INFERRING CONCEPTUAL GRAPHS AND RELATIONSHIPS FROM LARGE UNSTRUCTURED DATASETS

Submitted in partial fulfillment of the requirements
for the award of the degree of
DOCTOR OF PHILOSOPHY

by

ANOOP V. S.
Reg.No : 4852

under the supervision of

Dr. ASHARAF S
Professor
Indian Institute of Information Technology and Management - Kerala (IIITM-K)
Thiruvananthapuram, Kerala

COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY (CUSAT)
KOCHI, KERALA, INDIA- 682 022

Conducted by

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT - KERALA (IIITM-K)
THIRUVANANTHAPURAM, KERALA - 695581

January 2020

INFERRING CONCEPTUAL GRAPHS AND RELATIONSHIPS FROM LARGE UNSTRUCTURED DATASETS

Ph.D Thesis under the Faculty of Technology

Author:

ANOOP V.S.
Ph.D Research Scholar
Indian Institute of Information Technology and Management - Kerala (IIITM-K)
Thiruvananthapuram, Kerala, India – 695581

Supervising Guide:

Dr. ASHARAF S.
Professor
Indian Institute of Information Technology and Management - Kerala (IIITM-K)
Thiruvananthapuram, Kerala, India – 695581

COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY (CUSAT)
KOCHI, KERALA, INDIA- 682 022

Conducted by

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT - KERALA (IIITM-K)
THIRUVANANTHAPURAM, KERALA - 695581

Dedicated to my Mother & Grandmother

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT - KERALA (IIITM-K)
THIRUVANANTHAPURAM, KERALA - 695581

Certificate

Certified that the work presented in this thesis entitled **"Inferring Conceptual Graphs and Relationships from Large Unstructured Datasets"** is based on the authentic record of research done by **Mr. ANOOP V.S.** towards the partial fulfillment of the requirements for the award of degree of **Doctor of Philosophy** of the Cochin University of Science and Technology, under my guidance and supervision and that this work has not been submitted elsewhere for the award of any degree.

Signed: _____

Dr. ASHARAF S
(Supervising Guide)
Professor
Indian Institute of Information Technology and Management - Kerala (IIITM-K)
Thiruvananthapuram, Kerala

Date: _____

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT - KERALA (IIITM-K)
THIRUVANANTHAPURAM, KERALA - 695581

Certificate

Certified that the work presented in this thesis entitled **"Inferring Conceptual Graphs and Relationships from Large Unstructured Datasets"** submitted to Cochin University of Science and Technology by **Mr. ANOOP V.S.** for the award of degree of **Doctor of Philosophy** under the Faculty of Technology, contains all the relevant corrections and modifications suggested by the audience during the pre-synopsis seminar and recommended by the Doctoral Committee

Signed: _____

Dr. ASHARAF S
(Supervising Guide)
Professor
Indian Institute of Information Technology and Management - Kerala (IIITM-K)
Thiruvananthapuram, Kerala

Date: _____

Declaration

I hereby declare that, the work presented in this thesis entitled **"Inferring Conceptual Graphs and Relationships from Large Unstructured Datasets"** is based on the original research work carried out by me under the guidance and supervision of Dr. Asharaf S., Professor, Indian Institute of Information Technology and Management - Kerala (IIITM-K), Thiruvananthapuram – 695581 in partial fulfillment of the requirements for the award of the Degree of Doctor of Philosophy. I further declare that no part of the work reported in this thesis has been presented for the award of any degree from any other institution.

Signed: _____

ANOOP V. S.
Ph.D Research Scholar
Indian Institute of Information Technology and Management - Kerala (IIITM-K)
Thiruvananthapuram, Kerala

Date: _____

Acknowledgements

Firstly, I would like to express my sincere gratitude and admiration towards my research supervisor Dr. S. Asharaf (Professor, Indian Institute of Information Technology and Management - Kerala, Thiruvananthapuram) for his valuable guidance and encouragement that motivated me to take scientific research of this scale. I consider myself very fortunate to have worked under his guidance and support and his belief, patience and constructive feedback immensely helped me reach where I am today. My association with him taught me passion, involvement, hard work and attitude for what I believe in.

I would like to thank Dr. Saji Gopinath, Director, Indian Institute of Information Technology and Management - Kerala (IIITM-K) for all the facilities provided at IIITM-K for conducting this research and all his support and valuable suggestions that significantly improved the quality of this thesis. My sincere thanks also goes to Dr. Rajasree M.S. (Former Director, IIITM-K), who provided me an opportunity to join and pursue my research work in a premier institution like IIITM-K and also for her immense help, suggestions and encouragement for broadening my research spectrum. I specially thank my research collaborator Dr. Deepak Padmanabhan (Assistant Professor, School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, United Kingdom) for all his guidance. He was very helpful and instrumental in giving his thoughts on the research problems I attempted.

I am thankful to all the members of Doctoral Committee, especially Dr. Tony Thomas (Associate Professor, IIITM-K), all the members of the Research Committee and all other faculty members of IIITM-K for their constructive criticism and immense support that encouraged me to widen my research on various perspectives.

I also take this opportunity to thank my colleagues at Data Engineering Lab, IIITM-K for their support throughout this research and pleasant environment they have created. I thank my fellow researchers Mr. Afzal A.L. , Mr. Adarsh S, Ms. Jennath H.S., Mr. Nikhil Chandran V, Ms. Nikhitha K Nair, Ms. Arya Krishnan and Mr. Roopak for all their constructive views, criticisms and discussions. The encouragement and support given by the staffs of Kerala Blockchain Academy (KBA) is priceless and I am always

grateful to each one of them for making me at ease when writing this thesis. I thank Ms. Srikirupa V (Academic Assistant, IIITM-K) for all her motivation and timely support to complete the thesis formalities on time.

I am always grateful to my ever loving grandmother Smt. Lalithamma and my mother Smt. K Vasantha Kumary who have provided me through moral and emotional support in my life. They kept showing me unconditional love and support that pushed me to pursue my dreams every morning, ignoring the fact that I was not there when they needed me. I could never have reached this point without their love, prayers, support and nourishment.

I am also grateful to all my relatives and friends who have supported me along the way. Especially the encouragement given by my uncle Mr. Sajumon, my aunty Ms. Viji Saju and my cousins Gokul and Athul, throughout my research journey. I also take this opportunity to thank my friend Mr. Jeevan Kamal and all my well-wishers for their strong mental support to complete this thesis when I was going through one of the difficult stages of my life.

Last, but not least, I acknowledge all the other fellow beings and well wishers for supporting me mentally, physically and spiritually throughout my life.

Sincerely,

ANOOP V.S.

Abstract

Recent years witnessed an exponential growth in the amount of unstructured text data being generated from text producing applications such as social networks, electronic journals, web portals etc. Even though the Identification of useful patterns from such huge unstructured text archives is a very interesting opportunity in many application domains, it is often labour intensive. What is pragmatic is the algorithmic exploration of such corpora to unearth semantically rich and actionable insights. But most of the algorithms that are proposed in this area are purely statistical which do not consider the semantics or meaning of the data they operate on. On the other hand, there were few attempts that induces semantics into data using human curated ontologies and external knowledgebases such as DBPedia. Such ontologies comprise of entities, concepts and their relationships often find several applications involving meaning aware computing paradigms. Manual creation of such ontologies from large unstructured text archives is time consuming and inefficient. Such manually created ontologies are available only in a very limited set of application areas and hence there is a pressing need for novel algorithms that can induct potentially useful semantic structures such as concept graphs similar to ontologies. This thesis proposed algorithms that bridge the gap between the statistically generated text abstractions such as topic models the manually crafted ontological structures. The proposed methods make use of statistically inferred latent themes from unstructured text contents, and then incorporates natural language processing on top of this for extracting semantically valid concepts and key-phrases. It also explore dimensions for generating hierarchies of such concepts that may find applications in meaning aware systems that uses ontological structures. Using formal concept analysis and distributional semantics this thesis also explored algorithms for unearthing interpretable concepts and their enrichment. Three potential applications of such hierarchies and concept graphs in real-world problems such as phrase query based document ranking, aspect oriented sentiment analysis and information retrieval from online social networks are also explored in this thesis.

The first algorithm proposed in this thesis discusses an approach for extracting concepts and key-phrases from the outputs generated by algorithms that uses statistical or mathematical assumptions. The proposed technique uses the topics generated by topic

modeling algorithms, where topics are a collection of co-occuring words in a document, as the input. It then utilizes natural language processing techniques on top of these topics to generate more meaningful concepts. The second algorithm attempts to learn hierarchies of the extracted concepts and key-phrases which may aid in the process of creating ontological structures. When compared with the state-of-the-art approaches on real-life datasets, the proposed concept extraction and hierarchy learning algorithms performed better in terms of the quality of conceptual structures generated. An approach for identifying commonly occurring relations that connects concepts in unstructured text data and attempts to learn such relations using machine learning techniques is also proposed in this thesis. This approach can identify and learn commonly used relationships in textual data such as subsumption ("is-a"), Hearst patterns ("such as", "or other", "and other", "including", "especially" etc.) and other potential relations among concepts and key-phrases. It uses Formal Concept Analysis to form context table and concept lattices from the concepts and relationships identified from the unstructured data. The experiments conducted on real-world dataset shows the potential of the proposed approach in generating conceptual structures from unstructured text data.

The distributional semantics is a relatively new area in the field of machine learning and natural language processing. This thesis proposes an approach for conceptual knowledge discovery using distributional semantics. The proposed approach attempts to combine phrase embeddings and short text conceptualization for creating semantically rich and distinguishable phrase clusters. When executed on real-world datasets, the proposed algorithms showed interesting results and could conclude that the distributional semantics has a potential in conceptual knowledge discovery from unstructured text data. The final chapter in this thesis discusses a few potential real-world applications of the proposed algorithms such as phrase query based document ranking, aspect oriented sentiment analysis and a distributional semantics based information retrieval for online social networks. Rigorous and systematic experiments on real-world unstructured text datasets show that the proposed approaches can generate useful conceptual structures from large text corpora with many a real-world applications.

Contents

List of Figures **iv**

List of Tables **vi**

1 Introduction **1**
1.1 Research Motivation . . . 3
1.2 Problem Definition . . . 4
1.3 Thesis Outline . . . 4

2 Literature Survey **6**
2.1 Text Mining and Natural Language Processing . . . 7
2.2 Topic Modeling . . . 7
2.2.1 Latent Dirichlet Allocation (LDA) . . . 8
2.3 Concept Extraction from Unstructured Text . . . 10
2.4 Topical Phrase Discovery from Text . . . 12
2.5 Relationship Extraction from Text . . . 14
2.6 Distributional Representation of Phrases . . . 16

3 Concept Extraction and Concept Hierarchy Generation **17**
3.1 Extracting Concepts and Concept Hierarchies from Statistical Topic Models 18
3.1.1 Concept Extraction . . . 19
3.1.2 Concept Hierarchy Learning . . . 21
3.2 Experimental Setup . . . 23
3.2.1 Dataset Collection and Pre-processing . . . 23
3.2.2 Modeling Topics . . . 24
3.2.3 Topic - Document Clustering . . . 24
3.2.4 TF-ITF Weighting . . . 25
3.2.5 Sentence Extraction & POS Tagging . . . 26
3.2.6 Linguistic Processing & Concept Identification . . . 26
3.2.7 Concept Hierarchy Learning . . . 26
3.2.8 Evaluation of Results . . . 27
3.3 Conclusions . . . 29

4 Concept Lattice Generation and Relationship Extraction **30**
4.1 Concept Lattice Generation and Relationship Extraction from Unstructured Text . . . 32
4.2 Experimental Setup . . . 33
4.2.1 Dataset Description . . . 34

4.2.2 Dataset Pre-processing . . . 34
4.2.3 Building Machine Learning Model . . . 35
4.2.4 Creating Concept Lattices . . . 35
4.2.5 Baselines Chosen . . . 37
4.2.6 Results and Evaluation . . . 37
4.3 Conclusions . . . 40

5 Conceptual Knowledge Discovery using Distributional Semantic Models 41
5.1 Conceptualized Phrase Clustering . . . 43
5.1.1 Dataset Pre-processing . . . 46
5.1.2 Phrase Identification and Tagging . . . 46
5.1.3 Generating Phrase Embedding and Clustering Phrases . . . 46
5.1.4 Conceptualization of Phrases . . . 47
5.2 Experimental Setup . . . 47
5.2.1 Dataset Description . . . 48
5.2.2 Experimental Testbed . . . 48
5.3 Results and Evaluation . . . 49
5.4 Conclusions . . . 51

6 A Few Real-world Applications of the Proposed Algorithms 53
6.1 Phrase Query based Document Ranking . . . 54
6.1.1 Problem definition . . . 55
6.1.2 Topic modeling guided concept extraction . . . 55
6.1.3 Topic modeling guided document retrieval . . . 57
6.1.4 Experimental setup . . . 58
6.1.5 Dataset description . . . 59
6.1.6 Experimental testbed for topic induced concept extraction . . . 59
6.1.7 Experimental testbed for document retrieval . . . 61
6.1.8 Result and evaluation . . . 61
6.1.9 Qualitative evaluation . . . 61
6.1.10 Evaluation of document retrieval . . . 62
6.2 Aspect Oriented Sentiment Analysis . . . 64
6.2.1 Proposed Approach . . . 65
6.2.2 Experimental Setup . . . 68
6.2.3 Dataset Description . . . 68
6.2.4 Experimental Testbed . . . 68
6.2.5 Baselines . . . 69
6.2.6 Results and Evaluation . . . 69
6.3 Information Retrieval from Social Network Data . . . 72
6.3.1 Background: Distributional Representation of Phrases . . . 75
6.3.2 Proposed Approach . . . 76
6.3.2.1 Online Processing . . . 76
6.3.2.2 Offline Processing . . . 77
6.3.3 Experimental Setup . . . 79
6.3.4 Dataset Description . . . 79
6.3.5 Phrase Extraction . . . 80

6.3.6 Experimental Testbed 80
6.3.7 Results and Evaluation 82
6.3.8 Conclusions 84

7 Conclusions and Future Work **86**
7.1 Conclusions 86
7.2 Future Works 88

References **89**

Publication **105**

List of Figures

3.1 Workflow of the proposed concept hierarchy learning method 19
3.2 A partial snapshot of one of the concept hierarchy 27
3.3 Precision, Recall and F-measure comparison of ACE, ICE and proposed method 28
3.4 Performance of concept extraction algorithm on first 100 topics generated 29

4.1 Formal context showing airlines and their sector of operations 31
4.2 Concept lattice generated for formal context given in Fig. 4.1 31
4.3 Overall workflow of the proposed approach 33
4.4 A snapshot of disease description collected from web 34
4.5 Part of a formal context generated from the original dataset using our proposed method 36
4.6 Part of a concept lattice generated from the original dataset using our proposed method 36
4.7 Graph representation of classifier performance on the chosen dataset . . . 38
4.8 Normalized confusion matrix for Adaboost, Random Forest, Decision Tree and SVM classifiers 39

5.1 Proposed method 44
5.2 Cluster visualization of phrase vectors obtained from phrase2vec with varying sample sizes (5000 and 10000) with centers = 5, number of features = 2, and random state = 0 51
5.3 Cluster visualization of phrase vectors obtained from phrase2vec with varying sample sizes (20000 and 40000) with centers = 5, number of features = 2, and random state = 0 51

6.1 Precision, Recall and F-measure values of concept extraction and document retrieval when compared with baseline methods on Reuters, BBC and StackExchange datasets 63
6.2 Overall workflow of the proposed approach for topic modeling powered aspect oriented sentiment analysis 66
6.3 ROC plots for 'Battery' & 'Body' 71
6.4 ROC plots for 'Display' & 'Network' 71
6.5 ROC plots for 'Price' & 'Service' 71
6.6 Screenshot of web application developed for finding aspect oriented sentiment of customers for mobile phones and accessories using proposed algorithms. This system has been extended to production level and a number of vendors are already using pilot version of the same. 74

6.7 Skip-gram model for phrases used in this work. $p(t)$ in the input layer denotes the input/query phrase and $p(t-2)$, $p(t-1)$, $p(t+1)$, $p(t+2)$ in the output layer denotes phrases from distributed representation which are semantically closer to $p(t)$ 76
6.8 : Overall workflow of the proposed framework 78
6.9 A snapshot of tweets collected for the hashtag "#DeMonetisation" 79
6.10 Precision, Recall and F-measure values obtained for the proposed algorithm on tweets retrieval task. 85

List of Tables

3.1 Top 10 topic words from Topic 1 and Topic 2 along with their computed TF-ITF weight . . . 24
3.2 Top 10 topic words from Topic 3 and Topic 4 along with their computed TF-ITF weight . . . 25
3.3 Top concepts extracted against first 4 topics . . . 25
3.4 Comparison of ACE, ICE and our proposed method . . . 28

4.1 Some of the concepts / medical phrases extracted for cardiology and neurology categories. . . . 38
4.2 Precision, Recall and F1 score of different classifier algorithms on our dataset . . . 39
4.3 Precision, Recall and F1 score comparison of baselines and our proposed method for relation extraction . . . 39
4.4 Precision, Recall and F1 score comparison of baselines and our proposed method for concept lattice generation . . . 39

5.1 Common NLP rules for identifying noun-phrases from corpus . . . 47
5.2 A partial list of phrases identified from BBC corpus (technology domain) using NLP rules given in Table 1 . . . 50
5.3 Clustering performance of proposed algorithms (using standard k - means and distributed k - means) with chosen baselines on three different datasets - BBC News, StackExchange and Reuters 21578 . . . 50
5.4 Result of conceptualization using Microsoft Concept Graph API service (for phrases given in Table 2) . . . 52

6.1 Comparison of proposed topic modeling guided concept extraction method with Topical N-gram, TopMine baselines. Proposed method outperforms significantly better than all the chosen baselines . . . 62
6.2 Comparison of proposed document retrieval method with BM25, flat position index and inverted index baselines. . . . 62
6.3 Topics extracted by LDA algorithm (Due to space constraints we show only first 5 topics generated) . . . 70
6.4 Result of Topic - Aspect mapping. Based on number of words that are related to an aspect, we map the entire topic to corresponding aspect. . . . 72
6.5 Polarity score and sentiment of Review texts for the aspect - 'Battery' . . . 72
6.6 Polarity count for various Aspects . . . 73
6.7 Resut of aspect oriented sentiment analysis. . . . 73
6.8 Performance of our proposed aspect oriented sentiment analysis when compared with chosen baselines [152][145] . . . 73

6.9 Common NLP rules for extracting semantically valid phrases from tweets 81
6.10 Comparison of proposed method with BM25, flat position index, inverted index and entity based baselines for twitter hashtags - "#KashmirUnrest", "#SurgicalStrike", and "#Demonetisation" 84
6.11 Comparison of proposed method with BM25, flat position index, inverted index and entity based baselines for twitter hashtags - "#Election2016", "#MakeInIndia", and "#TrumpRussia" 84

CHAPTER 1

Introduction

The ideas and attempts to build machines which are "intelligent" is as old as the computing era termed as "modern computing". The Turing Test, proposed by Alan Turing in 1950s, was a very important milestone in this journey that established the means for qualifying the intelligent behavior of a computer system. This thought provoking proposal pawed the way to many attempts that explored the potential of building artificially intelligent systems mimicking human behavior. During this period, many algorithms were proposed which tried to impart the learning capability of human brains to computing systems leading to the development of a new specialized area of artificial intelligence called machine learning. Another interesting development was the introduction of Natural Language Processing (NLP), a sub-filed of artificial intelligence that studies the potential of mimicking the language processing capabilities of the human brain. NLP mainly concentrates in the development of algorithms capable of processing text corpora represented in human interpretable languages (natural languages). The current era, the World Wide Web (WWW), offers the largest collection of digitized text content spanning over application areas such as web portals, electronic journals, e-books, blogs etc. The other sources of text data includes digitized media content, books, documents produced and used by various business and government agencies etc. Most of the text documents available are typically present in a semi-structured or unstructured fashion without a proper definition of a schema (structure) specifying the syntactical / semantic meaning of its contents. Generally, text data is categorized into three types; viz.

- **Structured text data:** Only 5%of the total digital text data that is being generated belong to this category where the data have a formal structure for representation and management. The data stored and processed by relational database management systems is an example of this category. The data will be highly organized thus analysis and retrieval will be very easy.

- **Semi-structured text data:** The data that belong to this category may not be organized as compared to the structured text but they have some organizational mechanisms and arrangement that helps to store and retrieve. These data often contains some markers such as pre-defined or user defined tags to identify the key elements or semantic units and is estimated that another 5%of the total text data being generated belong to this category.

- **Unstructured text data:** 90% of the text data being generated around the world accounts to this category and such data does not have a specific format or internal structure that makes it very difficult to process, store and analyze. The text generated by social media platforms, web pages, blogs, electronic journals etc. belong to this category.

The major share of text data being generated around us are unstructured in nature and this type of data is a goldmine for the organizations. This is due to interesting patterns and insights which are latent in this data. Intelligent algorithms which can operate on this data can unearth interesting patterns and to leverage in potential business applications. Considering this fact, the research on text mining attained significant attention in the recent past among practitioners and academia. Researchers often find this tedious when the volume and complexity of the available data proliferates beyond conventional limits. There are still a lot of avenues where text data is yet to be exploited fully and thus the domain ontology are incomplete but manual creation of them are not feasible. There are other semantic structures such as concept graphs and hierarchies which are lighter versions of ontology but find several applications in semantic computing.

In text mining, concepts and key-phrases may be defined as a sequence of words that constitute real or imaginary entities. Concept and key-phrase extraction are non-trivial for text producing and consuming applications such as automated ontology generation [1], document summarization [2] and aspect oriented sentiment analysis [3] to name a few. **Concept mining** and **key-phrase extraction** tasks play a key role in text

mining because these concepts and key-phrases are highly useful for inducting meaning into other linguistic elements and ontological structures thus useful for many tasks such as information retrieval, opinion mining etc. The key-phrases convey more meaning and thus highly interpretable when compared to the typical word uni-gram based analysis. The large number of algorithms that operates on unstructured text produces word uni-grams as their outputs using well established mathematical and statistical approaches. For example, consider the ”topics” generated by the topic modeling algorithms in which the ”topics” are sequence of word uni-grams. The interpretation of such topics still remain as a hindrance to most of the text mining practitioners [85]. In real life scenarios, humans understand the concepts and key-phrases as a combination of words such as bi-grams and tri-grams. Splitting these bi-grams and trigrams into uni-grams cause its semantic meaning to be lost and that would generate ambiguous and irrelevant topics. For example, the concept ”network security attack” is more interpretable for humans in the context of ”computer security” compared to the word unigram - ”attacks”. Chopping them into ”network”, ”security” and ”attack” unigrams cause its semantics to be lost.

1.1 Research Motivation

The overwhelming amount of unstructured text data being produced by applications such as online social networks, digital journals, websites etc. show that text is considered as an important and most widely used data type. Identifying and leveraging highly useful patterns from these archives are always a tedious task. On one hand, there are algorithms which are purely statistical that do not consider the semantics or meaning of data they operate upon. On the other side, there are attempts reported that induces semantics into data archives with external knowledgebases and human curated ontologies. Manual creation of such ontologies from large archives of unstructured text is time consuming and may be inefficient. There are still a lot of avenues where domain ontologies are incomplete. Hence there is a need for developing novel algorithms that can operate on statistically generated outputs to induct semantics for generating concepts, key-phrases and concept graphs.

1.2 Problem Definition

Exploring the possibility of inducting semantic structures over statistically computed data abstractions from large unstructured data. Some of the potential dimensions explored are:

1. Bridging the gap between statistically generated text abstractions and domain ontologies that use external knowledgebases.
2. Possibility of inducing semantics into text abstractions that are generated by statistical algorithms that may aid generating semantically valid concepts and phrases from text.
3. Possibility of building concept hierarchies that connect concepts and keyphrases that may find applications in meaning aware computing paradigms.
4. Show applications of such hierarchies in real-world problems such as phrase query based document ranking and aspect oriented sentiment analysis.

1.3 Thesis Outline

Chapter 1 This chapter presents the background and motivations of proposed research and provides the thesis outline.

Chapter 2 This chapter provides a comprehensive description on related methods and methodologies used in this research.

Chapter 3 The main focus of this chapter is on extracting concepts from unstructured text data using statistical topic models that generate word uni-grams.

Chapter 4 The main theme of this chapter is concept lattice generation and relation extraction from unstructured text data. It describes how the unstructured text data can be represented in concept lattice and some of the commonly used relationships can be leveraged from it.

Chapter 5 This chapter gives the details of conceptual knowledge discovery using distributional semantics. An attempt on generating semantic phrase clusters is described in this chapter.

Chapter 6 This chapter shows some of the real-world applications which are developed using the algorithms proposed in this thesis.

Chapter 7 Conclusions and future works are mentioned in this chapter.

All the papers published in various international journals and conferences from the above works and references are presented at the end of this thesis.

CHAPTER 2

Literature Survey

UNSTRUCTURED text is considered to be the most widely generated datatype that contain highly useful patterns that can be leveraged and represented in the form of semantic structures that are consumed by meaning-aware computing systems. Mining of such patterns requires text understanding algorithms with deeper analysis capabilities and associated contextual information. Text mining algorithms attempts to leverage highly useful patterns mostly from unstructured text data. It is worth to note that in the recent past, a large array of algorithms introduced in the field of text mining and natural language processing utilizing the principles of mathematics and statistics. There is another category of algorithms that attempts to leverage context and induce semantics to text using human curated domain ontologies. In this context it is worth to walk-through the main concepts and method that we have adopted from various literatures on text mining and conceptual structure generation from unstructured text. The purpose of this chapter is to provide a brief introduction to the available research work conducted in the area of concept and key-phrase extraction from large unstructured text corpus. This chapter is organized as follows. A very brief introduction to text mining and natural language processing are given in Section 2.1. Section 2.2 discusses topic modeling in general but outlines the Latent Dirichlet Allocation (LDA) algorithm. The basic concepts and related literatures on concept extraction from unstructured text is discussed in Section 2.3. Major works on topical phrase discovery from text is discussed in Section 2.4 and in Section 2.5, methods of relationship extraction from unstructured text are discussed. Section 2.6 briefly discusses methods for representing linguistic elements such as words, phrases, sentences etc. in high dimensional vector

space also called distributional representation which is a recent development in natural language processing.

2.1 Text Mining and Natural Language Processing

The technological advancements in computing and communication technologies accelerated the rate of generating huge amount of data within and outside organizations. A vast majority of the contents of such archives are textual in nature due to a large number of text producing applications introduced. Analyzing and finding out the patterns which are latent in such repositories are important for the organizations to get insights for taking intelligent but crucial decisions. The process of identifying and extracting patterns, topics, keywords and concepts from unstructured text data is known as text mining and this became an active research area in computer science in the recent past due to the enormous volume of unstructured text produced. Earlier attempts of text mining attempted to extract certain standard and commonly occurring entities such as people, organizations, places etc. but those methods couldn't extract concepts and relationships between them. Text mining communities and researchers find it nontrivial to extract topics, keywords and concepts which may or may not explicitly cited in the text, for better knowledge representation and reasoning. Natural Language Processing (NLP) is a sub field of artificial intelligence that is primarily deals with the study of how computers process human languages. NLP is highly overlapping with text mining in which it helps the computers to "read" and process text. Natural language processing is used for processes such as automatically summarizing unstructured text, tagging parts-of-speech, extracting entities and associated relations etc.

