self-aggregation. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 2270–2276. AAAI Press, 2015. ISBN 978-1-57735-738-4. URL http://dl.acm.org/citation.cfm?id=2832415.2832564.

[81] F. Rahutomo, T. Kitasuka, and M. Aritsugi. Semantic cosine similarity. 2012.

[82] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, pages 487–494, Arlington, Virginia, United States, 2004. AUAI Press. ISBN 0-9749039-0-6.

[83] M. Sachan, D. Contractor, T. A. Faruquie, and L. V. Subramaniam. Using content and interactions for discovering communities in social networks. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 331–340, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1229-5.

[84] H. Saif, Y. He, and H. Alani. Semantic sentiment analysis of twitter. In P. Cudré-Mauroux, J. Heflin, E. Sirin, T. Tudorache, J. Euzenat, M. Hauswirth, J. X. Parreira, J. Hendler, G. Schreiber, A. Bernstein, and E. Blomqvist, editors, *The Semantic Web – ISWC 2012*, pages 508–524, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-35176-1.

[85] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing Management*, 24(5):513 – 523, 1988. ISSN 0306-4573. doi: https://doi.org/10.1016/0306-4573(88)90021-0. URL http://www.sciencedirect.com/science/article/pii/0306457388900210.

[86] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths. Probabilistic author-topic models for information discovery. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 306–315, New York, NY, USA, 2004. ACM. ISBN 1-58113-888-1. doi: 10.1145/1014052.1014087. URL http://doi.acm.org/10.1145/1014052.1014087.

[87] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A large ontology from wikipedia and wordnet. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(3): 203–217, 2008.

[88] L. Tang and H. Liu. *Community Detection and Mining in Social Media*. Morgan and Claypool Publishers, 1st edition, 2010. ISBN 1608453545, 9781608453542.

[89] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Sharing clusters among related groups: Hierarchical dirichlet processes. In *In Advances in Neural Information Processing Systems*, pages 1385–1392. MIT Press, 2005.

[90] I. Titov and R. McDonald. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th international conference on World Wide Web*, pages 111–120. ACM, 2008.

[91] C. Wang, M. Danilevsky, N. Desai, Y. Zhang, P. Nguyen, T. Taula, and J. Han. A phrase mining framework for recursive construction of a topical hierarchy. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 437–445. ACM, 2013.

[92] C. Wang, M. Danilevsky, J. Liu, N. Desai, H. Ji, and J. Han. Constructing topical hierarchies in heterogeneous information networks. In *2013 IEEE 13th International Conference on Data Mining*, pages 767–776, Dec 2013.

[93] C. Wang, X. Yu, Y. Li, C. Zhai, and J. Han. Content coverage maximization on word networks for hierarchical topic summarization. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 249–258, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2263-8.

[94] C. Wang, X. Liu, Y. Song, and J. Han. Scalable and robust construction of topical hierarchies. *CoRR*, abs/1403.3460, 2014.

[95] C. Wang, X. Liu, Y. Song, and J. Han. Towards interactive construction of topical hierarchy: A recursive tensor decomposition approach. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, pages 1225–1234, 2015.

[96] X. Wang and E. Grimson. Spatial latent dirichlet allocation. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, pages 1577–

1584, USA, 2007. Curran Associates Inc. ISBN 978-1-60560-352-0. URL http://dl.acm.org/citation.cfm?id=2981562.2981760.

[97] X. Wang and A. McCallum. Topics over time: A non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 424–433. ACM, 2006. ISBN 1-59593-339-5.

[98] J. Weng, E.-P. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 261–270. ACM, 2010.

[99] J. Xie, B. K. Szymanski, and X. Liu. Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 344–349, Dec 2011. doi: 10.1109/ICDMW.2011.154.

[100] Y. Xu, J. Yin, J. Huang, and Y. Yin. Hierarchical topic modeling with automatic knowledge mining. *Expert Systems with Applications*, 103:106 – 117, 2018. ISSN 0957-4174.

[101] X. Yan, J. Guo, Y. Lan, and X. Cheng. A biterm topic model for short texts. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1445–1456. ACM, 2013.

[102] B. Yang and S. Manandhar. Stc: A joint sentiment-topic model for community identification. In W.-C. Peng, H. Wang, J. Bailey, V. S. Tseng, T. B. Ho, Z.-H. Zhou, and A. L. Chen, editors, *Trends and Applications in Knowledge Discovery and Data Mining*, 2014.

[103] Z. Yang, A. Kotov, A. Mohan, and S. Lu. Parametric and non-parametric user-aware sentiment topic models. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 413–422, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3621-5. doi: 10.1145/2766462.2767758.

[104] L. Yao, D. Mimno, and A. McCallum. Efficient methods for topic model inference on streaming document collections. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 937–946, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-495-9.

[105] Z. Yin, L. Cao, Q. Gu, and J. Han. Latent community topic analysis: Integration of community discovery with topic modeling. *ACM Trans. Intell. Syst. Technol.*, 3(4):63:1–63:21, Sept. 2012. ISSN 2157-6904. doi: 10.1145/2337542.2337548. URL http://doi.acm.org/10.1145/2337542.2337548.

[106] C. Zhang, F. Tao, X. Chen, J. Shen, M. Jiang, B. Sadler, M. Vanni, and J. Han. Taxogen: Unsupervised topic taxonomy construction by adaptive term embedding and clustering. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pages 2701–2709, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5552-0.

[107] P. Zhao, X. Li, and K. Wang. Feature extraction from micro-blogs for comparison of products and services. In X. Lin, Y. Manolopoulos, D. Srivastava, and G. Huang, editors, *Web Information Systems Engineering – WISE 2013*, pages 82–91, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[108] T. Zhao, C. Li, Q. Ding, and L. Li. User-sentiment topic model: Refining user's topics with sentiment information. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, MDS '12, pages 10:1–10:9, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1546-3. doi: 10.1145/2350190.2350200. URL http://doi.acm.org/10.1145/2350190.2350200.

[109] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li. Comparing twitter and traditional media using topic models. In P. Clough, C. Foley, C. Gurrin, G. J. F. Jones, W. Kraaij, H. Lee, and V. Mudoch, editors, *Advances in Information Retrieval*, pages 338–349, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-20161-5.

[110] D. Zhou, E. Manavoglu, J. Li, C. L. Giles, and H. Zha. Probabilistic models for discovering e-communities. In *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, pages 173–182, New York, NY, USA, 2006. ACM. ISBN 1-59593-323-9. doi: 10.1145/1135777.1135807. URL http://doi.acm.org/10.1145/1135777.1135807.

[111] W. Zhou, H. Jin, and Y. Liu. Community discovery and profiling with social messages. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 388–396, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1462-6.

[112] X. Zhu, Z.-Y. Ming, X. Zhu, and T.-S. Chua. Topic hierarchy construction for the organization of multi-source user generated contents. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 233–242, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2034-4. doi: 10.1145/2484028.2484032. URL http://doi.acm.org/10.1145/2484028.2484032.