2.2 Topic Modeling

In text mining, topic models are a suite of algorithms that take a large collection of unstructured text documents as their input and generate "topics" which are recurring patterns of words that are latent in the document. Topic modeling algorithms are widely used by text mining practitioners and research communities for discovering and analyzing hidden thematic structures from the unstructured text body. A good number of topic modeling algorithms are introduced in the recent past which varies in their method of

working mainly with the assumptions they adopt for the statistical processing. The earlier attempt for topic modeling started with Latent Semantic Analysis (LSA)[202][203] which is a matrix decomposition approach. LSA takes a document-terms matrix and decompose it into document-topic and topic-term matrix. Later, a probabilistic approach for LSA, called pLSA (Probabilistic Latent Semantic Anaysis)[204][205] had been reported in the literature that replaced the matrix decomposition approach. The idea behind the probabilistic approach is to derive a model with hidden topics that can generate the data that may be observed in the document-term matrix used by LSA approach. An automated document indexing method based on a latent class model for factor analysis of count data in the latent semantic space has been introduced by Thomas Hofman [18]. This generative data model called Probabilistic Latent Semantic Indexing (PLSI)[18], considered as an alternative to the basic Latent Semantic Indexing has a strong statistical foundation. The basic assumption of PLSI is that each word in a document corresponds to only one topic but in real-life, this may not be true for all the cases.

2.2.1 Latent Dirichlet Allocation (LDA)

Inspired from previous topic models, Blei et. al. introduced a new topic modeling algorithm known as Latent Dirichlet Allocation (LDA) [82]. This model assumes that a document contains multiple topics and such topics are extracted using a Dirichlet Prior process. The following section briefly describes the underlying principle of LDA [82]. Even though LDA works well on any discrete dataset, unstructured text is considered to be a good example to which the model can be best applied. The process of generating a document with n words by LDA can be described as follows [82]:

1. Choose the number of words, n, according to Poisson Distribution;
2. Choose the distribution over topics, θ, for this document by Dirichlet Distribution;
   (a) Choose a topic $T^{(i)} \sim \text{Multinomial}(\theta)$
   (b) Choose a word $W^{(i)}$ from $P\left(W^{(i)}|T^{(i)}, \beta\right)$

Thus the marginal distribution of the document can be obtained from the above process as :

$$P\left(d\right) = \int_{\theta} \left(\prod_{i=1}^{n} \sum_{T^{(i)}} P(W^{(i)})|T^{(i)}, \beta) P(T^{(i)}|\theta) \right) P(\theta|\alpha) d\theta \tag{2.1}$$

where, $P(\theta|\alpha)$ is derived by Dirichlet Distribution parameterized by α, and - $P(W^{(i)})|T^{(i)}, \beta)$ is the probability of $W^{(i)}$ under topic $T^{(i)}$ parameterized by β. The parameter α can be viewed as a prior observation counting on the number of times each topic is sampled in a document, before we actually seen any word from that document. The parameter β is a hyper-parameter determining the number of times words are sampled from a topic [82], before any word of the corpus is observed. At the end, the probability of the whole corpus D can be derived by taking the product of all documents' marginal probability as:

$$P(D) = \prod_{i=1}^{M} P(d_i) \tag{2.2}$$

Topic modeling algorithms such as Latent Dirichlet Allocation (LDA) [82], Probabilistic Latent Semantic Indexing (PLSI) [83] and Probabilistic Latent Semantic Analysis (PLSA) [84] are proved to be extensively useful in bringing out hidden themes from textual content which are used for further analysis. Given a large collection of text documents, these algorithms generate such themes by representing each document as a random mixture over latent topics and assume each topic be a probabilistic distribution over words. This idea is used in various scenarios such as document analysis and pattern identification in text mining, because text is considered to be the best data on which topic modeling can be easily applied. Majority of the topic modeling algorithms work on the assumption called "bag-of-words" to generate "topics". These statistically generated "topics" are sequences of word unigrams. The interpretation of such topics still remain as a hindrance to most of the text mining practitioners [85]. In real life scenarios, humans understand the concepts and key-phrases as a combination of words such as bi-grams and tri-grams. Splitting these bigrams and trigrams into unigrams cause its semantic meaning to be lost and that would generate ambiguous and irrelevant topics. For example, the concept "network security attack" is more interpretable for humans compared to the word unigram - "attacks". Chopping them into "network", "security" and "attack" unigrams cause its semantic meaning to be lost.

2.3 Concept Extraction from Unstructured Text

Concept mining and extraction plays a key role in text mining where each concept is a combination of words which are real or imaginary. Those concepts are used for tasks such as document retrieval [86, 87], classification [88], concept based sentiment analysis [89], semantic product search in e-commerce [90, 91] and concept hierarchy learning [102]. Search engines and document databases also use such concepts to locate related information from huge text archives. Thus concept identification is a key process in leveraging potential knowledge out of the data. Understanding, analyzing and summarizing key concepts from large volumes of text data is non-trivial and crucial in knowledge discovery process.

Automatic Concept Extractor (ACE), a system specifically designed for extracting concepts from HTML pages and making use of the text body and some visual clues on HTML tags for identifying potential concepts was proposed by Ramirez and Mattmann [9]. Even though this method could outperform some state of the art methods, dependency with HTML was a major drawback. Turney[10] proposed another system named GenEx, which employed a genetic algorithm supported rule learning mechanism for concept extraction. A system which extracts concepts from user tag and query log dataset is proposed by Parameswaran et.al.[11] which uses techniques similar to association rule mining. This method uses features such as frequency of occurrences and the popularity among users for extracting core concepts and attempts to build a web of concepts. Even though this algorithm can be applied to any large dataset, a lot of additional processing is required when dealing with web pages. A bag-of-word approach was proposed by Gelfand et.al.[12] for concept extraction from plain text and used these to form a closely tied semantic relations graph for representing relationships between them. They have applied this technique specifically for some classification tasks and found that their method produces better concepts than the Naive Bayes text classifier.

Dheeraj Rajagopal et.al.[13] introduced another graph based approach for commonsense concept extraction and detection of semantic similarity among those concepts. They used a manually labeled dataset of 200 multi-word concept pairs for evaluating their parser capable of detecting semantic similarity and showed that their method was capable of effectively finding syntactically and semantically related concepts. The main

disadvantage of that method is the use of manually labeled dataset and the creation of such dataset is time consuming and requires human effort. Another work reported in this domain is the method proposed by Krulwich and Burkey [14] which uses a simple heuristics rule based approach to extract key phrases from document by considering visual clues such as the usage of bold and italic characters as features. They have shown that this technique can be extended for automatic document classification experiments. A key phrase extraction system called Automatic Keyphrase Extraction (KEA) developed by Witten et.al[15] was reported in the concept extraction literatures which creates a Naive Bayes learning model with known key phrases extracted from training documents and uses this model for inferring key phrases from new set of documents. As an extension to this KEA framework, Song et. al.[16] proposed a method which uses the information gain measure for ranking candidate key phrases based on some distance and tf-idf features which was first introduced in [15]. Another impressive and widely used method was introduced by Frantzi et. al.[17] which extracts multi-word terms from medical documents and named as C/NC method. The algorithm uses a POS tagger POS pattern filter for collecting noun phrases and then uses some statistical measures for determining the termhood of candidate multi-words.

A method for automatic ontology learning from domain-specific short unstructured data is proposed by Yiming Xu et. al.[211]. In this work, the authors proposed a two-stage classification system to automatically learn an ontology from unstructured text data. The proposed approach first collects candidate concepts, which are classified into concepts and irrelevant collocates by our first classifier[211]. The concepts from the first classifier are further classified by the second classifier into different concept types[211]. Mikhail A. Khovrichev et. al. proposed an approach for context dependent synonym and concept extraction for training dialog systems[212]. This method was unsupervised in nature and capable of extracting synonym and concept extraction from natural language texts[212]. The method includes context-dependent algorithms that applies word embedding thus there is no need for tagged data and external resources[212]. An approach for extending ontologies in the nanotechnology domain using topic models and formal topical concept analysis has been reported very recently in the literature[213]. In this work, the authors use a phrase-based topic model approach and formal topical concept analysis on unstructured text in this domain to suggest additional concepts and axioms for the ontology that should be validated by a domain expert[213]. Matt Le

et. al. proposed an approach for inferring concept hierarchies from text corpora using hyperbolic embeddings [214]. They have attempted the task of inferring is-a relationships from large text corpora and proposed a new method combining hyperbolic embeddings and Hearst patterns[214].

2.4 Topical Phrase Discovery from Text

In this section we evaluate related researches on the dimension of inferring concepts from text data using topic modeling and critically review those works which are closely similar to our proposed method. The first notable work which explored beyond the traditional "bag-of-word" approach in topic modeling is the Bigram topic model [97]. In this model each topic word is generated from the distribution of words over a context which is given by a latent topic and the previous word. Later Wang et al. extended the basic Bigram topic model and proposed a new model called Topical n-gram [92] by incorporating a switching variable at each word position to denote the starting of a new n-gram. If this switching variable is not triggered, then the word will be considered as the continuation of a previously identified n-gram. The major shortfall of this model is that a post-processing is required to get the topic of a final word in a n-gram as the topic of the entire n-gram. This is because, in practical, words within an n-gram will not share same topic normally. Phrase discovering topic model [6] that uses pitman-yor process and TopMine [7] were two notable works that proposed algorithms for mining topical phrases from text documents. The former constructs a topic-word matrix before modeling topics but disadvantage of the approach was that creating such a matrix for large volume of text is often difficult. The latter approach makes use of a two stage process for modeling topics and mainly works with clinical documents. First it identifies phrases using some off-the-shelf tools and then trains a topic model with the identified phrases. Another work which uses topic models for generating multi-word phrases was the topical n-gram [8]. This makes use of some switching variable for identifying a new n-gram. The assumption of this method was that the words within an n-gram usually won't share same topic, which may not be true all the time.

Identifying these shortcomings of the Topical n-gram model [92], Lindsay et al. proposed PDLDA (Phrase discovering topic model) [93] which used the hierarchical Pitman-Yor

processes [98] for creating topic-word matrix. A topic segmentation model which incorporates word order [94] is later proposed by Jameel et al. Apart from topic detection, these models performs phrase segmentation which is computationally expensive when dealing with large text archives. More recent work in this dimension is reported in 2014 by El-Kishky et al. in which they proposed a topical phrase mining method called TopMine [99]. This framework uses a two step process - one for discovering phrases from text and next for training a traditional LDA model on these phrases and their assumption is that words in the same phrase should be assigned with the same topic. Other recent work reported is a framework proposed by Yulan He for extracting topical phrases from clinical documents [100]. In this two step work, the author first extracts medical phrases using an off-the-shelf tool and then train a topic model which takes a hierarchy of Pitman-Yor processes [98] as prior for modeling the generation of phrases of arbitrary length. Other notable work in this dimension is the Hierarchical Concept Topic Model (HCTM) [101] and Concept Topic Model (CTM) [103] which incorporates concepts into probabilistic topic models. HCTM employs a manual tagging mechanism to tag concepts in a document and involves assigning concepts to each word in a document. Then the semantic themes of a document content are revealed as a probability distribution over concepts and combine a hierarchy of human defined concepts with statistical topic models to combine the best of both. The major disadvantage of this work is that the tagging process is cumbersome and cannot be done without human annotators.

A cluster based iterative topical phrase mining framework [115] was recently introduced that present a novel framework for topical phrase mining. Their approach treats corpus as a mixture of clusters and each cluster is characterized by documents sharing similar topical distributions. Then this method iteratively performs phrase mining, topical inferring and cluster updating until a satisfactorily final result is obtained. Another notable work in this dimension was introduced by Xin Li and Wei Jin [116]. They proposed a semantic concept latent dirichlet allocation and semantic concept hierarchical dirichlet process based approaches by representing text as meaningful concepts rather than words. The authors implemented the algorithms in discovering new semantic relation between concepts from text documents. The method improved the search quality when compared with other LDA or HDP based approaches. Another very recent work was introduced by Xu Kang et. al. [117] that incorporates Wikipedia concepts and categories as prior knowledge into topic models. Their work utilizes entity knowledge,

concepts and categories in Wikipedia as prior knowledge into topic models so that it discover more coherent topics. Their method not only modeled the relationship between words and topics, but also utilizes knowledge of concept and category to model semantic relationship between them.

Baoji Li et. al. proposed phrase topic model based on the LDA model, which integrates a regular expression constraint condition[215]. The authors claimed that the proposed model makes the topic more meaningful and interpretable based on a limited increase in the dimensions of the vocabulary[215]. An unsupervised approach for concept association analysis in text collections was reported in the literature recently[216]. This approach proposed by Pavlo Kovalchuk et. al. propose a solution that integrates currently dispersed principles in a new unsupervised knowledge discovery process combining principles from topic modeling and formal concept analysis, thus not requiring prior domain knowledge to be applied in large document collections[216].

2.5 Relationship Extraction from Text

Relation Extraction (RE) is a subtask Information Extraction (IE) that aims at extracting relevant and potentially useful patterns or information from humongous data that is being generated day by day. The sheer volume and heterogeneity of data makes it difficult to analyze and extract these patterns manually. Thus we need automated techniques for this process. Natural Language Processing (NLP) is one of the major areas that address this issue by scanning natural language texts and extract useful patterns. IE tasks, specifically the relation extraction have a long history and goes back to late 1970s. However successful commercial systems were introduced in 1990s. In this section, we discuss some of the recent approaches in relation extraction. In addition, we also throw light on state-of-the-art in Formal Concept Analysis (FCA) based concept lattices generation approaches.

Informally, we can group all the relation extraction approaches introduced in literatures into five categories hand built patterns, bootstrapping methods, supervised methods, distant supervision and unsupervised approaches. Hand built patterns uses handcrafted rules for extracting potentially relevant relation words from text and one very notable work was introduced by Marti. A. Hearst known as Hearst Patterns [29]. One issue with

this approach is that it is difficult to write all set of possible rules and for other tasks such as meronyms extraction, the set of rules will be different. But still, a good number of extensions are being reported that uses Hearst Patterns as their foundation for relation extraction [31][32][33]. Another category of relation extraction are the bootstrapping based approach in which a specific set of seed relation instances are created and these are used for searching for new tuples. One such approach for extracting ”author book” relation was the DIPRE [34]. Later, another system was introduced that uses the idea of bootstrapping to Snowball [35] that extracts ”organization - location” relation pairs. The limitations with above said algorithms are with the specific relations they can only deal with. Users have to specify the type of relation they need to work with such as ”author-book” or ”organization - location” etc. Later, TextRunner [36] was introduced in the domain of relation extraction which can learn relations, classes and entities from a corpus in a self-supervised manner. This approach first tags the training data as positive and negative and then trains a classifier on the data to generate potential relations and entities. Another two stage bootstrapping algorithm [37] was proposed by Ang Sun. In the first step, the algorithm uses a bootstrapping method to scan the tuples and in the second stage, it learns relation nominals and contexts.

More recently, supervised and semi-supervised approaches are found to be promising and a very good number of literatures have been reported in the area of relation extraction that uses deep learning techniques for identifying relation patterns. Very recently a neural temporal relation extraction approach [38] has been introduced in which the authors experimented with neural architectures for temporal relation extraction. They showed that neural models that take tokens only outperform state-of-the-art hand engineered feature based models in performance. They also reported that encoding relation arguments with XML tags performs superior than a traditional position based encoding. Another notable approach was introduced that attempts neural relation extraction with selective attention over instances [39]. This work employs Convolutional Neural Networks (CNN) to embed the semantics of sentences. Experimental results show that this model could make full use of all informative sentences and achieved significant and consistent improvement on relation extraction task. An approach for extracting relationships from clinical text was introduced [40] that exploited CNN to learn features automatically that reduces the dependency on manual feature engineering. They showed that CNN can be effectively used for relation extraction in clinical text without being

dependent on expert knowledge on feature engineering. Our proposed relation extraction method uses machine learning technique to classify Hearst patterns [29] (”such as”, ”or other”, ”and other”, ”including”, ”especially” etc.) and other potential indications of relations such as ”is-a” among textual concepts.

2.6 Distributional Representation of Phrases

This section briefly outlines the necessary background on distributional representation of words and phrases which are current hot research areas in text mining and natural language processing. This area of research is attempting to develop methods for measuring the level of semantic similarities among linguistic elements in large samples of unstructured data. The idea of distributional representation of words and phrases got significant attention from text mining and natural language processing practitioners as these representations could project the semantic similarities of those linguistic units in high dimensional vector space. The seminal work in this dimension that explored beyond traditional syntactic and probabilistic approach was introduced by Tomas Mikolov which represented word uni-grams in high dimension vector spaces [162]. This work contributed two strong architectures - Continuous Bag of Words (CBOW) and Skip-gram - where the former can predict the current word based on the context and the latter can predict the context based on a given word. These models were initially used to work with word unigrams, later methods for distributed representation of phrases [163], sentences, paragraphs and even documents [164] have been reported.

Based on the thorough literature survey as discussed above, it is learned that the proposed problem statement is highly relevant in the context of conceptual knowledge discovery from large bodies of unstructured text documents.

CHAPTER 3

Concept Extraction and Concept Hierarchy Generation

DUE to the rapid growth of text producing and consuming applications, numerous tools and techniques were introduced in the recent past for extracting useful patterns from unstructured text. Often these patterns represent actionable knowledge aiding intelligent decision making and hence crucial for many organizations. The platforms such as social networks, e-commerce websites, blogs and research journals repositories having large quantities of unstructured data are some of the indicative application areas of such algorithms. With the advent of semantic web, various tools and techniques such as Web Ontology Language (OWL) were introduced for representing and utilizing abstract semantic knowledge pertaining to any specific application context. Concept graphs are one such representation where concepts are defined as a sequence of words that constitute real or imaginary entities. Extraction of such entities are critical for applications such as automated ontology generation [1], document summarization [2] and aspect oriented sentiment analysis [3] to name a few. Concept hierarchies and graphs are such representation mechanisms that are lighter versions of ontologies but find many applications in meaning aware computing. Constructing domain ontologies are labour intensive as the identification of entities, concepts, key-phrases and relationships unstructured text is often very difficult. There were attempts reported which introduced semi-automated methods for ontology generation but requires laborious human effort. Conceptual structures such as concept graphs and hierarchies are lighter versions of ontologies. The concept hierarchy learning algorithms attempt to extract and connect human interpretable concepts from unstructured text in a hierarchical fashion.

There exist many algorithms that operate on text and produces abstractions using well developed mathematical and statistical techniques (classification, clustering, topic modeling etc.) without considering the meaning of the input data. There is another category of algorithms with varying levels of success that uses external knowledgebases such as Probase and DBPedia for bringing in semantics to the textual data processing. This chapter discusses two novel algorithms proposed for bridging this gap. The first algorithm proposed attempts to generate concepts and key-phrases from ”topics” that are obtained by topic modeling algorithms. The second algorithm utilizes the concepts generated to make concept hierarchies which can be viewed as light-weight versions of domain ontologies. Experiments conducted on real-world datasets show that the algorithms discussed in this chapter can generate close to real-world and human interpretable concepts and key-phrases and make concept hierarchies from the statistically obtained topics models. The remainder of this chapter is organized as follows: Section 3.1 discusses how semantically valid and close to real-world concepts are identified and how the hierarchies are constructed from topics which are generated by topic modeling algorithm. Experimental setup and evaluation details are shown in Section 3.2 and Section 3.3 concludes this chapter.

3.1 Extracting Concepts and Concept Hierarchies from Statistical Topic Models

In the area of text mining, topic models or specifically probabilistic topic models are suite of algorithms which got wider recognition for its ability to leverage hidden thematic information from huge archives of text data. Text mining researchers use topic modeling algorithms such as Latent Semantic Analysis (LSA) [20], Probabilistic Latent Semantic Indexing (pLSI) [21], Latent Dirichlet Allocation (LDA) [22] etc. extensively for bringing out the themes or so called ”topics” from unstructured text data. Among the available topic modeling algorithms, LDA [22] got significant attention due to the assumptions it make, the useability and implementation front. Even though the power of LDA algorithm has been extensively used for leveraging topics, very few studies have been reported for generating conceptual structures from the statistically generated topics. The proposed algorithms in this chapter are attempts to address this gap. The algorithms use LDA to generate topics and then identifies concepts and key-phrases

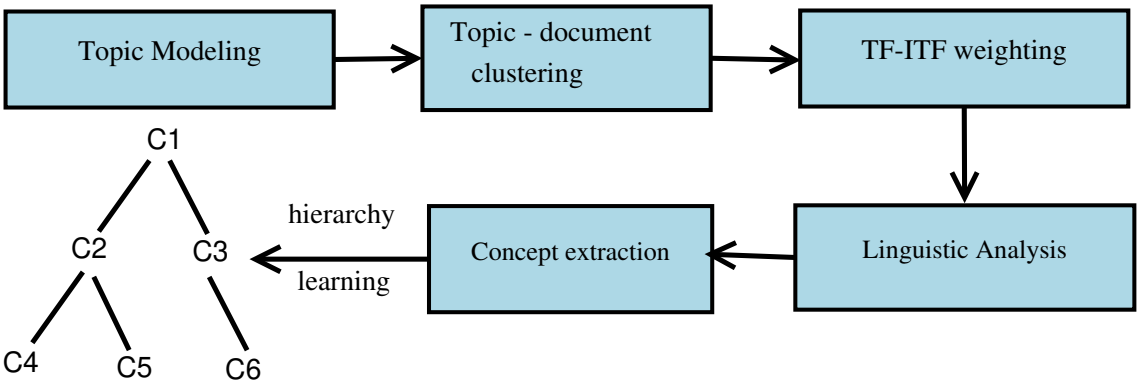


FIGURE 3.1: Workflow of the proposed concept hierarchy learning method

using statistical weighting scheme and lightweight linguistic method. The overall work flow of the proposed approach is depicted in Fig. 3.1. The proposed framework can be divided into 2 modules (i) Concept and key-phrase extraction and (ii) Concept hierarchy generation. The concept extraction module extract concepts from topics generated by LDA algorithm and the concept hierarchy generation module builds a hierarchy of extracted concepts by inducing subsumption relation between concepts and key-phrases. Detailed explanations of these modules are given below.

3.1.1 Concept Extraction

This section introduces a topic to concept mapping procedure for leveraging potential concepts and key-phrases from statistically computed topics which are generated by the LDA algorithm [22]. The first step of the proposed framework deals with pre-processing the data which is meant for removing noises. Latent Dirichlet Allocation[22] algorithm is executed on top of this preprocessed data which in turn generate topics through a statistical process. The LDA generated topics were considered and for each topic, this method creates a topic - document cluster by grouping the documents which generated such a topic and the same process has been executed for all topics under consideration. This work proposes a new weighting scheme called $tf - itf$ (term frequency - inverse topic frequency) which is used for finding out weights of words in each topic. This weighting scheme is introduced to filter out the relevant candidate topic words. Term frequency (tf) is the total number of times that particular topic word comes in the topic - document clusters. Normalized term frequency, N_{tf} of a topic word T_w can be

calculated as:

$$N_{tf} = \frac{count(T_w)\ in\ C_{td}}{count(total\ terms\ in\ C_{td})} \tag{3.1}$$

Inverse topic frequency itf is calculated as:

$$itf = \frac{count(total\ terms\ in\ C_{td})}{count(documents\ with\ T_w)} \tag{3.2}$$

$tf - itf$ is calculated using the following equation:

$$tf - itf = N_{tf} * itf \tag{3.3}$$

where N_{tf} denotes the normalized term frequency, T_w denotes a topic word, C_{td} means topic - document cluster and itf denotes inverse topic frequency.

This step is followed by a sentence extraction process in which all the sentences that contain the topic words having high tf-itf weight are extracted. Next, a POS (Parts of Speech) tagging was applied on these sentences to extract noun and adjective tags. This method extracted only noun and adjective tags because the key-phrases and concepts are generally a combination of these tags. In linguistic pre-processing step, we combine those words which are tagged as Noun and Adjectives using the patterns: (Noun + Noun), (Noun + Adjective) and (Adjective | Noun) + Noun. Concept identification is the last step in the process flow in which the term count of all the combinations of Noun + Noun, Noun + Adjective and (Adjective / Noun) + Noun. A positive term count implies that the chosen multi-word can be a potential concept / key-phrase and if we get a zero term count, then that multi-word can be ignored. The newly proposed algorithm for extracting the concepts is shown in Algorithm 1. The input to the algorithm is a collection of topics tc and the output is a collection of concepts identified. For every topic t, the algorithm first creates a topic-document cluster C_{td} and then for every topic-document cluster C_{td}, the $tf - itf$ weight is computed as given in the equation 3.3. Then for every topic t, the algorithms choose the first n words with highest $tf - itf$ score and the sentences from the original corpus that contains these top weighted words are extracted. The parts-of-speech tagging will be performed on these sentences to generate the tags. As the aim of the algorithm is to extract the major concepts and key-phrases, it only selects the ”noun” and ”adjective” tags. All the combinations of $noun + noun|adj + noun$ are taken and the number of times these words occur in the

original dataset is taken. If it is a non-zero positive value, then the word can be counted as a potential concept or else the algorithm will discard the same. These steps will be repeated for all the files in the corpus and at the end of the process, it returns a collection of concepts and key-phrases.

Algorithm 1 : Concept Extraction

function ExtractConcepts(tc)
    $\forall\ t$, create C_{td}
    $\forall C_{td}$, compute $tf - itf$ weight
    $\forall\ t$, choose n words with highest $tf - itf$
    $S[\ ]$ = sentences with top $tf - itf$ words
    $POS\ tag(S)$
    $W[\ ] = (NNP, NNS, NN, JJ)$
    $MW_c[\ ] = noun + noun|adj + noun$

    while $|MW_c| \neq 0$ **do**
        $termCount(MW)\ \forall\ MW$ in MW_c
        if $T_c > 0$ **then**
            Add MW into C
            Remove MW from MW_c
            Fetch next MW from MW_c
        else
            Remove MW from MW_c
            Fetch next MW from MW_c
        end if
    end while
end function

3.1.2 Concept Hierarchy Learning

In this module generates a hierarchy of leveraged concepts using a type of co-occurrence called "subsumption" relation. Subsumption relation is found to be simple but very effective way of inferring relationships between words and phrases without using any training data or clustering methods. The basic idea behind subsumption relation is very simple : for any two concepts C_a and C_b, C_a is said to be subsume C_b if 2 conditions hold. $P(C_a|C_b) = 1$ and $P(C_b|C_a) < 1$. To be more specific, C_a subsumes C_b if the documents which C_b occurs in are a subset of the documents which C_a occurs in. Because C_a subsumes C_b and because it is more frequent, in the hierarchy, C_a is the parent of C_b. This can also be observed from the conditional probability definition: $P(C_a|C_b) = P(C_a \cap C_b)/P(C_b)$; when C_b is a subset of C_a ($C_a \cap C_b$ becomes C_b), the conditions $P(C_a|C_b) = 1$ and $P(C_b|C_a) < 1$ holds good. The proposed concept hierarchy

learning algorithm is shown in Algorithm 2. The input to the algorithm is a collection of concepts extracted using Algorithm 1, which is denoted by C. The concept hierarchy learning algorithm first chooses a pair of concepts, say, C_a and C_b and then compute two probabilities $P(C_a|C_b)$ and $P(C_b|C_a)$. Then it checks if two probabilities hold which are $P(C_a|C_b) = 1$ and $P(C_b|C_a) < 1$. If this returns to true, then C_b will become child concept and C_a becomes the parent. If this returns false, then the algorithm proceed with choosing another pair of concepts and repeats the step. The algorithm will return concept hierarchies once it execute completely on the specified dataset.

To better understand these algorithms with a toy example, consider a dataset con-

Algorithm 2 : Concept Hierarchy Learning

```
1: function LearnHierarchy(C)
2:     Choose pair of concepts, say C_a and C_b
3:     Compute P(C_a|C_b) and P(C_b|C_a)
4:     if P(C_a|C_b) = 1 and P(C_b|C_a) < 1 then
5:         Assign C_a as the parent of C_b
6:     else
7:         Fetch next concept pairs
8:         Goto step 2, repeat ∀ C_a, C_b
9:     end if
10: end function
```

taining 2 text documents about "artificial intelligence". Firstly, the topic modeling algorithm will be executed on this dataset to generate topics and let's assume that the number of topics are set as five and number of words in each topic as ten. Topic modeling generate topics $t1 = artificial, language, intelligence, machine, etc.$, $t2 = learning, applications, supervised, deep, etc$ and so on upto $t5$. We need to first find out out of all these topics, which words are more important and for that the $tf - itf$ weight will be calculated as given in 3.3. Once this is done, then we extract the sentences that contains the words in the topics that has highest $tf - itf$ score (in this toy example, we assume we take top 5 words). Consider that the first sentence extracted is like *"Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed"*. On running the parts-of-speech-tag, we get the output *Machine_NN learning_NN, is_VBZ, an_DT application_NN, of_IN, artificial_JJ intelligence_NN, AI_NNP, that_WDT, provides_VBZ, systems_NNS, the_DT ability_NN, to_TO, automatically_RB, learn_VB, and_CC, improve_VB, from_IN, experience_NN, without_IN, being_VBG, explicitly_RB, programmed_VBN*. Since we

are interested to extract concepts and key-phrases, we filter out only those words tagged as Noun and Adjective. Then these words are combined. For example, in this example two words *Machine* and *Intelligence* - two topic words which are combined to get the phrase - *MachineLearning*. This step will be completed for the entire dataset to get a collection of key-phrases and concepts.

To understand the concept hierarchy learning algorithm with the output from the above example can be treated as the input. With the concept and key-phrase collection given as the input, as per the subsumption hierarchy definition given in section 3.1.2, it will assign a concept to a parent concept according to the probability conditions. For example, in this example, the concept *ArtificialIntelligence* will be the parent of the concept *MachineLearning* and similarly *SupervisedLearning* will be the child of the concept *MachineLearning*.

3.2 Experimental Setup

This section discusses the implementation details of proposed algorithms for topic modeling guided concept extraction and concept hierarchy generation. The details are given in following sections.

3.2.1 Dataset Collection and Pre-processing

This experiment uses publicly available datasets such as Reuters Corpus Volume 1 dataset[24] and BBC News Dataset[25] for the experiments. Reuters is the world's biggest international news agency and cater different news and related information through their website, video, interactive television and mobile platforms. Reuters Corpus Volume 1 is in XML format and is freely available for research purpose. Text messages are extracted by a thorough pre-processing such as removing XML tags, URLs and other special symbols and then created a new dataset exclusively for our experiment. BBC provides two benchmarked news article datasets which is freely available for machine learning research. The general BBC dataset consist of 2225 text documents directly from their website corresponding to stories in five areas such as business, entertainment, politics, sports and technology, from 2004 to 2005. A thorough pre-processing such as

stemming, and removal of stop-word, URLs and special characters on this dataset and made an experiment ready copy of the original dataset.

3.2.2 Modeling Topics

Latent Dirichlet Allocation (LDA)[22] algorithm has been applied on the pre-processed dataset to leverage topics for this experiment. The number of iterations is set to 300 as Gibbs sampling method usually approaches the target distribution after 300 iterations. This was found using trial and error method of running the experiment with different values for the epoch parameter. The number of topics is set to 50 for the initial experiment but later the value was changed to 100, 150 and 200 respectively.

TABLE 3.1: Top 10 topic words from Topic 1 and Topic 2 along with their computed TF-ITF weight

Topic 1	**TF-ITF Weight**	**Topic 2**	**TF-ITF Weight**
web	0.0048	set	0.0047
search	0.0048	software	0.0032
online	0.0047	virus	0.0028
news	0.0046	users	0.0027
google	0.0033	firms	0.0025
people	0.0032	microsoft	0.0025
information	0.0032	security	0.0022
internet	0.0029	windows	0.0022
website	0.0027	file	0.0013
users	0.0020	programs	0.0011

3.2.3 Topic - Document Clustering

In this step, we consider each topic and clustered top 50 documents which contributed the creation of that specific topic, using the probability distribution given by LDA. As an outcome, 50 topic-document clusters were created that contain documents which generated the topics.

TABLE 3.2: Top 10 topic words from Topic 3 and Topic 4 along with their computed TF-ITF weight

Topic 3	**TF-ITF Weight**	**Topic 4**	**TF-ITF Weight**
system	0.0064	site	0.0042
music	0.0045	net	0.0038
devices	0.0043	spam	0.0035
players	0.0035	mail	0.0028
media	0.0032	firm	0.0027
digital	0.0027	data	0.0024
market	0.0024	attacks	0.0019
technology	0.0022	network	0.0018
consumer	0.0021	web	0.0016
technologies	0.0018	research	0.0014

3.2.4 TF-ITF Weighting

Here, $tf-itf(term\ frequency - inverse\ topic\ frequency)$ weight of each word in every topic was calculated using Eq.(3.1), Eq.(3.2) and Eq.(3.3) to find out highly used topic words in the collection. Table 3.1 and 3.2 shows topic words along with their computed tf-itf weight.

TABLE 3.3: Top concepts extracted against first 4 topics

Top 5 concepts generated from Topic 1	**Top 5 concepts generated from Topic 3**
web search	music players
search engine	digital media
google news	digital technology
online news search	consumer devices
google search engine	market system
Top 5 concepts generated from Topic 3	**Top 5 concepts generated from Topic 4**
software users	spam mail
virus programs	spam website
windows security	network research
software forms	research firm
microsoft programs	website attacks

3.2.5 Sentence Extraction & POS Tagging

In this step, topic words having highest tf-itf weights are considered and sentences containing those topic words are extracted from the topic - document clusters. Then parts-of-speech tagging was performed identify words tagged as nouns and adjectives from these sentences as our aim is to extract potential concepts and key-phrases from the repository. For this experiment, Natural Language Toolkit (NLTK) [168] has been used which contains libraries for Natural Language Processing for Python programming language.

3.2.6 Linguistic Processing & Concept Identification

All words which are tagged as Nouns(NN | NNP | NNS) and Adjectives (JJ) are filtered out and all possible combinations of $Noun + Noun, Adjective + Noun$ and $(Noun/Adjective) + Noun$. Table 3.3 shows the top concepts identified for each topics generated by the topic modeling algorithm. Due to space constraints, the table shows concepts identified for the first four topics. These concepts have been identified using the natural language processing rules mentioned in Algorithm 1. For example, consider the first concept "web search". The parts-of-speech tags for the words "web" and "search" are "noun" and "noun" respectively. On combining the words, a new concept is identified which is "web search". Similarly the other concepts are also identified using Algorithm 1. The term count for each of these multi-word term is calculated against the original corpus and a positive term count implies that the corresponding multi-word can be a potential concept / key-phrase and we eliminate the term if we get a zero term count. This process has been repeated for all the multi-words we have filtered out using the linguistic patterns.

3.2.7 Concept Hierarchy Learning

Concept hierarchy generation algorithm attempts to create a subsumption hierarchy[5] by identifying "is-a" relationship between pair of concepts identified by Algorithm 1. Subsumption relation is considered as an important relationship in any ontological structure and two probability conditions are calculated for the same. For any given

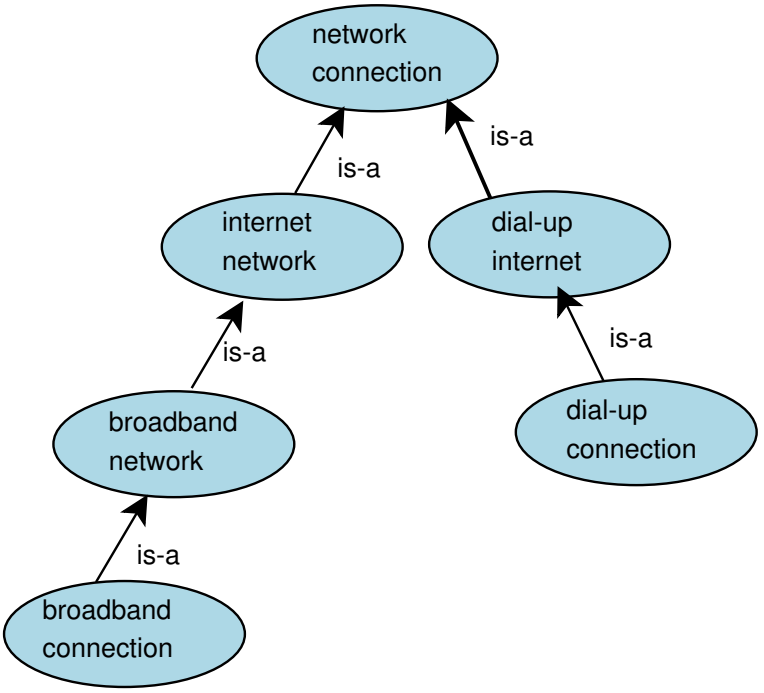


FIGURE 3.2: A partial snapshot of one of the concept hierarchy

concept pairs, $P(C_1|C_2)$ and $P(C_2|C1)$ are calculated in order to establish a subsumption relation. To connect two concepts using this, $P(C_1|C_2)$ must be 1 and $P(C_2|C1)$ should be less than 1. For example, consider concept pair C_1 and C_2, C_1 subsumes C_2 if the documents in which C_2 occurs is a subset of the documents which C_1 occurs in. For better understanding, consider two phrases identified by Algorithm 1: *dial-up internet* and *network connection*, the subsumption algorithm computes $P(dialup\ internet|network\ connection)$ and $P(network\ connection|dial-up\ internet)$ and found that the number of documents in which *dialup internet* occurs is a subset of number of documents in which *network connection* occurs. That means there exists a subsumption relation between these two concepts and *dialup internet* concept may be subsumed by *network connection* concept. This process has been repeated for all concepts in the collection, and a part of such a hierarchy generated using our proposed algorithm is shown in Fig. 3.2.

3.2.8 Evaluation of Results

The results produced by the algorithms proposed in this thesis are evaluated in terms of precision, recall and F1 score. A human created and curated concept and key-phrases

TABLE 3.4: Comparison of ACE, ICE and our proposed method

Algorithm	Precision	Recall	F1
ACE	0.2372	0.2689	0.2517
ICE	0.7113	0.8147	0.7595
Proposed	**0.8165**	**0.8901**	**0.8516**

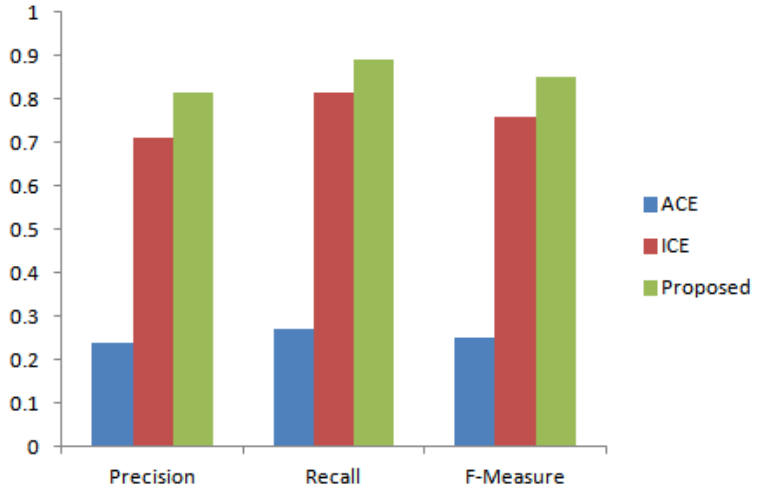


FIGURE 3.3: Precision, Recall and F-measure comparison of ACE, ICE and proposed method

collection from the datasets used for the experiment was generated beforehand to verify the quality of machine generated concepts and key-phrases. Precision computes the fraction of machine extracted concepts/key-phrases that are also human generated, and recall measures concepts/key-phrases which are extracted by proposed algorithms that are also human authored. In information retrieval, it is estimated that achieving high precision and recall at same time is difficult and using a measure called F1, we can balance these two. Here, true positive is defined as the number of overlapped concepts between human generated concepts and concepts extracted by our proposed algorithm, false positive is the number of extracted concepts that are not truly human authored concepts and false negative is the human authored concepts that are missed by the concept extraction method. Using these measures, we have compared our proposed method against some of the existing concept extraction algorithms and the result is shown in Table 3.4. A graphical comparison has been given in Figure 3.3. From the performance graph shown in Figure 3.4, it is clear that our proposed algorithm extracts better human interpretable concepts and key-phrases as the number of topics are increasing. This shows that the proposed algorithms outperforms the baselines when extracting

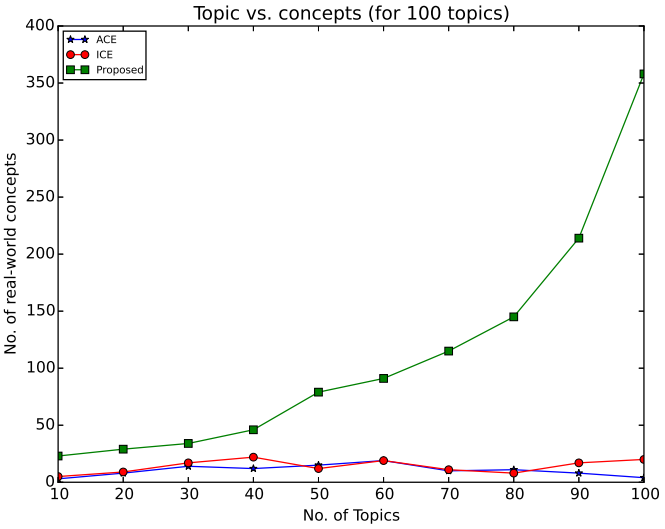


FIGURE 3.4: Performance of concept extraction algorithm on first 100 topics generated

real-world concepts from large number of statistically generated topics.

3.3 Conclusions

This chapter proposed two algorithms - one for extracting close to real-world concepts from large collection of unstructured text corpus and the other for generating a concept hierarchy using subsumption relation between concepts and key-phrases in unstructured text. Experiments conducted on large real-world datasets such as Reuters and BBC news corpus show that the proposed algorithms outperform some of the already available methods. The next chapter discusses how concept lattices can be used to generate conceptual structures and also how interesting relations can be extracted from unstructured text data. The chapter proposes methods for identifying commonly occurring relations that connects concepts in unstructured text data and attempts to learn such relations using machine learning techniques

CHAPTER 4

Concept Lattice Generation and Relationship Extraction

AUTOMATED concept hierarchy learning from unstructured text is a critical step in building conceptual structures, similar to ontologies needed for building meaning aware / semantic text mining applications. Concept hierarchy learning algorithms extract concepts from text and connect those concepts using potential relations that exist among them. Such hierarchies may find useful applications in concept based ontology generation [26], concept guided document summarization [27], and concept guided information retrieval [28] to name a few. In text mining, many methods with varying degrees of usability are available for conceptual analysis and concept hierarchy generation. Formal Concept Analysis (FCA) is one such method that uses well established mathematical and lattice theory principles. Formal Concept Analysis (FCA) is defined as a mathematical model or framework based on lattice theory [45] which is well suited for knowledge engineering and processing tasks. FCA makes use of an incident matrix for data analysis in which each row corresponds to attributes and the filed value denotes the relationship between them. FCA takes this incident matrix as input and then outputs concept lattice that reflects generalization and specialization between the derived formal concepts from the incidence matrix [46]. These formal concepts are extensively used for knowledge processing tasks containing distinct extents and intents (set of objects and their relationships). These relations are represented in the form of a formal context, F = (X, Y, R) where X is a set of objects, Y is a set of attributes and R is a binary relation between them. From this given context, FCA creates a set of objects (A) and the set of all attributes (B) that are common for these objects. The concept lattice that is built from incidence matrix (context table) determines the hierarchy of formal concepts that follows partial ordering principle as (A1, B1) $\leq$ (A2, B2) iff A1 $\leq$

A	B	C	D	E
	Latin Amer...	Asia	Europe	Middle East
Air Canada	X		X	
Air New Zealand	X	X		
Air India		X		X

FIGURE 4.1: Formal context showing airlines and their sector of operations

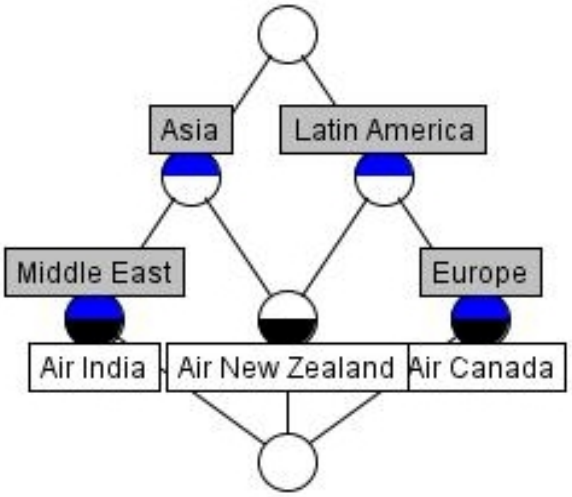


FIGURE 4.2: Concept lattice generated for formal context given in Fig. 4.1

A2 (B2 $\leq$ B1) and then gives generalization and specialization between the concepts. That is (A1, B1) is more specific than (A2, B2). The attributes of each formal concept are inherited from the most general maximum node and the objects are inherited from the most specific minimum node [26]. The attribute implications are represented in the form of A $\rightarrow$ B over the set Y. There were several algorithms developed for generating concept lattices [52][53][54][55]. An example of a formal context showing airlines and their sector of operations and corresponding concept lattice visualization are shown in Fig. 4.1 and Fig. 4.2 respectively. Figure 4.1 represents a formal context showing airlines and their sectors of operations for three airlines - Air Canada, Air New Zealand, and Air India. The elements on the left side (that is column A) are called formal objects and the elements at the top (that is first values in column B, C, D and E) are called formal attributes. If a formal object has a formal attribute, then the corresponding cell of the table will be denoted with a cross symbol. That means, if a cell is blank then that object has no corresponding attribute. So, in the context shown in Figure 4.1, Air India flies to Middle East but does not fly to Europe. The corresponding concept lattice generated is shown in Figure 4.2 which can be interpreted as follows. To retrieve the extension of a concept, one begins at the node where the attribute is labeled and traces

all paths which lead down from the node. Any objects one meets along the way are the objects which have that particular attribute.

This chapter proposes a method for identifying commonly occurring relations that connects concepts in unstructured text data and attempts to learn such relations using machine learning techniques. Specifically, the proposed approach can identify and learn commonly used relationships in textual data such as subsumption (”is-a”), Hearst patterns [29] (”such as”, ”or other”, ”and other”, ”including”, ”especially” etc.) and other potential relations among concepts and key-phrases. This approach makes use of Formal Concept Analysis (FCA) [30] which is a well established mathematical theory of analyzing data, to form context table and concept lattices [30] from the identified concept and relations. These lattices may then be used for the generation and enrichment of domain ontology which are the backbone of intelligent and meaning aware computing systems. The rest of this chapter is organized as follows: Section 4.1 explores concept lattice generation and relationship extraction method from unstructured text. Experimental details and evaluations are discussed in Section 4.2 and Section 4.3 concludes the chapter.

4.1 Concept Lattice Generation and Relationship Extraction from Unstructured Text

In this section, we outline our proposed approach for identifying and extracting relationships that connect entities or concepts which are extracted from unstructured text. Considering the complex sentence and language structures, concept and relation extraction is an extremely difficult task in natural language processing and information retrieval. Even though there were many attempts reported on how to extract them, majority of the works were heavily dependent on the specific corpus that chosen for experiment. In previous work, we have also attempted the concept extraction task which is guided by a topic modeling process that suits well on any plain text corpus [47, 48]. In this work, our main focus is on relation extraction task and thus the process of concept extraction is not emphasized. Here, we are using an off-the-shelf tool for identifying potential entities, phrases and concepts from our static document corpus and then implements our relationship extraction algorithm on top of it to extract relation patterns.We

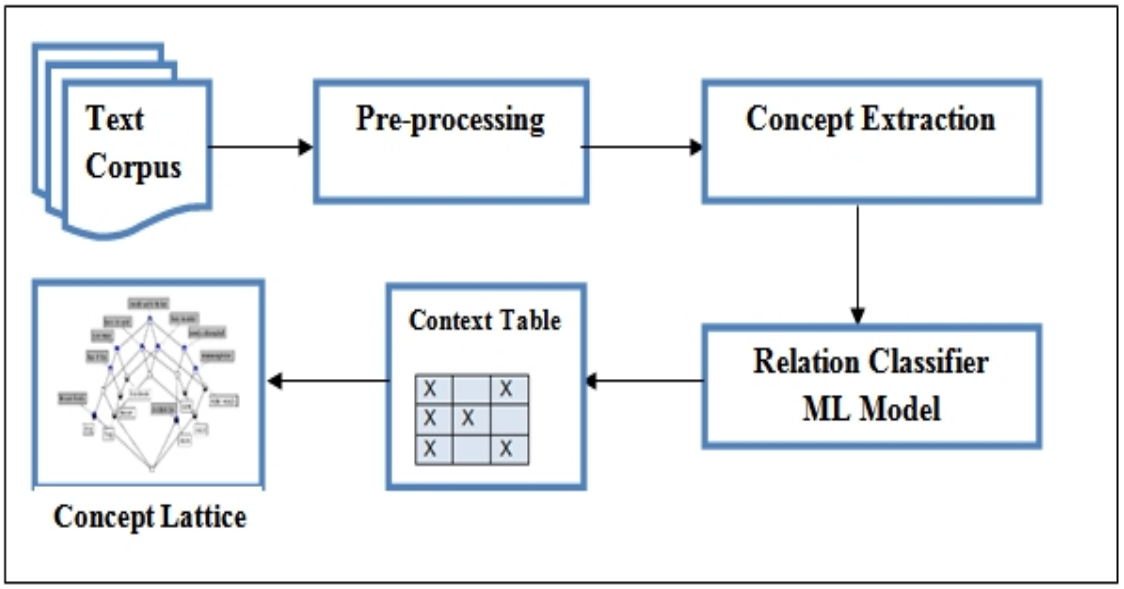


FIGURE 4.3: Overall workflow of the proposed approach

now define the problem formally. Relation extraction problem is the task of detecting and classifying semantic relationships that connects entities or phrases or concepts in a corpus of interest. Given a static document corpus D, the relationship extraction task identifies valid relation words that connect two concepts together. For example, consider the sentence, "Alzheimer is a degenerative disease". The words "Alzheimer" and "degenerative disease" are potential concepts in a medical text document. The relationship extraction method identifies "is-a" as a potential relation that connect these two concepts. Given a static document corpus $D = d_1, d_2, ..., d_n$ that contain tagged concepts $C = c_1, c_2, ..., c_n$, our problem is to identify semantically distinguishable relations that connects $c_1, c_2, ..., c_n$.We also address the problem of representing this knowledge using Formal Concept Analysis (FCA) which is a widely used knowledge representation framework that comes with well implemented mathematical models. An overall workflow of the proposed approach is shown in Figure 4.3.

4.2 Experimental Setup

In this section, we describe our experimental setup and a detailed evaluation of the results. Two separate experiments are conducted, one for extracting semantically valid relationships that connects concepts extracted. The second one is for validating the

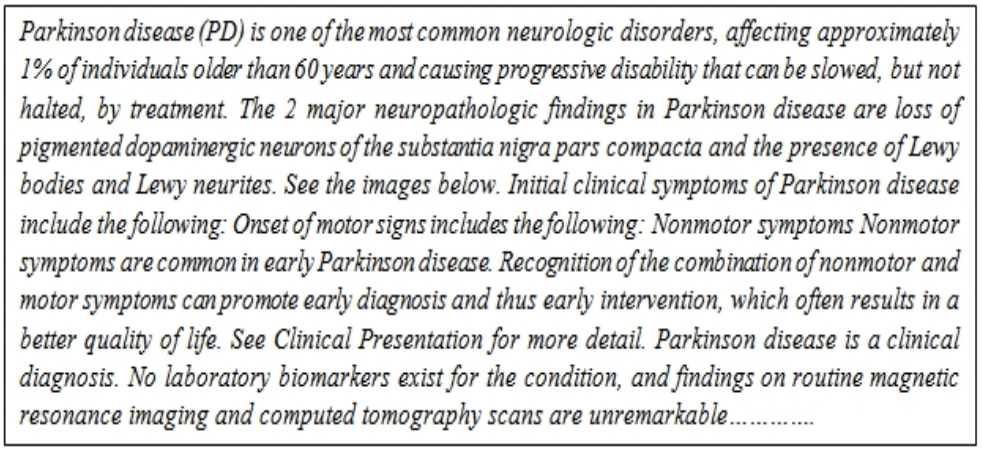

Parkinson disease (PD) is one of the most common neurologic disorders, affecting approximately 1% of individuals older than 60 years and causing progressive disability that can be slowed, but not halted, by treatment. The 2 major neuropathologic findings in Parkinson disease are loss of pigmented dopaminergic neurons of the substantia nigra pars compacta and the presence of Lewy bodies and Lewy neurites. See the images below. Initial clinical symptoms of Parkinson disease include the following: Onset of motor signs includes the following: Nonmotor symptoms Nonmotor symptoms are common in early Parkinson disease. Recognition of the combination of nonmotor and motor symptoms can promote early diagnosis and thus early intervention, which often results in a better quality of life. See Clinical Presentation for more detail. Parkinson disease is a clinical diagnosis. No laboratory biomarkers exist for the condition, and findings on routine magnetic resonance imaging and computed tomography scans are unremarkable……..

FIGURE 4.4: A snapshot of disease description collected from web

usefulness of Formal Concept Analysis (FCA) for representing concepts and relationships extracted from plain text corpus. The entire dataset, noun-phrases and the Python code is available in an open repository and can be freely accessed from the url https://github.com/anoop-research/relation-extraction.

4.2.1 Dataset Description

For the experiment, we have used disease description data which is publically available in unstructured form on Medscape website (http://www.medscape.com), which is an online global destination for physicians and other health care professionals. This website offers very latest medical news and expert opinions and disease details. We have crawled the website for diseases and treatment descriptions for two categories cardiology and neurology and collected data is stored in plain text files. For cardiology category, we have collected descriptions of 45 diseases such as Acute Coronary Syndrome, Alcoholic Cardiomyopathy, Heart Failure, Hypertension etc. and for neurology there are 42 descriptions for diseases such as Parkinson, Depression, Alzheimer, etc. A snapshot of such a description (for Parkinson disease) is shown in Figure 4.

4.2.2 Dataset Pre-processing

Dataset pre-processing concentrate on tidying up the data by removing unwanted characters, words, pecial symbols and links to external sources. We have not removed stop-words from the corpus for this experiment as some words in the stop-words list may

be found useful in identifying a particular relation word, for example, ”is-a”. Special symbols and other irrelevant characters are removed using regular expressions and used Snowball stemmer for reducing the words to its root word, say, ”affecting” to ”affect”. Then we have vectorized these words to feed into our proposed machine learning model.

4.2.3 Building Machine Learning Model

Our next step is to create a machine learning classifier that learns the Hearst patterns and other potential indications of relation patterns. We have used many classification algorithms to build the model, such as Decision Tree, Random Forest, Adaboost, and Support Vector Machines (SVM). The entire dataset has been splitted into 70 - 30, 70% is for training and 30% for testing the model. For training and testing, we have used a server machine configured with AMD Opteron 6376 with 2.3 GHz speed and 16 core processor and 16 GB of main memory. For implementing the classifier, Python 2.7 version is used along with ”sci-kit learn” [49] library.

For decision tree classifier, ”gini” criterion is used as a measure of quality of split and maximum depth parameter has been chosen as 15, by a trial and error method. Secondly, random forest classifier has been implemented with number of estimators as 300, maximum depth 15 and random state as 42. The values for these parameters are also set using trial and error method, starting with the value of 10 for the number of estimators, it is found that the optimum value for the current experimental setup is 300. The same method is used for the parameters maximum depth and random states. Our adaboost classifier used decision tree classifier as the base estimator that have maximum depth of 8 and random state of 42. The learning rate parameter, the number of estimators and random state were chosen to 0.9, 500 and 1332 respectively. For our support vector machine classifier, we have set number of random state as 22 and 100 as the maximum number of iterations.

4.2.4 Creating Concept Lattices

Once major concepts, phrases and relationships have been extracted, the knowledge may be represented in an intuitive and informative way so that meaning aware applications can be built on top of the knowledge. We use Formal Concept Analysis (FCA) for deriving concept hierarchy or formal ontology that represents concepts and associated

A	B	C	D	E	F
	blood pres...	breathing ...	cytotoxic eff...	fulminant ...	genetic trait
hypertension	X	X			
myocarditis			X		
nephrosclerosis				X	
pulmonic stenosis					
brown syndrome					X

FIGURE 4.5: Part of a formal context generated from the original dataset using our proposed method

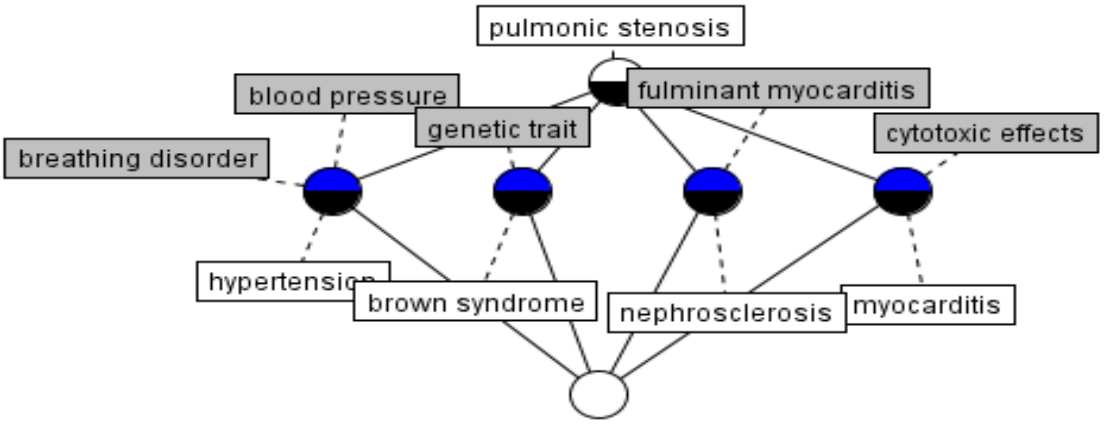


FIGURE 4.6: Part of a concept lattice generated from the original dataset using our proposed method

relationships we have leveraged. In order to build the lattice, a table with logical attributes represented as a triplet $\langle X, Y, R\rangle$ should be created first. In the triplet $\langle X, Y, R\rangle$ R denotes a binary relation between the objects X and Y. In our case the X denotes a set of disease names and Y denotes a set of attributes of the disease. For example, consider "hypertension" as the case, then we have "blood pressure", "breathing disorder", "cortisol stress reactivity" etc as its attributes. The binary relation R has a value 1 if "hypertension" have a particular attribute or else the value will be zero. A part of such a context table created is shown in Figure 4.5 and the corresponding concept lattice is shown in Figure 4.6.

4.2.5 Baselines Chosen

We have chosen two baselines for comparing our proposed method. Out of all classifier algorithms chosen (Decision Tree, Random Forest, Adaboost and Support Vector Machine), SVM showed better accuracy on relation classification and thus we compared the SVM model with the chosen baselines. We have chosen two baselines;

- **Baseline 1:**[50] This approach extracts facts from natural language texts with conceptual modeling [50]. This work shows the application of Formal Concept Analysis (FCA) for extracting facts from natural language text. Their approach combines concept graphs and concept lattices for leveraging facts which is closely associated with our proposed approach. They use concept lattices to model relationships that connect words and then these relationships for interpreting formal concepts as possible facts.
- **Baseline 2:**[51] The second baseline on the other hand attempts to create a public dataset containing more than 400 million hypernymy relations from CommonCrawl web corpus. Even though we are not considering all the relations used in their work, we have chosen this as our second baseline as our proposed work is aligned to their workflow in a greater extent.

The comparison of results of these baselines with our proposed framework and a detailed evaluation is discussed in Results and Evaluation Section.

4.2.6 Results and Evaluation

This section describes the results of our rigorous and systematic experiment on relation classification and knowledge representation using Formal Concept Analysis. As explained in Experimental Setup Section, for the relation classification, we have chosen four different algorithms Adaboost, Random Forest, Decision Tree and Support Vector Machine (SVM) and the Precision, Recall and F1 score reported by our machine learning classifier is shown in Table 4.2. Out of the four different algorithms chosen with optimal parameters, SVM is found to be showing better classification accuracy and thus that model has been chosen for comparing the performance of our proposed method with the chosen baselines. The classification accuracy comparison in terms of Precision,

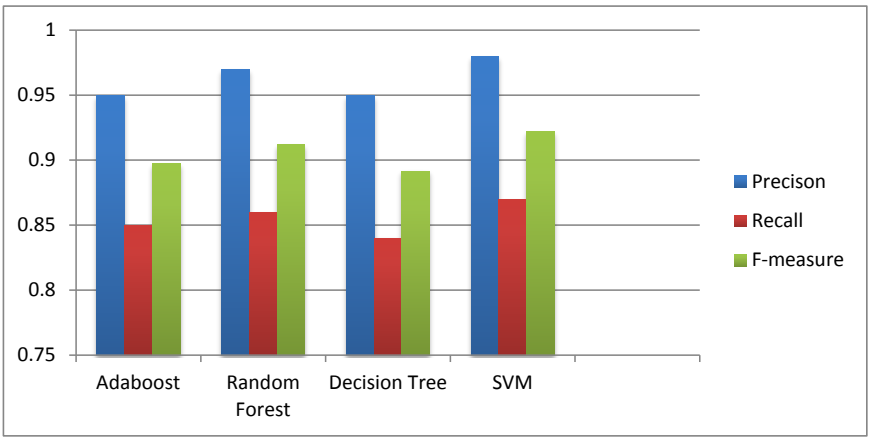
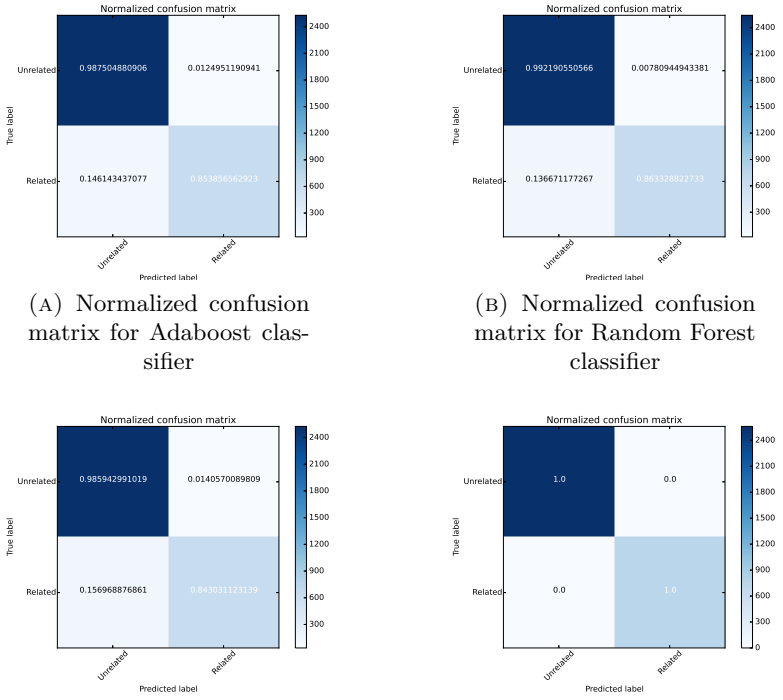


FIGURE 4.7: Graph representation of classifier performance on the chosen dataset

TABLE 4.1: Some of the concepts / medical phrases extracted for cardiology and neurology categories.

Cardiology	**Neurology**
Acute Aortic Dissection	central nervous system
Marfan syndrome	potential toxic metabolites
Type III dissections	pathological processes
beta adrenergic blocker	progressive disability
Thoracic aortic dissections	vascular parkinsonism
atherosclerotic disease	thalamocortical pathway
lymphocyte activation	painful muscular contractions
urine microalbumin	umbilical cord contamination
rheumatogenic strains	neonatal tetanus
streptococcal infections	elastic membrane

Recall and F1 score is shown in Figure 4.7 and the normalized confusion matrix for all the four classification algorithms are shown in Figure 4.8. Table 4.3 and 4.4 shows the performance of the proposed method in terms of precision, recall and accuracy when compared with the chosen baselines.



(A) Normalized confusion matrix for Adaboost classifier

(B) Normalized confusion matrix for Random Forest classifier

(C) Normalized confusion matrix for Decision Tree classifier

(D) Normalized confusion matrix for SVM classifier

FIGURE 4.8: Normalized confusion matrix for Adaboost, Random Forest, Decision Tree and SVM classifiers

TABLE 4.2: Precision, Recall and F1 score of different classifier algorithms on our dataset

Algorithm	**Precision**	**Recall**	**F1 Score**
Adaboost	0.95	0.85	0.90
Random Forest	0.97	0.86	0.91
Decision Tree	0.95	0.84	0.89
SVM	0.98	0.87	0.92

TABLE 4.3: Precision, Recall and F1 score comparison of baselines and our proposed method for relation extraction

Algorithm	**Precision**	**Recall**	**F1 Score**
Baseline 1[51]	0.79	0.81	0.79
Proposed	0.91	0.87	0.88

TABLE 4.4: Precision, Recall and F1 score comparison of baselines and our proposed method for concept lattice generation

Algorithm	**Precision**	**Recall**	**F1 Score**
Baseline 1[50]	0.80	0.78	0.78
Proposed	0.88	0.87	0.87

4.3 Conclusions

This chapter proposed a framework for extracting relationships that connect concepts / key-phrases found in unstructured text documents. It makes use of machine learning based approaches for identifying commonly occurring relations such as ”is-a” and other patterns discussed in Hearst Patterns, namely, ”such as”, ”or other”, ”and other”, ”including”, ”especially” etc. The proposed framework employs different machine learning algorithms viz; Adaboost, Random Forest, Decision Tree and Support Vector Machines to classify potential relations mentioned above. It uses Formal Concept Analysis (FCA) to represent concepts / key-phrases and relations that are leveraged using proposed relation extraction algorithm. Experiments on real-world dataset shows that the proposed method extracts better conceptual structures when compared with other state-of-the-art approaches. The next chapter introduces an interesting area in text mining and natural language processing called distributional semantics and its applications in conceptual knowledge discovery from unstructured text data. The algorithms proposed in the next chapter attempts to combine phrase embeddings and short text conceptualization for creating semantically rich and distinguishable phrase clusters.

CHAPTER 5

Conceptual Knowledge Discovery using Distributional Semantic Models

A vast majority of text mining and machine learning algorithms such as topic models, classification, clustering are based on statistical methods thus the semantics or meaning of the words or phrases are not considered. Interpretation of outputs generated by such algorithms are difficult for humans because of the absence of sufficient contextual information. Distributional semantics is a relatively new but active research area in natural language processing that quantifies semantic similarities between linguistic elements considering the context in which they occur. Conceptualization algorithms on the other hand enriches short text such as words and phrases. Analysis and interpretation of outputs generated by most text mining and machine learning algorithms are often difficult for humans. Major algorithms in machine learning such as topic modeling, clustering etc. are working on statistical and /or mathematical assumptions that do not take into consideration the meaning of words and phrases they operate upon. For instance, topic modeling algorithms such as LDA generates topics (word clusters) based on some probability distributions. Many text mining and machine learning algorithms do not understand the semantics of words and phrases but simply take them as a bag of words without taking into account the context in which those words occur. Such algorithms and systems are easy to train but the lack of understanding of enough context make it hard to do the analysis and interpretation of the outputs.

In natural language processing research, two algorithms that got significant attention are the ”word2vec” and ”phrase2vec” that attempts to represent words and phrases in vector spaces which are high dimensional. The advantage of these representations are that it uses semantic similarities among linguistic elements such as words and phrases

in a vector space. The first notable work that highlighted this idea rather than traditional syntactic and probability based bag-of-words method was the work proposed by Tomas Mikolov that represents words and phrases in vector spaces [56]. This approach introduced two algorithmic models called Continuous Bag of Words (CBOW) and Skip-gram. While traditional bag-of-words based approach operates on a collection of words, CBOW model make use of a continuous and distributed representation of the context in which these words appear. The second approach named Skip-gram model maximizes classification probability of one word based on another word in the same sentence [56]. Initially, the above said models could only work with word unigrams and phrases that were identified apriori, but later techniques for distributed representations of phrases [57], sentences , paragraphs and even documents [58] have been reported in the literature. Several deep learning applications in machine learning and natural language processing uses these words, phrases, sentence level embeddings as feature vectors for their processing. Very recently reported works on sentiment classification [174], vector embedding for Wikipedia entities and concepts [71] and sentence classification [72] shows the widespread applications of word, phrase or sentence embeddings.

On the other hand, several recent attempts were reported that tries to link or associate words and phrases to human curated ontologies such as Dbpedia [73][74] and Wordnet [75]. These works bring in semantically similar concepts and relationships to contextualize words and phrases which is known as short text conceptualization. This is an interesting research area in text mining and has been explored by research communities for contextualizing short texts that may help automatically enrich ontologies, which are backbone of meaning aware computing. Earlier but notable approach in short text conceptualization was reported towards the creation of a taxonomy called Probase for better text understanding [60]. Later several prominent methods and algorithms [59][76][77][78] were developed for conceptualizing short texts extracted from unstructured text data. This work proposes a framework that attempts to combine phrase embeddings and short text conceptualization for creating semantically rich and distinguishable phrase clusters. The framework uses a standard k - means and a distributed k - means clustering algorithms for the generation of conceptualized clusters. The rest of this chapter is organized as follows: Section 5.1 discusses the proposed conceptualized phrase clustering algorithms and in Section 5.2, the experimental setup is described. Section 5.3 outlines the results and evaluation of the proposed algorithms and Section

5.4 concludes the chapter.

5.1 Conceptualized Phrase Clustering

This section details our proposed framework that integrate distributed representation of phrase vectors (phrase2vec) and short text conceptualization for generating semantically distinguishable and close to real-world clusters. We propose two algorithms here - (a) an algorithm that generates conceptualized phrase embedding clusters using standard k - means algorithm and (b) an algorithm that uses MapReduce framework (distributed k - means) for generating conceptualized phrase embedding clusters. Such easily interpretable clusters may find immediate applications in document classification, phrase based document retrieval etc. Workflow of our new framework is shown in Figure 5.1. Proposed novel algorithms are shown in Algorithm 3 (standard k - means) and Algorithm 4 (distributed k - means) respectively. The approach takes a large corpus of unstructured text documents as input and generates conceptualized phrase clusters as output. Our distributed k - means based approach have two methods: $k - meansMap$ and $k - meansReduce$ given in Equation 5.1and Equation 5.2 respectively. Our "Map" function works on every data point x in the dataset under consideration. For any given data point x, we calculate the squared distance between the data point x and each mean, and then computes the mean μ_i in such a way that this mean will minimize this distance. This outputs a key-value pair with the computed mean's index as the "key" (in this case, i) and $(x, 1)$ as the value.

$$\textbf{k-meansMap}(x) : emit(argmin_i ||x - \mu_i||^2, (x, 1)) \tag{5.1}$$

Our "Reduce" function is a summation which is pairwise that operate over the values for each key, i, and for any two given value pairs for a specific key, we first add corresponding elements and then combine them. So our function is:

$$\textbf{k-meansReduce}(index, [(x, p), (y, q)]) : return(index, (x + y; p + q)) \tag{5.2}$$

MapReduce function will accept input in $< key, value >$ pair format and in our case, we have identified key as center of the cluster and $value$ as the vector representation of phrases given by phrase2vec implementation. The Mapper function calculate the

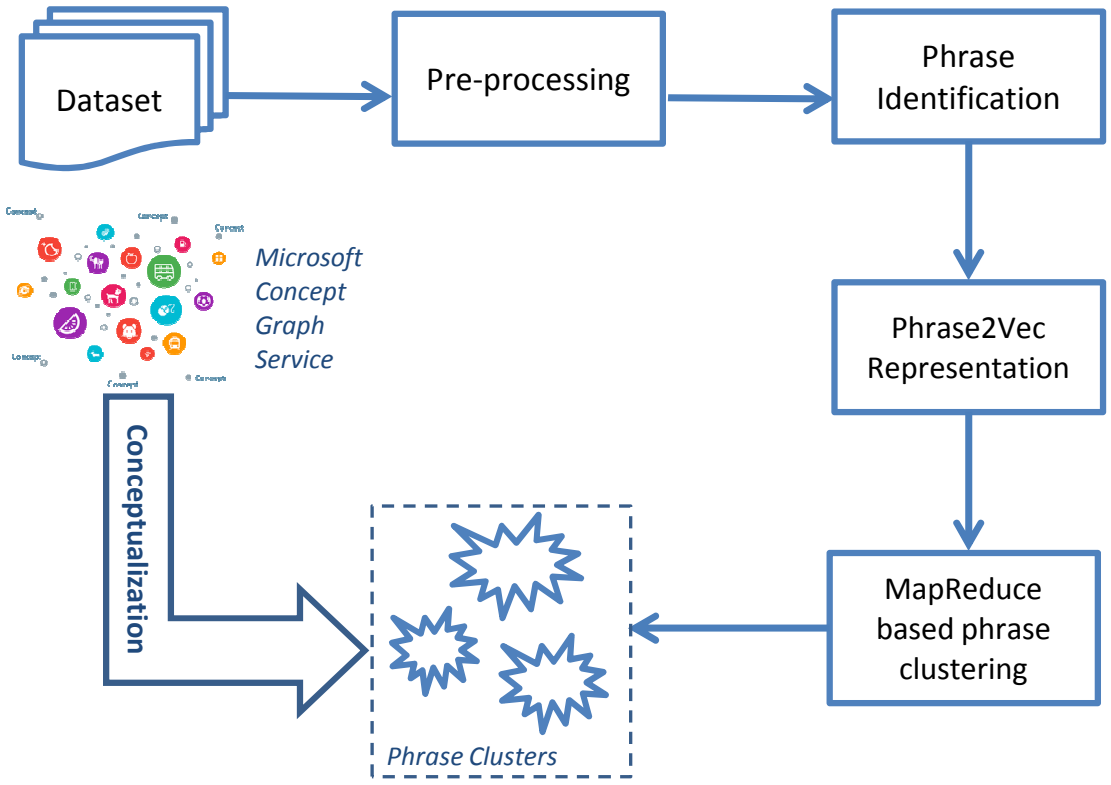


FIGURE 5.1: Proposed method

distance among vectors and each cluster centers and also find the cluster from which any specified vector is nearer. On completing the calculation of distances in loops, the vector will get assign to the closest cluster. A complete description of the steps involved in our proposed approach are discussed in the subsequent sections . The input to

Algorithm 3 : Semantic Phrase Clustering and Conceptualization using Standard k-means

```
1: function ClusterAndConceptualize(D)
2:     for each text document d from D do
3:         Pre-process d to remove numbers, special characters, etc.
4:         Run POS tagger and collect tags
5:         Take combinations of POS tags and tag them in the corpus
6:     end for
7:     Extract phrase vectors from D using Phrase2Vec
8:     Perform k means on Phrase2Vec generated vectors to output phrase clusters
9:     Find similarity of phrase vectors using cosine similarity
10:    for each phrase from clusters C1, C2,...,Ck do
11:        Query Microsoft Concept Graph to conceptualize the phrase clusters
12:        Write json output for visualization
13:    end for
14: end function
```

Algorithm 1 is D, a dataset containing a set of documents d. For each document d, the

algorithm removes special characters, symbols, etc. using regular expressions and run a parts-of-speech tagger to collect tags. In the next step, the algorithm take combination of the parts-of-speech tags such as *noun*, *adjective*, etc. and tag them in the original document d. Then the phrase2vec algorithm will be executed that will create distributed phrase vectors of all the documents d available in the corpus D. The algorithm then performs a k- means clustering where the value of k will be decided based on trial and error method. The aim of this step is to create semantic phrase clusters where the similarity is calculated using cosine values. As the final step, for every cluster C_i, the algorithm query Microsoft concept graph service to conceptualize and enrich the phrase clusters. The output will be written to a json file for visualizing the clusters. The

Algorithm 4 : Meaning aware Phrase Clustering and Conceptualization, using Distributed k-means (MapReduce)

function ClusterMapReduce(D)
  for each text document d from D **do**
    Pre-process d to remove numbers, special characters, etc.
    Run POS tagger and collect tags
    Take combinations of POS tags and tag them in the corpus
  end for
  Train tagged corpus D using Phrase2Vec and obtain phrase vectors.
  k-meansMap(x) :
  $emit(argmin_i||x - \mu_i||^2, (x, 1))$
  k-meansReducer($i, [(x, s), (y, t)]$) :
  $return(i, (x + y; s + t))$
  for each phrase from clusters C_1, C_2,...,C_k **do**
    Query Microsoft Concept Graph to conceptualize the phrase clusters
    Write json output for visualization
  end for
end function

Algorithm 4 describes the phrase clustering and conceptualization using a distributed k - means method. For this algorithm, the steps until the phrase2vec generation from the corpus D remains the same as Algorithm 3. The algorithm defines two sub-routines $k - meansMap$ and $k - meansReducer$ which are used for distributed execution of the phrase clustering. After the clustering process, the algorithm uses Microsoft concept graph service to conceptualize and enrich the phrase clusters and the output is written to a json file for the visualization.

5.1.1 Dataset Pre-processing

As most of the unstructured text datasets contain noises and irrelevant or junk characters, we need to do a thorough pre-processing before proceeding to the actual computation. In this step, we have removed all the stop words such as "a", "and" etc. and the digits, special characters etc. are also removed. This has been done for each and every documents in the corpus and cleansed data is written onto new files and made an experiment ready copy of the corpus.

5.1.2 Phrase Identification and Tagging

This step identifies phrases from documents using natural language processing rules and tags them. POS (Parts-of-Speech) tagging was done on the experimental ready copy of the dataset and identified all adjectives, nouns and verb tags. On analyzing real-world keyphrases, we observed that keyphrases are combinations of nouns and adjectives. For instance, the phrase "super computer" is made up of words "super" and "computer" where the former (super) is tagged as adjective and the latter is tagged as noun. In this work we have identified all such rules which are shown in Table 5.1. We then rewrite the phrases and append an underscore between words in the identified phrase. In the above mentioned example, we merge the words "super" and "computer" as "super_computer" and tagged as a phrase for our further experiment.

5.1.3 Generating Phrase Embedding and Clustering Phrases

In this phase, we train phrase2vec model using our experimental ready copy of dataset that represent phrases in a high dimensional vector space. We make use of the method discussed in Tomas Milokov's paper [57] to create distributed phrase vectors and the context is also represented with phrases and the similarity can be computed as the cosine angle between any two phrase vectors. Once phrase2vec embeddings are generated, we cluster them using k-means so that phrases which are related will come in one cluster. For the k-means, we started with the value of k as 50 and increased upto 100 on subsequent iterations of the experiment. The value for k is chosen as 50 using trial and error method after running the experiment for several such values. For the lower values of k, the quality of the cluster was not good and for the value 50, the cluster was distinguishable. This

TABLE 5.1: Common NLP rules for identifying noun-phrases from corpus

No.	NLP Rule
1	JJ + NN [Adjective + Noun]
2	JJ + NNS [Adjective + Noun]
3	JJ + VBG [Adjective + Verb]
4	JJR + VBG [Adjective + Verb]
5	NN + NN [Noun + Noun]
6	NN + NNS [Noun + Noun]
7	NN + VB [Noun + Verb]
8	NN + VBG [Noun + Verb]
9	NN + VBN [Noun + Verb]
10	NNP + NNS [Proper noun + Noun]
11	NNS + VB [Noun + Verb]
12	NNP + NNS [Proper noun + Noun]

experiment got reasonably good clusters for the values from 50 to 100 but for the values greater than 100 and less than 50, it is observed that the clusters become overlapping.

5.1.4 Conceptualization of Phrases

In this phase, we conceptualize identified phrases that offer better interpretation and our proposed method make use of cosine vector similarity of the phrase2vec model along with the conceptualized information. We use Microsoft's concept tagging and conceptualization model [59][60] that uses a knowledge base called Probase [60]. Given each identified phrase, we use Probase [60] to query the associated and related concepts and phrases. For example, consider the phrase "computer network", the concept tagging service may give associated concepts / phrases such as "large network setting", "signal transmission network", "communication system" etc. as the related concepts and phrases.

5.2 Experimental Setup

This section detail the experimental setup used for implementing our proposed algorithms on real-world datasets. A brief description of datasets used in this experiment and the experimental testbed descriptions are due.

5.2.1 Dataset Description

For this experiment, we used three unstructured text datasets which are publicly available – the Reuters 21578, BBC and StackExchange datasets. The details of datasets are given below.

- **BBC News Dataset:** This dataset has total number of 2225 unstructured text documents taken from BBC website that is related to various domains such as technology, entertainment etc. The dataset can be freely downloaded using the url: http://mlg.ucd.ie/datasets/bbc.html.

- **StackExchange Dataset:** This dataset contains user contributed but anonymized contents on the StackExchange platform. For this experiment, we have considered four categories such as academia, android, gaming and travel. We combined all posts under a single category and dumped in one document and thus our final dataset have 7884 text documents in total. This dataset can be downloaded from https://archive.org/details/stackexchange.

- **Reuters Dataset:** This dataset contains a collection of news articles widely used among data mining researchers. The dataset can be downloaded using the url: http://www.daviddlewis.com/resources/testcollections/reuters21578

5.2.2 Experimental Testbed

The proposed algorithms were implemented on a server computer that has a configuration of AMD Opteron 6376 @ 2.3 GHz and also has 16 core processor and 16 GB of main memory. The Parts-of-Speech (POS) tagging was done on preprocessed text data and potential phrases were found out using natural language processing rules shown in Table 1. Then the identified phrases are tagged in the corpus and trained phrase2vec implementation publicly available at the repository, https://github.com/zseymour/phrase2vec. This library is an implementation of phrase2vec from modified word2vec code available at Google repository. We used skip-gram based architecture of phrase2vec for this experiment in which the input layer of the model accepts $p(t)$ as the phrase and

generates $p(t-2)$, $p(t-1)$, $p(t+1)$, $p(t+2)$ etc. as the output which are semantically similar to the input phrase $p(t)$.

Initially we cluster phrase vectors obtained from phrase2vec using standard k-means[81] algorithm implemented using scikit.learn machine learning library[68]. We initially start with 50 as the value for k, and subsequently increased up to 100, with varying sizes of samples ranging from 5000 to 40000. Then used the square root method where we took square root of the number of data points divided by two and finally used 'elbow' method as well. The idea behind 'elbow' method is to take the value of k at which the sum of squared error shows a decrease. Later, we implemented the framework using distributed k - means by employing MapReduce framework to increase the scalability of the approach as we found that the performance of standard k - means algorithm was degraded when working with large datasets used in this experiment. Once clusters were formed, next step is to conceptualize phrases obtained from phrase2vec clusters so that enough contextual information can be brought into the clusters for interpretation. We make use of Microsoft's Concept Tagging Model (MCTM) [59][60] for conceptualizing short texts. MCTM attempts to associate text entities into semantic concepts with probabilities. Given a linguistic element such as word or phrase, this conceptualization model add meaning to it by associating them into concepts for short conceptualizing phrases. We use Microsoft concept graph API accessed using the url: https://concept.research.microsoft.com/Home/API. The json responses returned by the service is parsed and stored for visualizing conceptualized clusters.

5.3 Results and Evaluation

Here, the authors present the experimental results and a detailed evaluation. Table 2 shows a partial list of phrases identified from BBC News Dataset (for technology domain) using handcrafted parts-of-speech tagging based rules. For performance comparison, we have chosen two popular and state-of-the-art clustering algorithms, Self Organizing Map (SOM)[79] and Expectation Maximization[80], as baselines. Our proposed algorithms uses standard k - means and distributed k - means for clustering semantic phrase vectors obtained from phrase2vec model. The performance of the chosen baselines along with our proposed algorithms are shown in Table 5.3. From the results, it is evident that our proposed algorithms shows better clustering accuracy and performance when compared

TABLE 5.2: A partial list of phrases identified from BBC corpus (technology domain) using NLP rules given in Table 1

Sl. no.	Phrase	Sl. no	Phrase
1	internet_connection	7	dialup_internet
2	internet_networks	8	dialup_phone
3	broadband_internet	9	dialup_connection
4	broadband_networks	10	web_search
5	network_connection	11	netflix_company
6	phone_connection	12	lowcost_mobiles

TABLE 5.3: Clustering performance of proposed algorithms (using standard k - means and distributed k - means) with chosen baselines on three different datasets - BBC News, StackExchange and Reuters 21578

	Cluster Performance				
Dataset / Algorithm	No. of Clusters	EM[80]	SOM[79]	Standard k-means[81]	Distributed k-means
BBC News Dataset	8	62	61	**74**	**89**
	16	67	56	**78**	**93**
	32	76	49	**84**	**92**
StackExchange Dataset	8	69	71	**81**	**91**
	16	77	58	**79**	**90**
	32	79	51	**88**	**95**
Reuters 21578	8	64	59	**79**	**89**
	16	74	55	**87**	**94**
	32	71	48	**88**	**96**

with SOM[79] and EM[80]. We have chosen different cluster sizes for this experiment (8,16,32) and observed that as the number of clusters increases, the performance of SOM is decreasing but EM algorithm shows better cluster accuracies compared to SOM. In all the datasets, our proposed algorithm with standard k - means and distributed k - means implementations shows significantly better cluster accuracy. Visualization of phrase clusters obtained using our proposed algorithms with sample size 5000, 10000, 20000 and 40000 are shown in Fig. 5.2(a), Fig. 5.2(b), Fig. 5.3(a) and Fig. 5.3(b) respectively. Results of conceptualization using Microsoft Concept Graph is shown in Table 5.4 for the phrases given in Table 5.2 that are obtained from the clustering results.

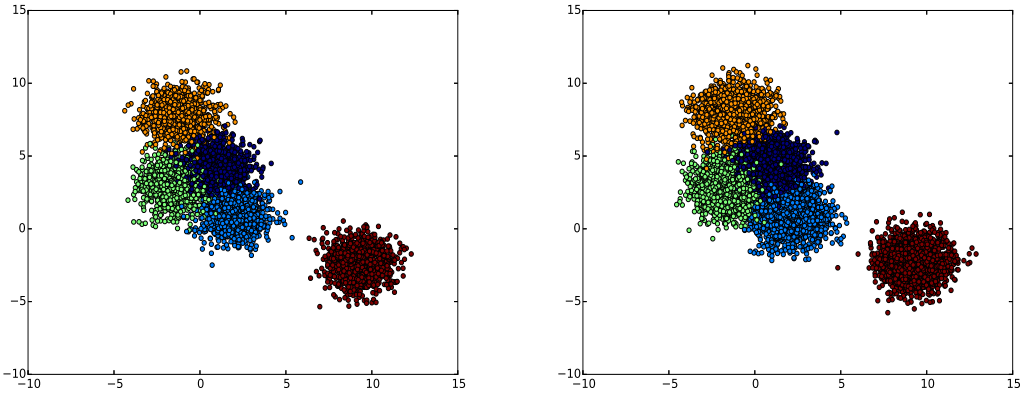


FIGURE 5.2: Cluster visualization of phrase vectors obtained from phrase2vec with varying sample sizes (5000 and 10000) with centers = 5, number of features = 2, and random state = 0

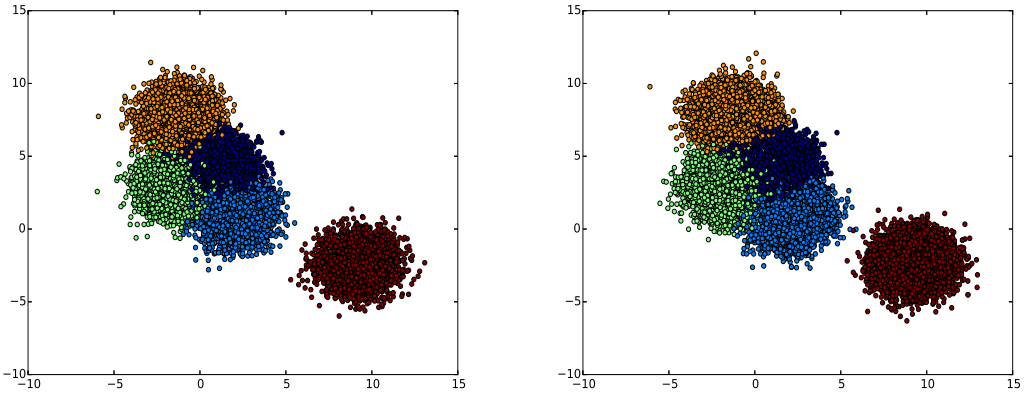


FIGURE 5.3: Cluster visualization of phrase vectors obtained from phrase2vec with varying sample sizes (20000 and 40000) with centers = 5, number of features = 2, and random state = 0

5.4 Conclusions

This work proposed a hybrid algorithmic approach for distributed representations of phrases in high dimensional vector space and conceptualization of short texts. This algorithm generates semantically valid and close to real-world phrase clusters consists of conceptualized phrases using a distributed k-means clustering in MapReduce framework. The advantage of the proposed method is that it can create highly distinguishable phrase clusters with enough contextual information for each phrase. Experimental evaluations with large unstructured text datasets show that the proposed approach gives better generalization performance than the state-of-the-art methods. The next chapter

TABLE 5.4: Result of conceptualization using Microsoft Concept Graph API service (for phrases given in Table 2)

Sl. no.	Phrase	Conceptualized Result
1	internet_connection	modern communication facility, configure computer equipment, packet switching application
2	internet_networks	static network, communication network, datum network
3	broadband_internet	telecommunication service, high bandwidth datum communication, modern datum service
4	broadband_networks	information technology, digital medium platform, second communication network
5	dialup_connection	internet connection service, low bandwidth connection, internet service

discusses some of the real-world applications of the algorithms proposed in the previous chapters. Some interesting applications discussed in this paper include topic modeling guided phrase query based document ranking, aspect oriented sentiment analysis, and information retrieval from online social networks.

CHAPTER 6

A Few Real-world Applications of the Proposed Algorithms

Text mining and natural language processing attained significant attention in the recent past due to an increase in number of intelligent applications being introduced in the industry. Due to the ever increasing relevance of world wide web and other text producing and consuming platforms, there are more demands for building systems that can read, analyze and unearth actionable insights from humongous amount of unstructured data. Text mining and natural language processing join hands together for extracting meaningful representations of text which may bring improvements to potential real-world applications. There are many applications where text mining with natural language processing algorithms are running in the background for extracting useful patterns from the unstructured text input. Some of these applications include semantic search engines[184][185][186][187], document indexing and ranking [188][189][190][191][192], sentiment analysis[193][194][195][196], question answering systems[197][198][199][200] to name a few. This chapter discusses a few potential real-world applications of the proposed algorithms such as phrase query based document ranking, aspect oriented sentiment analysis and a distributional semantics based information retrieval for online social networks. The rest of this chapter is organized as follows: Section 6.1 introduces algorithms for phrase query based document ranking which discusses methods for topic modeling guided concept extraction and document ranking. In Section 6.2, algorithms for aspect oriented sentiment analysis are discussed with real-world implementation which is under pilot running in different stages. Section 6.3 discusses a distributional semantics based information retrieval framework for social network data.

6.1 Phrase Query based Document Ranking

Topic modeling algorithms such as Latent Dirichlet Allocation (LDA) [82], Probabilistic Latent Semantic Indexing (PLSI) [83] and Probabilistic Latent Semantic Analysis (PLSA) [84] are proved to be extensively useful in bringing out hidden themes from textual content which are used for further analysis. Given a large collection of text documents, these algorithms generate such themes by representing each document as a random mixture over latent topics and assumes each topic be a probabilistic distribution over words. This idea is used in various scenarios such as document analysis and pattern identification in text mining, because text is considered to be the best data on which topic modeling can be applied. Majority of the topic modeling algorithms work on the assumption called ”bag-of-words” to generate ”topics” which are sequences of word unigrams. The interpretation of such topics still remain as a hindrance to most of the text mining practitioners [85]. In real life scenarios, humans understand the concepts and key-phrases as a combination of words such as bi-grams and tri-grams. Splitting these bigrams and trigrams into unigrams cause its semantic meaning to be lost and that would generate ambiguous and irrelevant topics. For example, the concept ”network security attack” is more interpretable for humans compared to the word unigram - ”attacks”. Chopping them into ”network”, ”security” and ”attack” unigrams cause its semantic meaning to be lost.

Concept mining and extraction plays a key role in text mining where each concept is a combination of words which are real or imaginary. Those concepts are used for tasks such as document retrieval [86, 87], classification [88], concept based sentiment analysis [89], semantic product search in e-commerce [90, 91] and concept hierarchy learning [102]. Search engines and document databases also use such concepts to locate related information from huge text archives. Thus concept and key-phrase identification is a key process in unearthing potential patterns out of the data. Understanding, analyzing and summarizing key concepts from large volumes of text data is non-trivial and crucial in knowledge discovery process.

We address the problem of interpreting statistically generated topics by inducing concepts and key-phrases from large document collection using probabilistic topic modeling. Different from previous studies, we attempt to extract semantically valid concepts and key-phrases directly from LDA [82] generated ”topics” using a simple framework

which is completely unsupervised and easy to implement. Exploration in these dimensions may enrich existing systems in better understanding of text data and can be applied to tasks such as document summarization [95], automated ontology generation [96] etc. This work also show that the proposed framework is fit for the task of searching and retrieving documents in large archives and outperforms state of the art probabilistic and phrase based mechanisms.
The main contributions of this research are summarized as follows:

1. Proposes a novel framework for extracting semantically rich and close to real-world concepts and key-phrases directly from statistically generated topics using a lightweight scoring algorithm.

2. Rigorous experimental comparison of the proposed concept extraction method with state-of-the-art approaches, establishes the effectiveness of proposed algorithms.

3. Demonstrates the usefulness of proposed method and algorithms in improving document retrieval performance task.

6.1.1 Problem definition

Here the problem is formally defined. Concept / key-phrase extraction problem is the task of identifying potential and relevant concepts / key-phrases from documents from a text corpus. Given a document $d_i, i = 1, 2, ..n$, from the corpus D, the concept extraction problem identifies semantically valid concepts from d_i such that we get a collection of concepts against each document d such that $C = \{c_1, c_2, ...c_n\}$. Given a static corpus D containing n documents such that $D = \{d_1, d_2, ..., d_n\}$, our problem is to extract close to real-world and semantically valid concepts from D using a topic modeling process. We also address the problem of phrase based document retrieval such that given a phrase, top- k relevant documents can be retrieved that contain phrases semantically similar to the query phrase.

6.1.2 Topic modeling guided concept extraction

This section presents the approach for extracting close to real-world and semantically valid concepts and key-phrases from relatively larger static document corpus. Firstly

the proposed concept scoring and topic modeling guided concept extraction algorithm is discussed. Then shows how the proposed method can better rank documents in a large text corpora and how the method is efficient in retrieving relevant documents. Firstly, we outline the proposed scoring function that is used to score real-world concepts that are latent in large text document corpora. A new scoring function is introduced where the score of a noun-phrase n in a topic t is computed as the probability of a topic over the document multiplied by the frequency of noun-phrase in that document. Notationally,

$$score(n,t) = \sum_{document_d} p(t|d) freq(n,d) \tag{6.1}$$

where $p(t|d)$ is the scoring for the document-topic pair obtained from the LDA and $freq(n,d)$ is the frequency of noun-phrase n in document d. After calculating score for each noun-phrase, top k noun-phrases are filtered according to the score. Now we propose Algorithm 5, which takes a set of topics which are probability distribution over words, generated by Latent Dirichlet Allocation (LDA) [82] and a set of static corpus of documents D. Using an off-the-shelf noun-phrase tagger, the algorithm first tags all the noun-phrases from static corpus D. The frequencies of noun-phrase in documents are then computed. The probability distribution of topic over documents, $p(t|d)$ is computed from LDA output. Then the score of already tagged noun-phrases in documents, $score(n,t)$ calculated as the product of $p(t|d)$ and $freq(n,d)$. Finally, top - k noun-phrases are filtered according to the score, $score(n,t)$ calculated using Equation 6.1.

Algorithm 5 : Algorithm for topic modeling guided concept extraction

1: **function** TM-ExtractConcepts(D,T) ▷ D: Static document corpus, T: Topic collection
2: **for** each document d from D **do**
3: run through a noun-phrase tagger, and collect noun-phrases
4: **end for**
5: **for** topic t and noun-phrase n **do**
6: score(n,t) = $\sum_{document_d} p(t|d) freq(n,d)$
7: **end for**
8: For each topic t, find the top-k noun-phrases according to the $score(n,t)$
9: **end function**

6.1.3 Topic modeling guided document retrieval

Given a multi-word query Q, computing the top - k relevant documents from a large collection of documents D is a fundamental problem in Information Retrieval (IR). The relevance between a given query and a document d is determined by using similarity functions such as BM25 [107] and generate scores for every document d in the collection D. Then the system reports k top scored documents as the ranked result. Earlier, vector based models were introduced in which documents are represented as a vector of terms, $\vec{d_j} = w_{1j}, w_{2j}, ...w_{tj}$ where t is the total number of words in the document and each w_{1j} >0 if and only if the word i is present in document d_j. In this model, the term weights are not binary as in boolean model since vector space model do not consider the presence or absence of terms. In the same fashion, query is also represented as vector and the similarity between a query vector and a document vector is calculated for measuring the relevance and the same is used as a ranking score. In vector space model, there are two widely used factors for calculating term weights; namely, term frequency (tf) and inverse document frequency (idf). The term weight is computed by multiplying these two factors and widely known as $tf - idf$ measure [106]. Then the similarity measure between the query vector and document vector is calculated using widely used similarity measures such as Okapi BM25 [107] as:

$$score(q, d) = \sum_{i=1}^{|q|} idf(q_i).\frac{tf(q_i, d).(k_1 + 1)}{tf(q_i, d) + k_1.(1 - b + b.\frac{|d|}{avgdl})} \tag{6.2}$$

where, $tf(q_i, d)$ is q_i's term frequency in the document d, $|d|$ is the length of the document d in words, and $avgdl$ is the average document length in the text collection from which documents are drawn. When dealing with large collection of documents, calculating tf, itf and then computing the similarity measures between query and document vectors are often difficult.

Later, a technique called inverted index was introduced which is the most widely used data structure in the area of information retrieval. For a document corpus, D, it maintains an index comprising of *terms* and an *inverted list*, where *term* refers to each word in the document collection and *inverted list* maintain all documents that contain corresponding word. Along with the document id, it also maintains a score that denotes how relevant the word is for a particular document. Major drawback of such a

structure is that it works only for a set of predefined words or terms and only be suited for documents having well defined word-boundaries. Recently different approaches were introduced for supporting phrase searching in inverted index and some naive approaches were reported in which an inverted list for all possible phrases are maintained [206]. The major drawback of such approach is with the size of index that restricts its use for practical applications. Another approach called "next-word-index" that uses a list of terms that will immediately occur with a specific term t was introduced in [207] [208] [209]. This approach could address phrase searching with exactly two words in the query. But when it goes beyond two words, some intersection algorithm should be used for concatenating the words to form phrase.

Here, we propose a new method for retrieving top - k documents from a large collection of text corpus using [concept - topic - document] triplet where concepts are extracted using the scoring function we have introduced in Section 6.1.2. Given a query, we first find relevant topics using the function score(n,t) = $\sum_{document_d} p(t|d)freq(n,d)$ where $p(t|d)$ is the probability of topic over document and $freq(n,d)$ is the frequency of phrase over document, which are precomputed. Using topic distribution over documents information given by LDA, a set of documents are obtained having highest topic probability and present top-k documents as the ranked collection. These procedures are given in Algorithm 6 where the algorithm takes a phrase query q and calculate the score of q in topic t according to the steps discussed.

Algorithm 6 : Algorithm for topic modeling guided document retrieval

1: **function** TM-RetrieveDocuments(q) ▷ q: A phrase query
2: retrieve the significant topics for q using scoring function, score(q,t)
3: where score(q,t) = $\sum_{document_d} p(t|d)freq(q,d)$
4: use score(q,t), to get topic - document distribution (given by LDA)
5: list top - k documents according to the distribution
6: **end function**

6.1.4 Experimental setup

In this section, the experimental setup and detailed evaluation of the results are discussed. Two separate experiments are conducted, one for leveraging semantically rich concepts and key-phrases from statistically generated topics as proposed in Section 6.1.2.

The closeness of identified concepts with real world are evaluated by comparing the results with a crowd-sourcing experiment s conducted. The second objective is to showcase the potentiality of proposed approach in real life applications such as document search and retrieval task. The dataset used in this experiment is described in Section 6.1.5 including how we made an experiment ready copy of such data.

6.1.5 Dataset description

We use three different datasets in our experimental evaluation - the BBC dataset, StackExchange dataset and a much larger Reuters 21578 dataset. A short description of these datasets are given below.

- **BBC News Dataset:**[1]. The general BBC dataset consist of 2225 text documents directly from their website corresponding to stories in five areas such as business, entertainment, politics, sports and technology, from 2004 to 2005.
- **StackExchange Dataset:**[2]. This dataset comprise of anonymized, user-contributed contents on the StackExchange website in different categories. Since our proposed approach uses a crowd-sourcing experiment we have carefully chosen categories in such a way that it matches with the general knowledge of the users participating. Specifically we have chosen academia, android, cooking, gaming and travel categories for this experiment. All post under a single thread is merged in one document and thus the final dataset consists of 2769 documents.
- **Reuters 21578 Dataset**[3]. Reuters 21578 dataset is a collection of newswire articles and is popular among data mining communities. The collection was made available for research purposes by Reuters in 1990.

6.1.6 Experimental testbed for topic induced concept extraction

This section describes the experimental setup used for proposed concept extraction experiment. All methods described in this work were implemented in Python 2.7 and the experiments were run on a server configured with AMD Opteron 6376 @ 2.3GHz / 16 core processor and 16 GB of main memory. Firstly, pre-processing was done for

[1]http://mlg.ucd.ie/datasets/bbc.html
[2]https://archive.org/details/stackexchange
[3]http://www.daviddlewis.com/resources/testcollections/reuters21578/

each document to remove stop-words such as ”the”, ”and” etc. Then TextBlob [110] library in Python is used for tagging noun-phrases from each document. This experiment considered noun-phrases that contain at least two words and for all datasets, we have filtered noun-phrases that are word n-grams where n $\geq$ 2. Topics were modeled using LDA algorithm and the score of each noun-phrase is calculated using Equation 6.1 and selected top -k noun-phrases for each topic. We use MALLET[4] implementation of the LDA model to generate topics and the number of iterations in Gibbs sampling used in this work is set as 300, when using trial and error method, we found that Gibbs sampling usually approaches the target distribution after 300 rounds of iterations. Alternatively, the iterations can stop when the distance of the current distribution from the target distribution is less than a specified threshold. The distance can be the relative entropy of the KL divergence that is used in information theory or any other distance measure. The parameters α and β are set as $\alpha = 50/Z$ and $\beta = 0.01$, respectively. The α controls the mixture of topics for any given document and the β hyperparameter controls the distribution of words per topic. If we decrease the value for α then the documents will likely have less of a mixture of topics and if we increase the value of α then the documents will likely have more of a mixture of topics. Likewise, if we decrease the value of β then the topics will likely have less words and if we increase the value of β then the topics will likely have more words.

Baselines - We compare our proposed concept extraction approach with the following baselines. For all the hyper-parameters, we use the default settings and optimizes these parameters every 100 Gibbs sampling iterations.

- **TNG**[92] - The topical n-gram model automatically decides whether to form an n-gram or not by considering its surrounding text. The TNG model provides a systematic way to model topical phrases and simultaneously detect n-grams and topics. The MALLET implementation of the TNG is used with all the default settings for hyperparameters.
- **TopMine**[99] - This algorithm first extracts phrases using a method similar to frequent pattern mining and then train a modified LDA model on the ”bag-of phrases” input.

[4]http://mallet.cs.umass.edu/

6.1.7 Experimental testbed for document retrieval

In this section, the experimental setup used for evaluating document retrieval using our proposed framework is discussed. We compare our proposed model to BM25 [107] which is a term-matching based baseline and two phrase based retrieval baselines that uses a flat position index [111] and inverted index [112] as their data structure. For the BM25 setup, we have used the parameters $k1 = 1$, $k2 = 0$, $k3 = 1$ and $b = 0.5$ for initial experiment and later finalized the values of $k1$ and b as 1.7 and 0.95 respectively as these values gave optimum retrieval results for the current experiment. This is done using a trail and error method. We implemented the inverted index and flat position index using Python 2.7 version and used public code libraries at https://www.rosettacode.org/wiki/Inverted_index_Python and https://github.com/matteobertozzi/b code/blob/master/py-inverted-index/invindex.py for developing some of the components of the indexes. All the experiments were run on a server configured with AMD Opteron 6376 with 2.3GHz with 16 core processor and 16 GB of main memory.

6.1.8 Result and evaluation

This section presents result and detailed evaluation of proposed concept extraction and document retrieval experiments. Table 6.1 shows the performance comparison of the proposed method for topic modeling guided concept extraction with the baseline methods chosen. On all the three datasets used such as BBC, Reuters and StackExchange, the proposed topic modeling guided concept extraction algorithm significantly outperformed the baselines when compared with precision, recall and f-measure.

6.1.9 Qualitative evaluation

For measuring closeness of the concepts generated using our proposed method with the real-world concepts, we have conducted a crowd-sourcing experiment which is widely used in text mining and processing tasks to validate and verify the computer generated outputs with human annotated contents. Many literatures are available that used crowd-sourcing to tasks such automated question-answering systems [108] and ontology alignment [109] to name a few. For this experiment, a simple web interface has been created that present the users with the concepts extracted by our algorithm and asked

TABLE 6.1: Comparison of proposed topic modeling guided concept extraction method with Topical N-gram, TopMine baselines. Proposed method outperforms significantly better than all the chosen baselines

Method	**Dataset**								
	Reuters			**BBC**			**StackExchange**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
TNG[92]	0.7015	0.6922	0.6968	0.7259	0.7199	0.7228	0.6325	0.7011	0.6650
TopMine[99]	0.7692	0.6988	0.7323	0.7903	0.7254	0.7564	0.7521	0.7741	0.7629
Proposed	**0.9012**	**0.8966**	**0.8988**	**0.9318**	**0.9458**	**0.9387**	**0.9187**	**0.9258**	**0.9222**

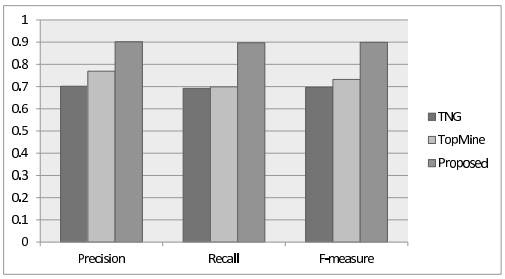
TABLE 6.2: Comparison of proposed document retrieval method with BM25, flat position index and inverted index baselines.

Method	**Dataset**								
	Reuters			**BBC**			**StackExchange**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
BM25[107]	0.6432	0.5920	0.6165	0.5177	0.4852	0.5009	0.6129	0.6233	0.6180
FPI[111]	0.7012	0.6903	0.6957	0.6911	0.7230	0.7066	0.6709	0.6237	0.6464
Inv. idx[112]	0.8126	0.8701	0.8403	0.7752	0.7659	0.7705	**0.8129**	**0.8788**	**0.8445**
Proposed	**0.9127**	**0.9201**	**0.9163**	**0.9322**	**0.9412**	**0.9366**	0.8099	0.8123	0.8110

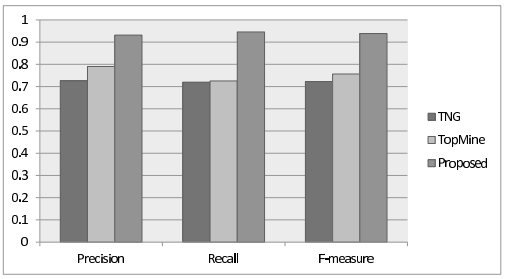
them to positively reward every concept which seems to be semantically valid. Fleiss Kappa score [113] has been used for the inter-annotator agreement. Measures such as $precision(P)$, $recall(R)$ and $F - measure(F)$ is used for computing the quality of extracted concepts to the human identified concepts. Here, true positive is defined as the number of overlapped concepts between human authored concepts and concepts generated by the algorithm, false positive is the number of extracted concepts that are not truly human authored concepts and false negative is the human authored concepts that are missed by the concept extraction method. Our precision, recall and f-measure values for concept and key-phrase extraction and document retrieval tasks are shown in Table 6.1 and Table 6.2 respectively. The precision values obtained are ranging from 90.12% to 94.58%. This shows that the proposed methods extract concepts and key-phrases that are close to real world when compared with the results of a crowd-sourcing experiment.

6.1.10 Evaluation of document retrieval

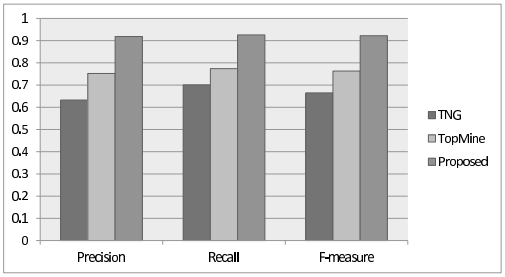
Here the proposed document retrieval method is evaluated and compares with three baselines: [107], [111] and [112]. Table 6.2 shows the performance comparison in terms of precision, recall and f-measure. From the values, it is evident that the proposed method significantly outperforms all the three baselines on two out of three datasets we



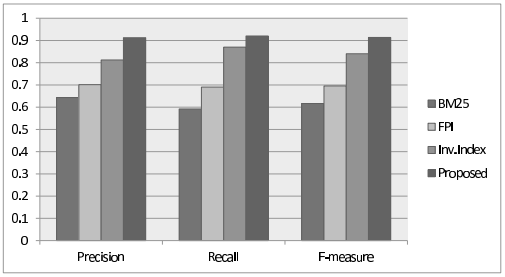
(A) Precision, Recall and F-measure values of concept extraction when compared with baseline methods on Reuters, BBC and StackExchange datasets



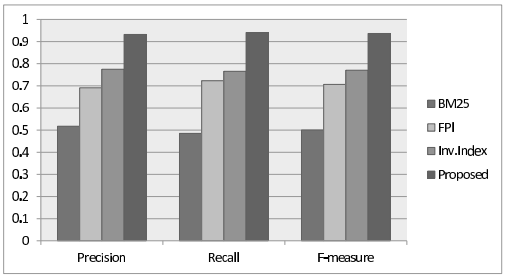
(B) Precision, Recall and F-measure values of concept extraction when compared with baseline methods on Reuters, BBC and StackExchange datasets



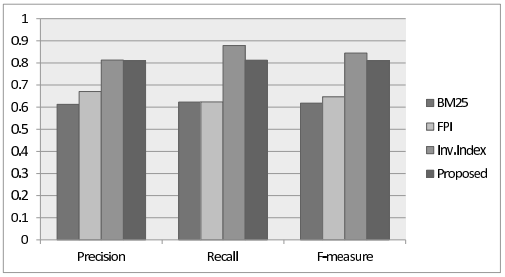
(C) Precision, Recall and F-measure values of concept extraction when compared with baseline methods on Reuters, BBC and StackExchange datasets



(D) Precision, Recall and F-measure values of document retrieval when compared with baseline methods on Reuters, BBC and StackExchange datasets



(E) Precision, Recall and F-measure values of document retrieval when compared with baseline methods on Reuters, BBC and StackExchange datasets



(F) Precision, Recall and F-measure values of document retrieval when compared with baseline methods on Reuters, BBC and StackExchange datasets

FIGURE 6.1: Precision, Recall and F-measure values of concept extraction and document retrieval when compared with baseline methods on Reuters, BBC and StackExchange datasets

have chosen. The recorded precision values for our proposed method are ranging from 91.27% to 94.12% for BBC and Reuters dataset respectively. For the StackExchange dataset, we have noticed that our closest competitor[112] achieved better precision than the proposed method. This is due to comparatively less number of concepts extracted for different categories considered for the StackExchange dataset such as academia,

android and cooking. After further observation, it is found that the noun-phrase tagger was unable to tag phrases from those less popular categories. Figure 6.1 shows the precision, recall and f-measure values of concept extraction task when compared with chosen baselines, and the precision, recall and f-measure values for the document retrieval task with baselines on different datasets.

6.2 Aspect Oriented Sentiment Analysis

Platforms such as social networks, micro-blogs, online review and discussion forums are growing very fast, and thus the need for analyzing sentiments of the users are also increasing. Sentiment analysis proves to be very effective to businesses and social domains because opinions matter and is critical for all human activities and thus they have become the key influence of human behavior. There are two broadly categorized types of the textual representation; one is fact another is opinion. While facts are objective expressions about entities, events, and their properties opinions are subjective expressions that describe people's sentiments towards entities or events. Product reviews in e-commerce platforms help the customer to analyze the product and take better judgments based on its previous usage history. Sentiment Analysis is the process of extracting subjective information from natural language text and it also expresses the opinion or view of a user towards a topic. The insights and applications from sentiment analysis is useful in other areas such as politics, law, policy making, sociology and psychology. The main challenges in sentiment analysis are that people express their opinions in complex ways and the relevance of lexical content in text. The words or short phrases are the building blocks of the sentiment expressed. Next step is to find the polarity keywords to identify the sentiment behind them. For example, there will be some relation between the words and positive reviews, and such keywords can be found out by data-driven methods for better accuracy. In the case of social networks like Twitter and Facebook, sentiment is most succinctly represented with emotions or smileys and in such scenario emotion recognition is performed to capture the sentiment expressed in them [121]. Therefore, customers are not only interested in the whole quality of the product rather than some specific aspects of the product also. For example, considering a laptop as an example, customers are not only interested in the whole product but also into some other aspects

such as 'screen quality', 'processor generation' or 'RAM speed'. This leads to the extraction of various aspects of a product[118] to perform sentiment analysis of product reviews. For each class of products, there is a set of very general categorical aspects that are common to all products in that class. Aspects can be identified either manually or an appropriate automated method. This work aims to identify aspects and then perform the sentiment analysis of product reviews using topic modeling. The main contributions of this approach are summarized as follows:

1. Proposes a topic modeling powered approach for aspect extraction and aspect based sentiment analysis from unstructured product reviews.
2. Verify experimentally the effectiveness and usefulness of proposed algorithms in extracting aspect oriented sentiments from real-world and dynamic datasets.

6.2.1 Proposed Approach

This section, describes the proposed framework for finding aspect oriented sentiment of product reviews expressed by customers. Those reviews mostly contain mixed opinions so that individual word level polarity computation may not be useful. Our topic modeling powered approach first models topics from customer reviews which are collection of word uni-grams and then extracts aspects from it. Based on the probability, we map these topics to corresponding aspects and then aspect level polarity is computed and the sentiments are classified using a machine learning classifier. The overall work-flow of the proposed approach is shown in Figure 6.2. This work uses Amazon product review dataset which consists of a large collection of review texts on mobile phones. Pre-processing is performed for converting raw text file into a well-defined sequence of linguistically meaningful units. This is an essential part of this system, since the characters, words and sentences identified at this stage are the basic elements passed to all other processing stages. The words such as "and", "the", "but" and other punctuations and hyperlinks are removed from the reviews. Then LDA algorithm is executed on top of this cleansed dataset to generate topics. The Algorithm 7 takes the product review corpus as the input and generate aspect specific sentences as the output. Each topic generated by LDA is mapped to a particular aspect and for each aspect, the algorithm scans the pre-processed corpus to find a match with topic words to extract aspect specific sentences. Thus, for each aspect, the aspect specific sentences will be written to a

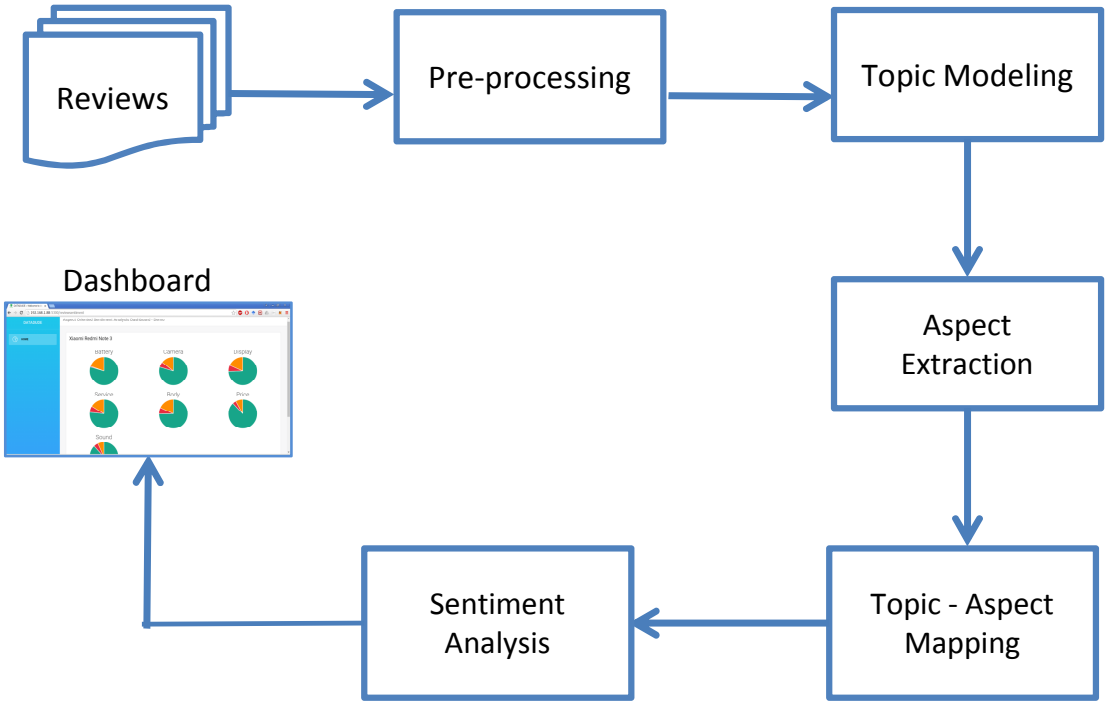


FIGURE 6.2: Overall workflow of the proposed approach for topic modeling powered aspect oriented sentiment analysis

Algorithm 7 : Topic Modeling & Topic-Aspect Mapping

```
function TopicAspectMapping
    Pre-process Review Corpus
    Remove Stopwords
    Remove Symbols
    Topics = LDA(Pre-processed Review Corpus)
    Aspects = Product Aspects
    for each "Topic" do
        a Topic-Aspect Mapping
    end for
    for each "Aspect" do
        Scan the Review text lines
        if line contains "Topic words" then
            Write "line" to the file
        else
            Skip the line
        end if
    end for
end function
```

text file for further processing. The Algorithm 8 takes the file containing aspect specific

Algorithm 8 : Aspect specific Sentiment Analysis

```
1: function AspectSpecificSentimentAnalysis
2:     Sentiment Analysis() = File
3:     while not the end of lines do
4:         Calculate Sentiment Polarity Score
5:         For each 'line'
6:         if Sentiment Polarity Score(line)>0 then
7:             Sentiment ← Positive
8:         end if
9:         if Sentiment Polarity Score(line)<0 then
10:            Sentiment ← Negative
11:        end if
12:        if Sentiment Polarity Score(line)=0 then
13:            Sentiment ← Neutral
14:        end if
15:    end while
16: end function
```

sentences as the input and it calculates the polarity score for each such sentence. The polarity score determines the sentiment of a sentence. If the polarity score of a sentence is greater than zero, the sentiment is "Positive" and if the polarity score is less than zero, then the sentiment is "Negative". If the polarity score is equal to zero, then the sentiment towards the particular sentence is said to be "Neutral". The Algorithm 9

Algorithm 9 : Finding sentiment on "Aspect"

```
1: function SentimentOnAspect
2:     for each "Aspect" do
3:         Positive ← Total Positive polarity count
4:         Negative ← Total Negative polarity count
5:         Neutral ← Total Neutral count
6:         if Positive > (Negative & Neutral) then
7:             Opinion ← Positive
8:         end if
9:         if Negative > (Positive & Neutral) then
10:            Opinion ← Negative
11:        end if
12:        if Neutral > (Positive & Negative) then
13:            Opinion ← Neutral
14:        end if
15:    end for
16: end function
```

takes the total polarity count of aspect specific sentences as the input and it expresses the sentiment on that particular aspect. If the total positive polarity count is higher than the negative and neutral polarity counts, then the sentiment towards the particular

aspect is said to be "Positive". Similarly, if the total negative polarity count is higher than the positive and neutral polarity counts, then the sentiment towards a particular aspect is said to be "Negative". If the total neutral count is higher than the positive and negative polarity counts,then the sentiment towards the particular aspect is said to be "Neutral".

6.2.2 Experimental Setup

This section details the experimental setup used for implementing proposed algorithms on Amazon product reviews available on public web. A brief description of datasets used in this experiment and the experimental testbed details and discussed below.

6.2.3 Dataset Description

We use Amazon product review dataset available at the url http://jmcauley.ucsd.edu/data/amazon/links.html. This dataset contains product reviews and associated metadata from Amazon, including 142.8 million reviews spanning for the period May 1996 - July 2014. This dataset includes reviews (ratings, text, helpfulness votes), product metadata (descriptions, category information, price, brand, and image features), and links (also viewed/also bought). It contains reviews for a variety of product categories such as books, electronics, Beauty, Automotive etc. This experiment used "Cell Phones and Accessories" category and used 194,439 reviews for our experiment.

6.2.4 Experimental Testbed

All algorithms discussed in this chapter were implemented on a server configured with AMD Opteron 6376 @ 2.3 GHz having 16 core processor and 16 GB of main memory. These algorithms were programmed using Python 2.7 and removal of stopwords and other pre-processing were done using Python NLTK (Natural Language Tool Kit) [210] and used stopword list for English language. For topic modeling, we have used MALLET (MAchine Learning for LanguagE Toolkit) which is freely available at the url http://mallet.cs.umass.edu/. For modeling topics, we have chosen 300 as the number of iterations in Gibbs sampling as by trail and error method we found that Gibbs sampling approaches the traget distribution after 300 rounds of iterations. The α and β

parameters of LDA was set to $50/Z$ and 0.01 respectively. The α controls the mixture of topics for any given document and the β hyperparameter controls the distribution of words per topic. If we decrease the value for α then the documents will likely have less of a mixture of topics and if we increase the value of α then the documents will likely have more of a mixture of topics. Likewise, if we decrease the value of β then the topics will likely have less words and if we increase the value of β then the topics will likely have more words.

6.2.5 Baselines

Proposed algorithms are compared with two state-of-the-art methods available in the literature for aspect oriented sentiment analysis. The first baseline [152] is an approach that uses ontology for extraction of aspects and eliminates lexicon based sentiment scoring issues. Second baseline[145] is a linguistic rule based approach for aspect level sentiment analysis of movie reviews. The authors have devised a linguistic rule-based approach which identifies the aspects from movie reviews, locates opinion about that aspect and computes the sentiment polarity of that opinion using linguistic approaches [145]. A comparison result in terms of Precision, Recall and Accuracy is given in the experimental results section.

6.2.6 Results and Evaluation

This section details the results obtained from the experimental setup discussed in Section 6.2.2. The topics generated by the LDA algorithm is shown in Table 1. Due to space constraints, we show only first 5 topics generated by LDA. These topics are then mapped to aspects such as Network, Battery, Price which are identified from the product descriptions. The output of the mappings are shown in Table 6.4. In Table 6.5, we show the polarity score and sentiment of review text for the aspect - "Battery". Various polarity count for such identified aspects are given in Table 6.6. In Table 6.7 we outline the result of our aspect oriented sentiment analysis algorithm. For further analysis and interpretation, we plot the True Positive Rate (TPR) and Flase Positive Rate (FPR) using ROC plot and such graphs for the aspects "Battery and Body", "Display and Network", "Price and Service" are shown in Figure 6.3, Figure 6.4 and Figure 6.5 respectively.

Proposed algorithms are compared with two state-of-the-art methods for establishing the usefulness and better accuracy of our proposed algorithms. The results in terms of precision, recall and f-measure is given in Table 6.8 and the data clearly shows that our proposed method outperforms our chosen baselines for all the aspects we have considered. Figure 6.6 shows a user interface of the web application developed to visualize the results of our aspect oriented sentiment analysis framework implemented on real-time data. The current screenshot shows the result of aspect based sentiment analysis for the mobile Xiaomi Redmi Note 3 and the corresponding user reviews are collected from the famous customer review platform called mouthshut.com. Our proposed algorithms show the user sentiments on different aspects of the mobile and also the algorithms are found to be scalable with dynamic data as well. This system has already been implemented for some e-commerce vendors and they are using it for a pilot version.

TABLE 6.3: Topics extracted by LDA algorithm (Due to space constraints we show only first 5 topics generated)

Topic	Topic keywords
Topic 1	software treo palm keyboard blackberry device pda data email pc sync computer windows web internet mobile version support contacts
Topic 2	charger car cable work usb works plug charging power fine cord adapter charge worked device port doesn plugged connection
Topic 3	buy money worth love cheap thing blue bought quot buying don people dont color waste paid cost cool wanted
Topic 4	phone phones cell motorola nokia razr samsung lg reception features flip dropped calls drop basic sanyo vx razor krzr
Topic 5	small size pocket design makes bit feel find big large nice hand easily carry fact easier smaller style buttons

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{6.3}$$

$$Precision = TP/(TP + FP) \tag{6.4}$$

$$Recall = TP/(TP + FN) \tag{6.5}$$

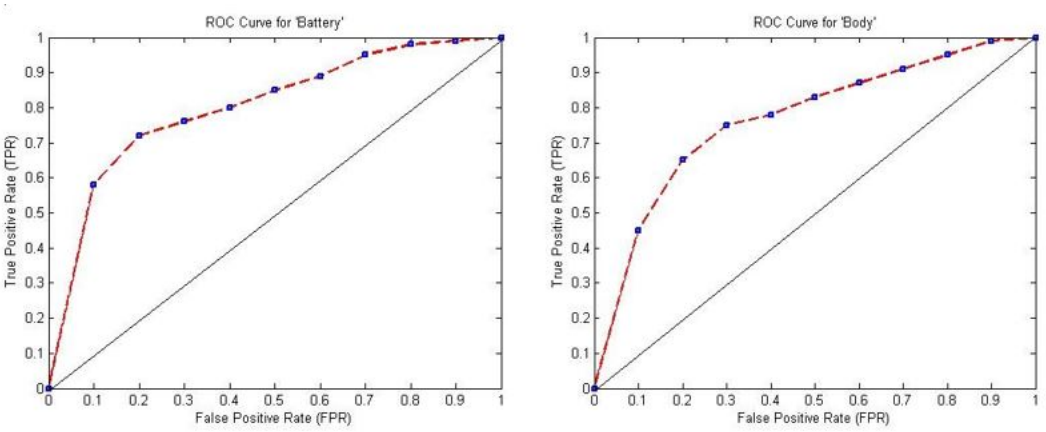


FIGURE 6.3: ROC plots for 'Battery' & 'Body'

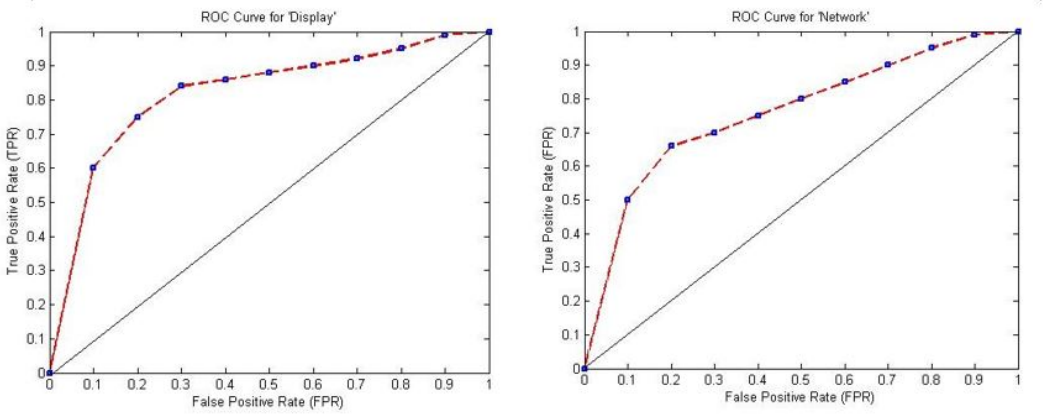


FIGURE 6.4: ROC plots for 'Display' & 'Network'

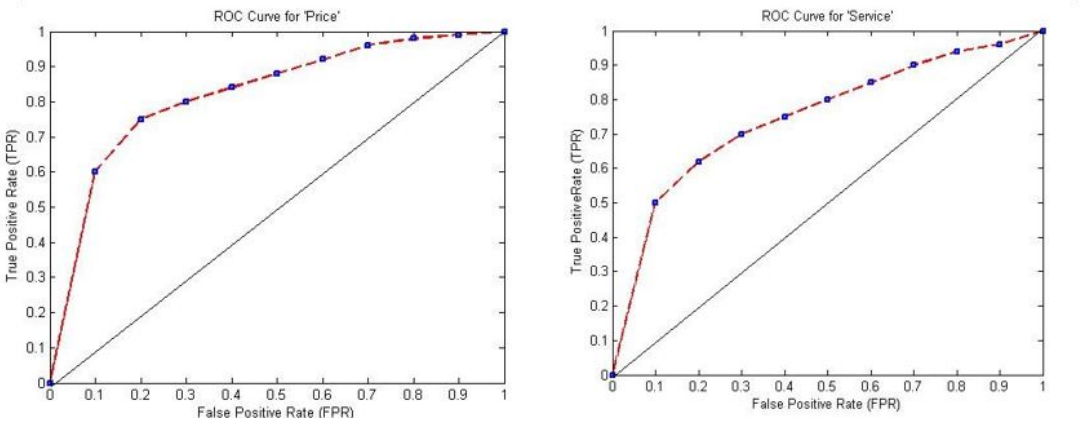


FIGURE 6.5: ROC plots for 'Price' & 'Service'

TABLE 6.4: Result of Topic - Aspect mapping. Based on number of words that are related to an aspect, we map the entire topic to corresponding aspect.

Topic	Topic words	Aspect
Topic 1	software treo palm keyboard blackberry device pda data email pc sync computer windows web internet mobile version support contacts	Network
Topic 2	charger car cable work usb works plug charging power fine cord adapter charge worked device port doesn plugged connection	Battery
Topic 3	buy money worth love cheap thing blue bought quot buying don people dont color waste paid cost cool wanted	Price
Topic 4	phone phones cell motorola nokia razr samsung lg reception features flip dropped calls drop basic sanyo vx razor krzr	Body
Topic 5	small size pocket design makes bit feel find big large nice hand easily carry fact easier smaller style buttons	Battery

TABLE 6.5: Polarity score and sentiment of Review texts for the aspect - 'Battery'

1	The first battery, from a different vendor was not good and needed charging daily.	-0.025	Neg	Correct
2	The battery gets hot when charging so I would not use this as my main battery.	0.20	Pos	Wrong
3	The battery was charging all night long and then this morning the battery died.	-0.05	Neg	Correct
4	Powerful battery, needs charging once in 3-4 days.	0.3	Pos	Correct

6.3 Information Retrieval from Social Network Data

Recent years witnessed an exponential growth in the number of social network users due to widespread popularity of mobile platforms and applications. This caused huge volume of messages to be generated while these users communicate each other [177]. People create and share pieces of text content that may be about some disasters such as earthquakes [172], or community view on elections [173], or new governmental agendas [175], or even opinions on a newly launched product in the market [174]. Thus retrieving relevant and useful information from those dynamically generated unstructured data is crucial as those messages may contain useful insights for any interested stakeholders.

TABLE 6.6: Polarity count for various Aspects

Aspect	**Total**	**Pos**	**Neg**	**Neu**
Battery	26323	7890	5591	12842
Body	23840	14446	4148	5246
Display	18421	10332	3451	4638
Network	17884	1577	14075	2232
Price	34135	18333	5816	9986
Service	37323	7701	16302	13320
Sound	67514	42366	12142	13006

TABLE 6.7: Resut of aspect oriented sentiment analysis.

No	**Aspect**	**Sentiment**
1	Battery	Neutral
2	Body	Positive
3	Display	Positive
4	Network	Negative
5	Price	Positive
6	Service	Negative
7	Sound	Positive

TABLE 6.8: Performance of our proposed aspect oriented sentiment analysis when compared with chosen baselines [152][145]

	Baseline 1[152]			**Baseline 2**[145]			**Proposed Method**		
Aspect	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Battery	0.6158	0.5485	0.5802	0.7155	0.6252	0.6673	**0.7845**	**0.7544**	**0.7691**
Body	0.5952	0.5812	0.5881	0.6925	0.6152	0.6515	**0.8012**	**0.7955**	**0.7983**
Display	0.6266	0.5984	0.6121	0.7852	0.6951	0.7374	**0.8101**	**0.7920**	**0.8009**
Network	0.6011	0.6652	0.6315	0.7241	0.7129	0.7184	**0.8522**	**0.8311**	**0.8415**
Price	0.7010	0.6123	0.6536	0.6612	0.6522	0.6566	**0.9054**	**0.8856**	**0.8953**
Service	0.6840	0.6551	0.6692	0.7852	0.7210	0.7517	**0.8995**	**0.7845**	**0.8380**
Sound	0.6450	0.5963	0.6196	0.8021	0.8045	0.8032	**0.9388**	**0.9012**	**0.9196**

Information shared by social network users are archived at the service provider's end and there may be situations where the users or other interested parties want to retrieve relevant messages from these enormous archives of data. So the need for intelligent and text understanding search, ranking and retrieval algorithms is ever increasing and crucial. The state-of-the-art algorithms that operate on a set of static documents may not suite in this situation due to the dynamic nature of information generated in social networks [176].

Text mining and information retrieval research communities devised many methods and algorithms for intelligent information retrieval from large unstructured text archives. These algorithms exhibits varying levels of accuracy when dealing with static

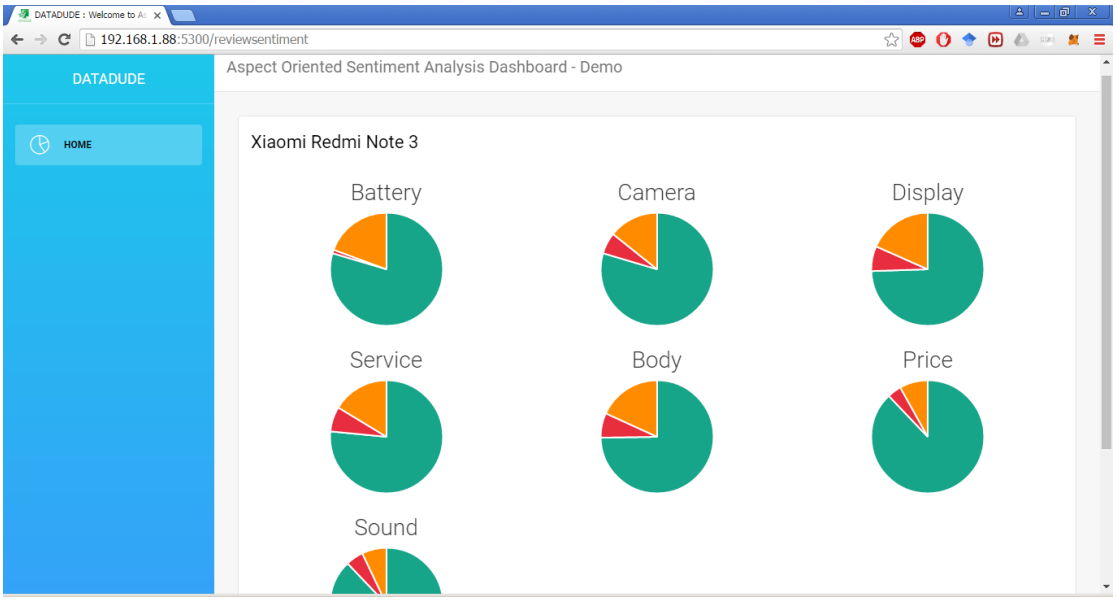


FIGURE 6.6: Screenshot of web application developed for finding aspect oriented sentiment of customers for mobile phones and accessories using proposed algorithms. This system has been extended to production level and a number of vendors are already using pilot version of the same.

documents but performance was degraded on dynamically generated data such as social network messages[178]. Due to this dynamic nature, social networking platforms require special type of algorithms to deal with the complexities in search and retrieval of relevant information. Even though a vast array of researches on this dimension have been reported, there are still a lot of avenues where we can make use of advanced text mining and natural language processing techniques for improving accuracy of such algorithms.

Topic modeling is a suite of text understanding algorithms that can leverage hidden themes from large archives of unstructured text data [179]. Several variations of such algorithms with varying degrees of accuracies have been devised in the past with only difference in the assumption they made in the process of generating themes. Topic models were further extended for mining topics from social networks such as Twitter and are used by social network research communities in detecting change in topics over a period of time [180][181][182][183]. On the other hand, deep learning approaches such as word2vec - distributed representation of words in a vector space [162], and phrase2vec - distributed representation of phrases in a vector space[163] etc. gained significant attention that considered the 'context' in which words or phrases have been used. This work attempts to make use of topic modeling[179] and distributed representation of phrases (phrase2vec) [163] for scoring social network messages and shows the application on a

phrase query based information retrieval task. The main research objectives are (1) Introduce the task of topic modeling guided scoring of unstructured text and its applications in online social networks, (2) Propose a framework that combines topic modeling and distributional representation of phrases for better retrieval of relevant information from text messages, (3) Verify experimentally the effectiveness of the proposed framework in extracting relevant and useful information from social network messages, (4) Compare proposed approach with state-of-the-art models to establish the fitness of our method in scoring social network messages and phrase query based information retrieval. The main contributions are summarized as follows:

1. Proposes a framework for retrieving relevant and useful content from large collection of social network messages.
2. Shows that a hybrid approach combining topic modeling and distributed representation of phrases can better retrieve and rank relevant information.
3. Experimental comparison of the proposed method with other state-of-the art approaches establishes the effectiveness of the proposed method.

6.3.1 Background: Distributional Representation of Phrases

This section briefly outlines a background on distributional representation of words and phrases which are current hot research areas in text mining and natural language processing. This area of research is attempting to develop methods for measuring the level of semantic similarities among linguistic elements in large samples of unstructured data. The idea of distributional representation of words and phrases got significant attention from text mining and natural language processing practitioners as these representations could project the semantic similarities of those linguistic units in high dimensional vector space. The seminal work in this dimension that explored beyond traditional syntactic and probabilistic approach was introduced by Tomas Mikolov which represented word unigrams in high dimension vector spaces [162]. This paper contributed two strong architectures - Continuous Bag of Words (CBOW) and Skip-gram - where the former can predict the current word based on the context and the latter can predict the context based on a given word. These models were initially used to work with word unigrams,

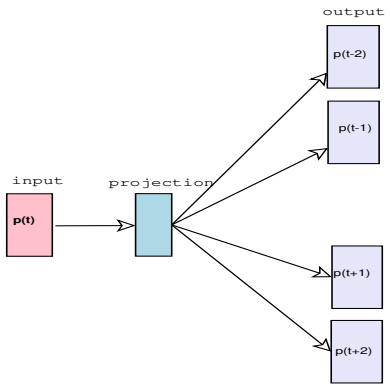


FIGURE 6.7: Skip-gram model for phrases used in this work. $p(t)$ in the input layer denotes the input/query phrase and $p(t-2)$, $p(t-1)$, $p(t+1)$, $p(t+2)$ in the output layer denotes phrases from distributed representation which are semantically closer to $p(t)$

later methods for distributed representation of phrases [163], sentences, paragraphs and even documents [164] have been reported.

In this work, we use the Skip-gram architecture shown in Figure 6.7 for learning distributed representation of phrases identified from tweets. In phrase2vec model, a large number of key-phrases are identified and during training, they are treated as individual tokens, with the objective to find phrase representations that may predict surrounding phrases in a sentence or given document [163]. Given a phrase (query), this work also utilizes this intuition to find semantically related phrases that are identified from tweets beforehand and represented in a distributed high dimensional vector space.

6.3.2 Proposed Approach

This section details the proposed framework for topic modeling powered distributional semantics based information retrieval from online social networks. The overall workflow of the proposed framework is shown in Figure 6.8. The framework consist of two modules in which the first one requires an online processing and the second one is executed offline. A detailed description of modules are outlined in the following sections.

6.3.2.1 Online Processing

The online processing module of this framework is used for extracting noun-phrases, generating topics and creating distributed vector representation of identified noun-phrases.

This proposed method uses an off-the-shelf tool for extracting noun-phrases from pre-processed tweets that will identify and extract semantically valid phrases. The decision of using an off-the-shelf tool was taken as we observed that the custom natural language processing rules based on parts-of-speech tagging is computationally expensive in terms of both memory and time thus the scalability of such an approach is very weak. Latent Dirichlet Allocation (LDA) [166] algorithm is then executed on tweets for generating latent topics that are word uni-grams. Now we introduce our formula for scoring noun-phrases in topics generated. The score for a noun-phrase n in a given topic t is computed as the product of probability of t over the document d multiplied by the frequency of noun-phrase in the document. Notationally,

$$score(n,t) = \sum_{document_d} p(t|m) freq(n,m) \tag{6.6}$$

where $p(t|m)$ is the scoring for the message (tweets)-topic pair obtained from the LDA and $freq(n,m)$ is the frequency of noun-phrase n in tweet message m. After calculating score for each noun-phrase, we filter top k noun-phrases according to the score. The proposed algorithm, Algorithm 10, which takes a set of topics which are probability distribution over words, generated by Latent Dirichlet Allocation (LDA) [166] and a set of dynamic corpus of tweet messages D, which are collected from Twitter, which is one of the widely used online social networks. Using an off-the-shelf noun-phrase tagger, the algorithm first tags all the noun-phrases from dynamic corpus D. The frequencies of noun-phrase in documents are then computed. The probability distribution of topic over documents, $p(t|m)$ is computed from LDA output. Then the score of already tagged noun-phrases in documents, $score(n,m)$ calculated as the product of $p(t|m)$ and $freq(n,m)$. Finally, we find top - k noun-phrases according to the score, $score(n,t)$ calculated using Equation 6.6.

6.3.2.2 Offline Processing

Offline module of this framework deals with ranking messages, given a phrase query by the user. The algorithm we have devised for retrieving relevant messages for a user specified query is given in Algorithm 11. Firstly, the phrase query given by the user is enhanced and this is achieved by using the phrase2vec embedding we have created during online processing stage. The query q is passed to the phrase2vec model and the output

Algorithm 10 Topical scoring noun-phrases from tweets

```
1: function ComputeScore(C,T)
2:     for each message m from the tweet corpus C do
3:         Extract noun-phrases from m
4:     end for
5:     for topic t and noun-phrase n do
6:         score(n,t) = ∑_{document_d} p(t|m) freq(n, m)
7:     end for
8:     for each topic t do
9:         Find the top-k noun-phrases according to the score(n, t)
10:    end for
11: end function
```

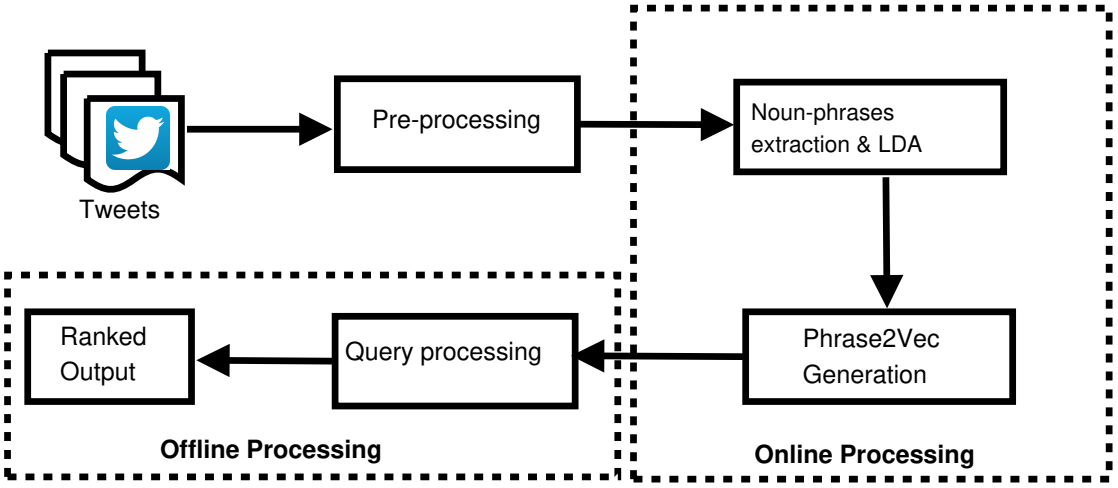


FIGURE 6.8: : Overall workflow of the proposed framework

(semantically similar phrases for q) q' is used to score the messages. The score of enhanced query q' is calculated using the formula $score(q', t) = \sum_{document_d} p(t|m) freq(q', m)$ where $p(t|m)$ is the probability of a topic t in a tweet message d and $freq(q', d)$ is the number of times enhanced query (obtained from phrase2vec, distributed representation of phrases) term appears in that document. After scoring, we select top - k documents according to the score. Thus in this approach, rather than searching for the presence or absence of user supplied query, we induct semantics using distributional representation of phrases in the vector space that will lead multiple ways to score and retrieve relevant messages from the collection. We implemented Algorithm 10 and Algorithm 11 for showcasing the usefulness and efficiency of our proposed method and a detailed explanation is given in Section 6.3.3.

Algorithm 11 : Query enhancement and tweet messages scoring

function RANKANDRETRIEVETWEETS(q)
  for q **do**
    Find related phrases from phrase2vec
    Enhance the query q to q'
  end for
  Retrieve topics for q' using scoring function, score(q',t)
  Score(q',t) = $\sum_{document_d} p(t|m) freq(q', m)$
  Use score(q',t), to get topic - message distribution
  Select top - k tweets according to the distribution
end function

FIGURE 6.9: A snapshot of tweets collected for the hashtag "#DeMonetisation"

6.3.3 Experimental Setup

This section details the experimental setup used for the proposed information extraction experiment. A complete description of the dataset used and the experimental testbed details are highlighted.

6.3.4 Dataset Description

This work uses messages from Twitter (tweets) for this experiment and collected tweets for 5 hashtags - "#KashmirUnrest", "#SurgicalStrike", "#Demonetisation", "#Election2016" and "#MakeInIndia". We have collected 25000 tweet messages for each hashtag from March 2017 to August 2017, using SocioViz[1] platform. Thus a total of 125000 tweets are used for carrying out this experiment. A snapshot of the tweets collected for the hashtag "#DeMonetisation" is shown in Figure 6.9

[1]http://www.socioviz.net [Last accessed on: 17/10/2017]

6.3.5 Phrase Extraction

The noun-phrases from pre-processed tweets are generated using two different methods. Initially, we have implemented natural language processing (NLP) rules based approach and later used an off-the-shelf tool. For NLP approach, firstly, a parts-of-speech (POS) tagging on pre-processed messages was done and then written NLP rules by combining different POS tags extracted. Some of such rules used for keyphrase extraction are shown in Table 6.9. More information on POS tags can be obtained from the url https://cs.nyu.edu/grishman/jet/guide/PennPOS.html. POS tagged words were combined by using an underscore symbol ("_") to form phrases. As we found NLP rules based on POS tagging took more time for generating keyphrases compared to TextBlob[1], which is an off-the-shelf-tool for key-phrase extraction, we decided to use TextBlob library for this experimentation. TextBlob uses NLTK [168], which is a collection of libraries and programs written in Python programming language and is used for symbolic and natural language processing tasks for English language. Since NLTK[168] is used by industry level applications that involve natural language processing and TextBlob uses NLTK[168] as its backbone, we found this library performs well in almost all domains in English language and specifically the messages (tweets) we have collected using the mentioned hashtags.

6.3.6 Experimental Testbed

All algorithms proposed were implemented using Python 2.7 and run on a server computer with AMD Opteron 6376 2.3 GHz 16 core processor and 16 GB of main memory. The collected tweets were thoroughly pre-processed to remove noises such as hashtags, web urls and special characters. Pre-processed tweet messages were saved in plain text files and prepared an experiment ready copy of the dataset. For topic generation from these tweets, we have executed Latent Dirichlet Allocation (LDA) algorithm and used MALLET (Machine Learning for Language Toolkit) tool available at http://mallet.cs.umass.edu/download.php. The number of iterations of Gibbs sampling used in this experiment is set to 300, as we found that Gibbs sampling usually approaches the target distribution after 300 iterations, using trial and error approach. The other two parameters α and β for LDA is set as $\alpha = 50/Z$ and $\beta = 0.01$ respectively for this

[1]https://textblob.readthedocs.io/en/dev/ [Last accessed on: 17/10/2017]

TABLE 6.9: Common NLP rules for extracting semantically valid phrases from tweets

Sl. no.	Pattern
1	JJ + NN [Adjective + Noun]
2	JJ + NNS [Adjective + Noun (Plural)]
3	JJ + VBG [Adjective + Verb (gerund or present participle)]
4	JJR + VBG [Adjective (comparative) + Verb (gerund or present participle)]
5	NN + NN [Noun (singular or mass) + Noun (singular or mass)]
6	NN + NNS [Noun (singular or mass) + Noun (plural)]
7	NN + VB [Noun (singular or mass) + Verb (base form)]
8	NN + VBG [Noun (singular or mass) + Verb (Verb, gerund or present participle)]
9	NN + VBN [Noun (singular or mass) + Verb (past participle)]
10	NNP + NNS [Proper noun (singular) + Noun (plural)]
11	NNS + VB [Noun (plural) + Verb (base form)]
12	NNP + NNS [Proper noun (singular) + Noun (plural)]

experiment. For generating phrase2vec embedding of the data, the public code library available at https://github.com/zseymour/phrase2vec is used which is an implementation of phrase2vec from modified word2vec code using C language. Noun-phrase tagged files are given as input and this code generated the distributed phrase representation of messages. After the online processing stage, experiment has been extended for scoring and ranking messages, given a user query. For scoring, dynamics of the topic (generated by LDA), phrases (generated by TextBlob) and the messages are used and it will lead multiple ways of reaching the message from the query phrase. Firstly, the given query is enhanced using the phrase2vec embedding generated and stored in the online processing stage of the experiment. Given a phrase query, the pre-trained model will give a set of phrases which are semantically related to the given query and is calculated as the cosine distance between two vectors in the distributional vector space. This enhanced query set is used for extracting relevant messages, rather than using the simple query phrase based information retrieval. Thus the message would be scored high if it contains a lot of phrases that are similar to the query phrase which is our hypothesis for this experiment. Proposed ranking approach is compared with four state-of-the-art baselines such as BM25 [167], a flat position index based approach [169], an inverted index based approach [170] and an entity based retrieval approach [171]. Out of these three approaches, BM25 [167] is a term matching based approach, other two phrase based retrieval mechanisms that uses flat position index and inverted index as their primary data structures and an entity-language based approach. We implemented them using Python programming language with the help of public code repositories available at

https://www.rosettacode.org/wiki/Inverted_index_Python and https://github.com/matteobertozzi/b code/blob/master/py-inverted-index/. Detailed analysis and evaluation of results from these experiments are given in Result and Evaluation section.

6.3.7 Results and Evaluation

In this section, evaluation of our proposed framework for efficient information retrieval from online social network is discussed. The method is compared with four other state-of-the-art baselines:

- **BM25 (Best Match 25)** [167]: In information retrieval, BM25, known as Okapi BM25, is a widely used ranking function by search engines to rank a large collection of documents given a search query. It is a probabilistic retrieval method that act on bag-of-words created from the document collection using the following equation [167]:

$$score(q,d) = \sum_{i=1}^{|q|} idf(q_i).\frac{tf(q_i,d).(k_1+1)}{tf(q_i,d)+k_1.(1-b+b.\frac{|d|}{avgdl})} \tag{6.7}$$

  where, $tf(q_i, d)$ is q_i's term frequency in the document d, $|d|$ is the length of the document d in words, and $avgdl$ is the average document length in the text collection from which documents are drawn. For this experiment, we have used the parameters $k1 = 1$, $k2 = 0$, $k3 = 1$ and $b = 0.5$ for initial set of experiments and then later finalized the values of $k1$ as 1.7 and b as 0.95 by trial and error approach and found these values give optimal results.

- **FPI (Flat Position Index) based approach**[169]: In this method, the authors propose a flat position index for phrase query evaluation. The entire document collection is treated as huge sequence of tokens and each token is represented by one flat position. This method could reduce the index size and speed up phrase querying substantially.

- **Inverted Index based approach**[170]: This method introduced a set of inverted indexes that work for strings and phrases that works well for phrase searching. This approach also highlighted top - k based retrieval under relevance metrics like

$tf - idf$ and frequency [170].

- **Entity - Language Model based approach**[171]: This paper introduced entity-based language models for document retrieval that utilizes information about single terms in the query and documents as term sequences marked as entities. This approach projected the merits of using language models for retrieval tasks and shows that language models can be effectively used for cluster based document retrieval[171].

The proposed algorithm is compared with four different state-of-the-art approaches for information retrieval task - [167][169][170] and [171]. The performance comparison in terms of precision, recall and f-measure for hashtags "#KashmirUnrest", "#Surgical-Strike", and "#Demonetisation" are shown in Table 6.10 and the hashtags "#Election2016", "#MakeInIndia", and "#TrumpRussia" are shown in Table 6.11 respectively. From the tables, it is clear that our proposed framework outperforms the chosen baselines in term of precision, recall and f-measure. For the hashtag "#KashmirUnrest", our method achieved 0.9114, 0.9274 and 0.9193 as precision, recall and f-measure respectively and the entity based approach [171], which is the closest competitor could achieve a f-measure value of 0.8065 only. For the next four hashtags such as "#SurgicalStrike", "#Demonetisation", "#Election2016" and "#MakeInIndia", our method showed better f-measure which was 0.9393, 0.9226, 0.9422 and 0.9341 respectively. But for the hashtag "#TrumpRussia", we found that our closest competitor for other hashtags, the entity based approach [171] showed better performance in terms of f-measure. This method showed 0.8984 as the f-measure and our method could achieve an f-measure of 0.8472 only. Upon further and detailed analysis, we found that the number of noun phrases generated from the tweets against the hashtag "#TrumpRussia" was less compared to the other hashtags. In future work, we may address this by comparing different tools for key-phrase extraction and enrich the process of phrase generation by incorporating custom natural language rules. A graph representing the precision, recall and f-measure comparison for the different hashtags used in the experiment is shown in Figure 6.10.

TABLE 6.10: Comparison of proposed method with BM25, flat position index, inverted index and entity based baselines for twitter hashtags - "#KashmirUnrest", "#SurgicalStrike", and "#Demonetisation"

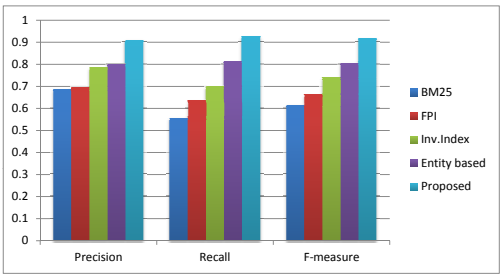
Method	**Hashtags**								
	"#KashmirUnrest"			**"#SurgicalStrike"**			**"#Demonetisation"**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
BM25[167]	0.6879	0.5578	0.6160	0.6643	0.5210	0.5839	0.6996	0.5945	0.6427
FPI[169]	0.6952	0.6354	0.6639	0.6930	0.7102	0.7014	0.6104	0.6332	0.6215
Inv. idx[170]	0.7855	0.6985	0.7394	0.6969	0.7011	0.6989	0.8014	0.7685	0.7846
Entity based [171]	0.8021	0.8110	0.8065	0.8215	0.7985	0.8139	0.8129	0.8854	0.8476
Proposed	0.9114	0.9274	0.9193	0.9375	0.9412	0.9393	0.9199	0.9254	0.9226

TABLE 6.11: Comparison of proposed method with BM25, flat position index, inverted index and entity based baselines for twitter hashtags - "#Election2016", "#MakeInIndia", and "#TrumpRussia"

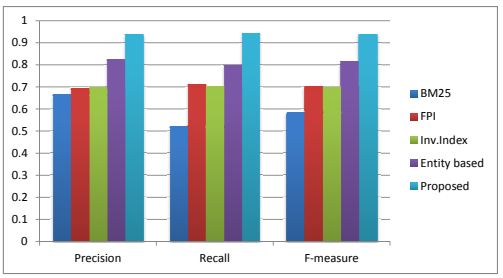
Method	**Hashtags**								
	"#Election2016"			**"#MakeInIndia"**			**"#TrumpRussia"**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
BM25[167]	0.6954	0.5874	0.6368	0.5685	0.4995	0.5317	0.6685	0.6235	0.6452
FPI[169]	0.7045	0.7125	0.7084	0.6696	0.7895	0.7246	0.6336	0.7452	0.6848
Inv. idx[170]	0.8022	0.8452	0.8231	0.7785	0.7452	0.7614	0.8457	0.8741	0.8596
Entity based [171]	0.8544	0.8630	0.8586	0.8741	0.8452	0.8594	0.9025	0.8944	0.8984
Proposed	0.9527	0.9320	0.9422	0.9740	0.8974	0.9341	0.8952	0.8041	0.8472

6.3.8 Conclusions

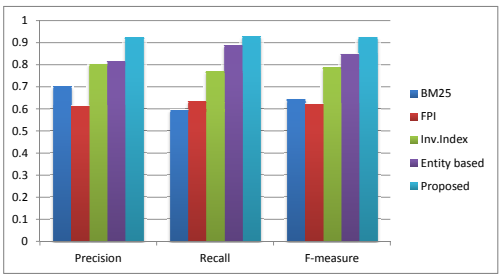
This chapter discussed a few potential real-world applications using proposed algorithms introduced in Chapter 3, Chapter 4 and Chapter 5. A method for phrase query based document ranking was proposed which can extract potentially meaningful and interpretable concepts and key-phrases from statistically generated topics. The usefulness of the proposed algorithm was shown in document ranking and retrieval task. Later, an approach for topic modeling guided aspect extraction and sentiment analysis was discussed which can classify sentiments based on aspects rather than just checking the word level polarity score. The proposed approach has been implemented in real-world dynamic dataset and found to be very useful for sentiment analysis tasks. Lastly, a framework for information retrieval from dynamic online social network is introduced which can score and retrieve messages, given a user query. The distributional semantic based approach outperformed some of the state-of-the-approaches for phrase based information retrieval task. These real-world applications show the usefulness of proposed algorithms in this thesis. The next chapter concludes the thesis and some interesting dimensions of future work are also discussed.



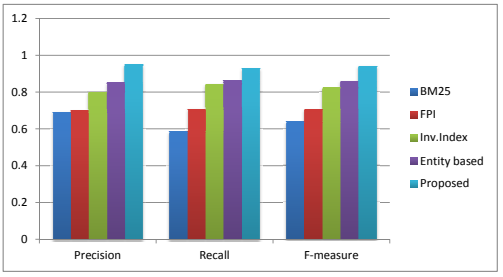
(A) Precision, Recall and F-measure comparison for the hashtag - "#KashmirUnrest"



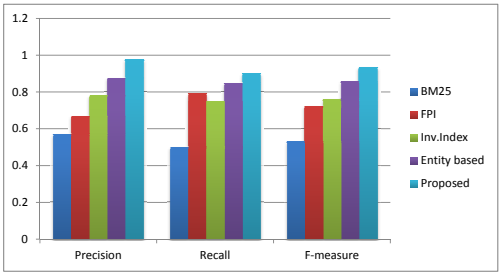
(B) Precision, Recall and F-measure comparison for the hashtag -"#SurgicalStrike"



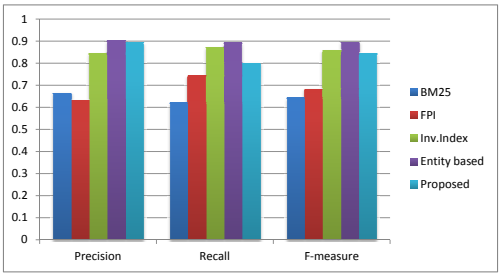
(C) Precision, Recall and F-measure comparison for the hashtag - "#Demonetisation"



(D) Precision, Recall and F-measure comparison for the hashtag - "#Election2016"



(E) Precision, Recall and F-measure comparison for the hashtag - "#MakeInIndia"



(F) Precision, Recall and F-measure comparison for the hashtag - "#TrumpRussia"

FIGURE 6.10: Precision, Recall and F-measure values obtained for the proposed algorithm on tweets retrieval task.

CHAPTER 7

Conclusions and Future Work

7.1 Conclusions

Recent years witnessed huge data explosions that caused an unprecedented growth of data being generated across the globe. A major share of this data is unstructured text that makes it very difficult to store, process, analyze, and more importantly to identify patterns and actionable knowledge out of it. Text mining researchers and communities have devised many algorithms with varying degrees of success, for unearthing patterns out of this goldmine of textual data. There are vast arrays of algorithms available which do not consider the meaning or semantics of the text they operate upon. Algorithms such as clustering, classification and topic modeling leveraging well defined mathematical and statistical methods and assumptions, fits in this category. There are several research works reported in the literature that attempts to incorporate meaning / semantics into text mining applications using handcrafted semantic structures such as ontologies. These techniques use human created / curated ontologies such as DBPedia, Probase to link and leverage the association and meaning between linguistic units and textual components. But there are several arenas where domain ontologies are incomplete or not available for building meaning aware computing systems. Manual creation of such ontologies from large collection of unstructured text is not efficient and labour intensive. This creates the need of novel text mining algorithms that can analyze large bodies of unstructured text data and unearth potentially useful and previously unknown patterns that may aid the process of constructing conceptual structures similar to ontologies.This research work reported in this thesis is an effort in this direction.

The algorithms proposed in this thesis tries to bridge the gap between statistical output generated by algorithms and human crafted / curated ontologies to build conceptual structures with potential applications in meaning aware / semantic computing paradigms. Some of the dimensions explored in this work are (i) Inducing semantics on top of statistically generated text outputs that may contribute to the automated generation of ontologies. (ii) Explore the dimensions for generating hierarchies of concepts and key-phrases that may find applications in meaning-aware computing paradigms. (iii) The use of formal concept analysis and distributional semantics in conceptual knowledge discovery process (iv) To show the proposed algorithms in real-world problems such as document ranking, sentiment analysis and knowledge discovery in e-commerce. In this direction, three methods were proposed.

- Topic Modeling Guided Concept Extraction and Concept Hierarchy Generation: This work attempted generating key-phrases and concepts from "topics" generated by a topic modeling algorithm. A linguistic analysis has been done on top of the generated topics to extract key-phrases and concepts and found that the proposed algorithm can leverage knowledge from large bodies of unstructured text if the process is guided by a topic modeling algorithm.

- Concept Lattice Generation and Relationship Extraction: This work is an effort to use Formal Concept Analysis (FCA) for analyzing and extracting relationships from unstructured text documents. Concept lattices are used to represent the concepts extracted by our proposed method and FCA is then applied on this lattice to extract relationships among them.

- Conceptual Knowledge Discovery using Distributional Semantics: Distributional representation of linguistic elements such as words, phrases, sentences etc. is a comparatively new area in text mining and natural language processing which has several applications. This work is an attempt to combine distributional representation of phrase (phrase2vec), short text conceptualization and clustering technique to generate semantically rich and human interpretable clusters.

The use of proposed algorithms in real-world applications such as phrase query based document retrieval, aspect oriented sentiment analysis and information retrieval from social network data are also discussed in this thesis. From the theoretical analysis and

experimental results, it is observed that the proposed methods are promising candidates in the context of conceptual knowledge discovery from unstructured text.

7.2 Future Works

The theoretical and empirical analysis done through this thesis work offers the following avenues for further explorations.

- Building higher level abstractions of concepts and key-phrases such as nested hierarchies is a potentially explorable problem.
- Concept graph learning using deep learning techniques is another potential dimension for further exploration.
- Explorative analysis of concept graphs and optimizing the traversal for applications such as question answering systems, document summarization is another interesting research problem.
- Design of scalable deep learning algorithm for deep text mining to unearth semantically rich patterns is a highly relevant research problem with many potential applications.

Bibliography

[1] Pospiech, Sebastian, Martin Pelke, and Robert Mertens. Semi-automated Ontology Creation for Semantic Search in Business Process Exploration *IEEE Tenth International Conference on Semantic Computing (ICSC).*, 2016.

[2] Marujo, Luís, et al. Exploring events and distributed representations of text in multi-document summarization. *Knowledge-Based Systems*, 94, 33-42, 2016.

[3] Manek AS, Shenoy PD, Mohan MC and Venugopal KR. Aspect term extraction for sentiment analysis in large movie reviews using Gini Index feature selection method and SVM classifier. *World Wide Web*, 1-20, 2016.

[4] Steyvers M and Griffiths T. Probabilistic topic models. *Handbook of latent semantic analysis.*, 427(7):424-40, 2007.

[5] Sanderson M and Croft B. Deriving concept hierarchies from text. *InProceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 206-213, 1999.

[6] Lindsey RV, Headden III WP and Stipicevic MJ. A phrase-discovering topic model using hierarchical pitman-yor processes. *InProceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, (pp. 214-222), 2012.

[7] El-Kishky A, Song Y, Wang C, Voss CR and Han J. Scalable topical phrase mining from text corpora. *Proceedings of the VLDB Endowment.*, 8(3):305-16, 2014

[8] Wang X, McCallum A and Wei X. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. *InSeventh IEEE International Conference on Data Mining (ICDM 2007)*, pp. 697-702, 2007.

[9] Ramirez PM, Mattmann CA. ACE: improving search engines via Automatic Concept Extraction. *InInformation Reuse and Integration, 2004. IRI 2004. Proceedings of the 2004 IEEE International Conference on* (2004), pp. 229-234

[10] Turney PD. Learning algorithms for keyphrase extraction. *Information Retrieval.*, 2(4):303-36, 2000.

[11] Parameswaran A, Garcia-Molina H and Rajaraman A. Towards the web of concepts: Extracting concepts from large datasets. *Proceedings of the VLDB Endowment.*, 3(1-2):566-77, 2010.

[12] Gelfand B, Wulfekuler M and Punch WF. Automated concept extraction from plain text. *In AAAI 1998 Workshop on Text Categorization.*, pp. 13-17, 1998.

[13] Rajagopal D, Cambria E, Olsher D and Kwok K. A graph-based approach to commonsense concept extraction and semantic similarity detection. *In Proceedings of the 22nd international conference on World Wide Web companion*, pp. 565-570, 2013.

[14] Krulwich B and Burkey C. Learning user information interests through extraction of semantically significant phrases. *InProceedings of the AAAI spring symposium on machine learning in information access*, pp. 100-112, 1996.

[15] Witten IH, Paynter GW, Frank E, Gutwin C and Nevill-Manning CG. KEA: Practical automatic keyphrase extraction. *InProceedings of the fourth ACM conference on Digital libraries.*, pp. 254-255, 1999.

[16] Song M, Song IY and Hu X. KPSpotter: a flexible information gain-based keyphrase extraction system. *InProceedings of the 5th ACM international workshop on Web information and data management.*, pp. 50-53, 2003.

[17] Frantzi K, Ananiadou S and Mima H. Automatic recognition of multi-word terms:. the c-value/nc-value method. *International Journal on Digital Libraries.*, 3(2):115-30, 2000.

[18] Hofmann T. Probabilistic latent semantic indexing. *InProceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 50-57, 1999.

[19] Blei DM, Ng AY and Jordan MI. Latent dirichlet allocation. *the Journal of machine Learning research.*, 3:993-1022, 2003.

[20] Dumais ST. Latent semantic analysis. *Annual review of information science and technology.*, 38(1):188-230, 2004.

[21] Hofmann T. Probabilistic latent semantic indexing. *InProceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 50-57, 1999.

[22] Blei DM, Ng AY and Jordan MI. Latent dirichlet allocation. *Journal of machine Learning research.*, 993-1022, 2003.

[23] Bird S. NLTK: the natural language toolkit. *InProceedings of the COLING/ACL on Interactive presentation sessions*, pp. 69-72, 2006.

[24] Lewis, D, Yang Y, Rose T and Li F. RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research*, 5:361-397, 2004.

[25] Greene D and Cunningham P. Practical solutions to the problem of diagonal dominance in kernel document clustering. *InProceedings of the 23rd international conference on Machine learning*, pp. 377-384, 2006.

[26] Kumar, N., Kumar, M., & Singh, M. (2016). Automated ontology generation from a plain text using statistical and NLP techniques. International Journal of System Assurance Engineering and Management, 7(1), 282-293.

[27] Oliveira, H., Lima, R., Lins, R. D., Freitas, F., Riss, M., & Simske, S. J. (2016). A Concept-Based Integer Linear Programming Approach for Single-Document Summarization. In Intelligent Systems (BRACIS), 2016 5th Brazilian Conference on (pp. 403-408). IEEE.

[28] Cui, C., Shen, J., Chen, Z., Wang, S.,& Ma, J. (2017). Learning to Rank Images for Complex Queries in Concept-based Search. Neurocomputing.

[29] Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In Proceedings of the 14th conference on Computational linguistics-Volume 2 (pp. 539-545). Association for Computational Linguistics.

[30] Wille, R. (1992). Concept lattices and conceptual knowledge systems. Computers& mathematics with applications, 23(6-9), 493-515.

[31] Kawaumra, T., Sekine, M., & Matsumura, K. (2017). Hyponym/Hypernym Detection in Science and Technology Thesauri from Bibliographic Datasets. In Semantic Computing (ICSC), 2017 IEEE 11th International Conference on (pp. 180-187). IEEE.

[32] Roller, S., & Erk, K. (2016). Relations such as hypernymy: Identifying and exploiting hearst patterns in distributional vectors for lexical entailment. arXiv preprint arXiv:1605.05433.

[33] Seitner, J., Bizer, C., Eckert, K., Faralli, S., Meusel, R., Paulheim, H., & Ponzetto, S. (2016). A large database of hypernymy relations extracted from the web. In Proceedings of the 10th edition of the Language Resources and Evaluation Conference, Portoroz, Slovenia.

[34] Brin, S. (1998). Extracting patterns and relations from the world wide web. In International Workshop on The World Wide Web and Databases (pp. 172-183). Springer Berlin Heidelberg.

[35] Agichtein, E., & Gravano, L. (2000). Snowball: Extracting relations from large plain-text collections. In Proceedings of the fifth ACM conference on Digital libraries (pp. 85-94). ACM.

[36] Yates, A., Cafarella, M., Banko, M., Etzioni, O., Broadhead, M., & Soderland, S. (2007). Textrunner: open information extraction on the web. In Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations (pp. 25-26). Association for Computational Linguistics.

[37] Sun, A. (2009). A Two-stage Bootstrapping Algorithm for Relation Extraction. In RANLP (pp. 76-82).

[38] Dligach, D., Miller, T., Lin, C., Bethard, S., & Savova, G. (2017). Neural Temporal Relation Extraction. EACL 2017, 746.

[39] Lin, Y., Shen, S., Liu, Z., Luan, H.,& Sun, M. (2016). Neural relation extraction with selective attention over instances. In Proceedings of ACL (Vol. 1, pp. 2124-2133).

[40] Sahu, S. K., Anand, A., Oruganty, K., & Gattu, M. (2016). Relation extraction from clinical texts using domain invariant convolutional neural network. arXiv preprint arXiv:1606.09370.

[41] Herawan, T., Deris, M. M., & Hamdan, A. R. (2017, February). FCA-ARMM: A Model for Mining Association Rules from Formal Concept Analysis. In Recent Advances on Soft Computing and Data Mining: The Second International Conference

on Soft Computing and Data Mining (SCDM-2016), Bandung, Indonesia, August 18th-20th, 2016 Proceedings (Vol. 549, p. 213). Springer.

[42] Negm, E., AbdelRahman, S., & Bahgat, R. (2017). PREFCA: A portal retrieval engine based on formal concept analysis. Information Processing & Management, 53(1), 203-222.

[43] Zhao, M., & Zhang, S. (2016). Identifying and Validating Ontology Mappings by Formal Concept Analysis. Ontology Matching, 61.

[44] González-Calabozo, J. M., Valverde-Albacete, F. J., & Peláez-Moreno, C. (2016). Interactive knowledge discovery and data mining on genomic expression data with numeric formal concept analysis. BMC bioinformatics, 17(1), 374.

[45] Wille, R. (1982). Restructuring lattice theory: an approach based on hierarchies of concepts. In Ordered sets (pp. 445-470). Springer Netherlands.

[46] Davey, B. A., & Priestley, H. A. (2002). Introduction to lattices and order. Cambridge university press.

[47] Anoop, V. S., Asharaf, S., & Deepak, P. (2016). Unsupervised Concept Hierarchy Learning: A Topic Modeling Guided Approach. Procedia Computer Science, 89, 386-394.

[48] Anoop, V. S., Asharaf, S., & Deepak, P. (2016). Learning Concept Hierarchies through Probabilistic Topic Modeling. International Journal of Information Processing, 10(3), 1 -11.

[49] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825-2830.

[50] Bogatyrev, M. (2016). Fact Extraction from Natural Language Texts with Conceptual Modeling. In International Conference on Data Analytics and Management in Data Intensive Domains (pp. 89-102). Springer, Cham.

[51] Seitner, J., Bizer, C., Eckert, K., Faralli, S., Meusel, R., Paulheim, H., & Ponzetto, S. (2016). A large database of hypernymy relations extracted from the web. In Proceedings of the 10th edition of the Language Resources and Evaluation Conference, Portoroz, Slovenia.

[52] Bartl, E., Rezankova, H., & Sobisek, L. (2011). Comparison of classical dimensionality reduction methods with novel approach based on formal concept analysis. In International Conference on Rough Sets and Knowledge Technology (pp. 26-35). Springer Berlin Heidelberg.

[53] Codocedo, V., Taramasco, C., & Astudillo, H. (2011). Cheating to achieve formal concept analysis over a large formal context. In The Eighth International Conference on Concept Lattices and their Applications-CLA 2011 (pp. 349-362).

[54] Kuznetsov, S. O., & Obiedkov, S. A. (2002). Comparing performance of algorithms for generating concept lattices. Journal of Experimental & Theoretical Artificial Intelligence, 14(2-3), 189-216.

[55] Outrata, J., & Vychodil, V. (2012). Fast algorithm for computing fixpoints of Galois connections induced by object-attribute relational data. Information Sciences, 185(1), 114-127.

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

[57] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).

[58] Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In Proceedings of the 31st International Conference on Machine Learning (ICML-14) (pp. 1188-1196).

[59] Song, Y., Wang, H., Wang, Z., Li, H., & Chen, W. (2011). Short text conceptualization using a probabilistic knowledgebase. In Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Vol. 3, (pp. 2330-2336). AAAI Press.

[60] Wu, W., Li, H., Wang, H., & Zhu, K. Q. (2012). Probase: A probabilistic taxonomy for text understanding. In Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data (pp. 481-492). ACM.

[61] Hua, W., Wang, Z., Wang, H., Zheng, K., & Zhou, X. (2015). Short text understanding through lexical semantic analysis. In IEEE 31st International Conference on Data Engineering (ICDE), (pp. 495-506). IEEE.

[62] Kim, H. K., Kim, H., & Cho, S. (2017). Bag-of-concepts: Comprehending document representation through clustering words in distributed representation. Neurocomputing, Vol. 266, (pp. 336-352), Elsevier.

[63] Baba, K., Nakatoh, T., & Minami, T. (2017). Vector representation of words for plagiarism detection based on string matching. In International Conference on Human Interface and the Management of Information, (pp. 341-350), Springer.

[64] Fleiss, J. L., & Cohen, J. (1973). The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. Educational and psychological measurement, 33(3), 613-619.

[65] Mrozinski, J., Whittaker, E., & Furui, S. (2008). Collecting a why-question corpus for development and evaluation of an automatic QA-system. In 46th Annual Meeting of the Association of Computational Linguistics: Human Language Technologies (pp. 443-451).

[66] Sarasua, C., Simperl, E., & Noy, N. F. (2012). Crowdmap: Crowdsourcing ontology alignment with microtasks. In International Semantic Web Conference (pp. 525-541).

[67] Bird, S. (2006, July). NLTK: the natural language toolkit. In Proceedings of the COLING/ACL on Interactive presentation sessions (pp. 69-72). Association for Computational Linguistics.

[68] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O. & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research, Vol. 12, (pp. 2825-2830).

[69] Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algo-rithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1), 100-108.

[70] Lu, C., Huang, H., Jian, P., Wang, D., & Guo, Y. D. (2017). A P-LSTM Neural Network for Sentiment Classification. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, (pp. 524-533), Springer.

[71] Sherkat, E., & Milios, E. E. (2017). Vector embedding of wikipedia concepts and entities. In International Conference on Applications of Natural Language to Information Systems (pp. 418-428). Springer, Cham.

[72] Hsu, S. T., Moon, C., Jones, P., & Samatova, N. F. (2017). A Hybrid CNN-RNN Alignment Model for Phrase-Aware Sentence Classification. EACL 2017, 443.

[73] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. The semantic web, 722-735.

[74] Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., & Hellmann, S. (2009). DBpedia-A crystallization point for the Web of Data. Web Semantics: science, services and agents on the world wide web, 7(3), 154-165.

[75] Miller, G. A. (1995). WordNet: a lexical database for English. Communications of the ACM, 38(11), 39-41.

[76] Song, Y., Wang, S., & Wang, H. (2015). Open Domain Short Text Conceptualization: A Generative + Descriptive Modeling Approach. In International Joint Conference on Artificial Intelligence, (pp. 3820-3826).

[77] Kim, D., Wang, H., & Oh, A. H. (2013). Context-Dependent Conceptualization. In IJCAI (pp. 2654-2661).

[78] Wang, Z., Zhao, K., Wang, H., Meng, X., & Wen, J. R. (2015). Query Understanding through Knowledge-Based Conceptualization. In IJCAI (pp. 3264-3270).

[79] Kohonen, T. (1998). The self-organizing map. Neurocomputing, 21(1), 1-6.

[80] Moon, T. K. (1996). The expectation-maximization algorithm. IEEE Signal processing magazine, 13(6), 47-60.

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

[82] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3, 993-1022.

[83] Hofmann, T. (1999). Probabilistic latent semantic indexing. *In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, (pp. 50-57). ACM.

[84] Hofmann, T. (1999). Probabilistic latent semantic analysis. In Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence (pp. 289-296). Morgan Kaufmann Publishers Inc..

[85] Chang, J., Boyd-Graber, J. L., Gerrish, S., Wang, C., & Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. *In NIPS*, 31, pp. 1-9

[86] Arvanitis, A., Wiley, M. T., & Hristidis, V. (2014). Efficient Concept-based Document Ranking. *In EDBT* (pp. 403-414).

[87] Egozi, O., Gabrilovich, E., & Markovitch, S. (2008). Concept-Based Feature Generation and Selection for Information Retrieval. *In AAAI* (pp. 1132-1137).

[88] Celikyilmaz, A., & Hakkani-Tür, D. (2011). Concept-based classification for multi-document summarization.

In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), (pp. 5540-5543).

[89] Cambria, E. (2013). An Introduction to Concept-Level Sentiment Analysis. *In MICAI*, 2, (pp. 478-483).

[90] Asharaf, S., Anoop, V. S., & Afzal, A. L. (2016). A Framework for Meaning Aware Product Discovery in E-Commerce. In Encyclopedia of E-Commerce Development, Implementation, and Management (pp. 1386-1398). IGI Global.

[91] Anoop, V. S., & Asharaf, S. (2017). A Topic Modeling Guided Approach for Semantic Knowledge Discovery in e-Commerce. International Journal of Interactive Multimedia and Artificial Intelligence, (pp. 1-8).

[92] Wang, X., McCallum, A., & Wei, X. (2007). Topical n-grams: Phrase and topic discovery, with an application to information retrieval. *In Proceedings of Seventh IEEE International Conference on Data Mining, (ICDM 2007)*, (pp. 697-702).

[93] Lindsey, R. V., Headden III, W. P., & Stipicevic, M. J. (2012). A phrase-discovering topic model using hierarchical pitman-yor processes. *In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (pp. 214-222).

[94] Jameel, S., & Lam, W. (2013). An unsupervised topic segmentation model incorporating word order. *In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval* (pp. 203-212).

[95] Yang, G., Wen, D., Chen, N. S., & Sutinen, E. (2015). A novel contextual topic model for multi-document summarization. *Expert Systems with Applications*, 42(3), 1340-1352.

[96] Sleeman, J., Finin, T., & Joshi, A. (2015). Topic Modeling for RDF Graphs. *In LD4IE@ ISWC* (pp. 48-62).

[97] Wallach, H. M. (2006). Topic modeling: beyond bag-of-words. *In Proceedings of the 23rd international conference on Machine learning* (pp. 977-984).

[98] Pitman, J., & Yor, M. (1997). The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, 855-900.

[99] El-Kishky, A., Song, Y., Wang, C., Voss, C. R., & Han, J. (2014). Scalable topical phrase mining from text corpora. *Proceedings of the VLDB Endowment*, 8(3), 305-316.

[100] He, Y. (2016). Extracting Topical Phrases from Clinical Documents. *In AAAI* (pp. 2957-2963).

[101] Chemudugunta, C., Smyth, P., & Steyvers, M. (2008). Combining concept hierarchies and statistical topic models. *In Proceedings of the 17th ACM conference on Information and knowledge management* (pp. 1469-1470).

[102] Anoop, V. S., Asharaf, S., & Deepak, P. (2016). Unsupervised Concept Hierarchy Learning: A Topic Modeling Guided Approach. Procedia Computer Science, 89, 386-394.

[103] Chemudugunta, C., Holloway, A., Smyth, P., & Steyvers, M. (2008). Modeling documents by combining semantic concepts with unsupervised statistical learning. *In International Semantic Web Conference* (pp. 229-244).

[104] Hofmann, T. (1999). Probabilistic latent semantic indexing. *In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 50-57).

[105] Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1), 5228-5235.

[106] Ramos, J. (2003). Using tf-idf to determine word relevance in document queries. *In Proceedings of the first instructional conference on machine learning.*

[107] Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., & Gatford, M. (1995). Okapi at TREC-3. *Nist Special Publication* Sp, 109, 109.

[108] Mrozinski, J., Whittaker, E., & Furui, S. (2008). Collecting a why-question corpus for development and evaluation of an automatic QA-system. *In 46th Annual Meeting of the Association of Computational Linguistics: Human Language Technologies* (pp. 443-451).

[109] Sarasua, C., Simperl, E., & Noy, N. F. (2012). Crowdmap: Crowdsourcing ontology alignment with microtasks. *In International Semantic Web Conference* (pp. 525-541).

[110] Loria, S. (2014). TextBlob: simplified text processing.

[111] Shan, D., Zhao, W. X., He, J., Yan, R., Yan, H., & Li, X. (2011). Efficient phrase querying with flat position index. *In Proceedings of the 20th ACM international conference on Information and knowledge management* (pp. 2001-2004).

[112] Patil, M., Thankachan, S. V., Shah, R., Hon, W. K., Vitter, J. S., & Chandrasekaran, S. (2011). Inverted indexes for phrases and strings. *In Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval* (pp. 555-564).

[113] Fleiss, J. L., & Cohen, J. (1973). The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement*, 33(3), 613-619.

[114] Newman, D., Lau, J. H., Grieser, K., & Baldwin, T. (2010). Automatic evaluation of topic coherence. *In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 100-108).

[115] Li, B., Wang, B., Zhou, R., Yang, X., & Liu, C. (2016). CITPM: A Cluster-Based Iterative Topical Phrase Mining Framework. *In International Conference on Database Systems for Advanced Applications* (pp. 197-213).

[116] Li, X., & Jin, W. (2016). Cross-Document Knowledge Discovery Using Semantic Concept Topic Model. *In 15th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 108-114).

[117] Xu, K., Qi, G., Huang, J., & Wu, T. (2017). Incorporating Wikipedia concepts and categories as prior knowledge into topic models. *Intelligent Data Analysis*, 21(2), 443-461.

[118] Shen, Aaron. Aspect-Specific Ranking of Product Reviews Using Topic Modeling.

[119] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *the Journal of machine Learning research* 3 (2003): 993-1022.

[120] Colorado Reed, Latent Dirichlet Allocation: Towards a Deeper Understanding. January 2012.

[121] Yu, Hong, and Vasileios Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. *In Proceedings of the conference on Empirical methods in natural language processing*, pp. 129-136. Association for Computational Linguistics, 2003.

[122] Neviarouskaya, Alena, Helmut Prendinger, and Mitsuru Ishizuka. Affect analysis model: novel rule-based approach to affect sensing from text. *Natural Language Engineering* 17, no. 01 (2011): 95-135.

[123] Li, Hao, Yu Chen, Heng Ji, Smaranda Muresan, and Dequan Zheng. Combining social cognitive theories with linguistic features for multi-genre sentiment analysis. *In Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation*. pp. 127-136. 2012.

[124] Kim, Suin, Jianwen Zhang, Zheng Chen, Alice H. Oh, and Shixia Liu. A Hierarchical Aspect-Sentiment Model for Online Reviews *In AAAI*. 2013.

[125] Poria, Soujanya, Alexander Gelbukh, B. Agarwal, E. Cambria, and N. Howard. Sentic Demo: A hybrid concept-level aspect-based sentiment analysis toolkit. *ESWC 2014* (2014).

[126] Malouf, Robert, and Tony Mullen. Taking sides: User classification for informal online political discourse. *Internet Research* 18, no. 2 (2008): 177-190.

[127] Li, Hao, Yu Chen, Heng Ji, Smaranda Muresan, and Dequan Zheng. Combining social cognitive theories with linguistic features for multi-genre sentiment analysis. *In Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation*. pp. 127-136. 2012.

[128] Kim, Suin, Jianwen Zhang, Zheng Chen, Alice H. Oh, and Shixia Liu. A Hierarchical Aspect-Sentiment Model for Online Reviews *In AAAI*. 2013.

[129] Maas, Andrew L., Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. *InProceedings*

of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pp. 142-150. Association for Computational Linguistics, 2011.

[130] Bespalov, Dmitriy, Bing Bai, Yanjun Qi, and Ali Shokoufandeh. Sentiment classification based on supervised latent n-gram analysis. *In Proceedings of the 20th ACM international conference on Information and knowledge management*, pp. 375-382. ACM, 2011.

[131] Huang, Jin, Jingjing Lu, and Charles X. Ling. Comparing naive Bayes, decision trees, and SVM with AUC and accuracy. *Data Mining*, 2003. *ICDM* 2003. *Third IEEE International Conference on. IEEE*, 2003.

[132] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. *In Proceedings of the ACL-02 conference on Empirical methods in natural language processing*-Volume 10, pp. 79-86. Association for Computational Linguistics, 2002.

[133] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. *In Proceedings of the ACL-02 conference on Empirical methods in natural language processing*-Volume 10, pp. 79-86. Association for Computational Linguistics, 2002.

[134] Paltoglou, Georgios, and Mike Thelwall. A study of information retrieval weighting schemes for sentiment analysis.*In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 1386-1395. Association for Computational Linguistics, 2010.

[135] Ding, Wanying. Similarity Dependency Dirichlet Process for Aspect-Based Sentiment Analysis. Wanying Ding Xiaoli Song Yue Shang Junhuan Zhu Lifan Guo Xiaohua Hu.

[136] Deerwester, Scott C., Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *JAsIs* 41, no. 6 (1990): 391-407.

[137] Hofmann, Thomas. Probabilistic latent semantic analysis. *In Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pp. 289-296. Morgan Kaufmann Publishers Inc., 1999.

[138] Bing Liu. Sentiment Analysis and Opinion Mining. Morgan & Claypool publishers, 2012

[139] Li, Tao, Yi Zhang, and Vikas Sindhwani. A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*:Volume 1. Association for Computational Linguistics, 2009.

[140] McAuley, Julian, and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 2013.

[141] McCallum, Andrew K. MALLET: A Machine Learning for Language Toolkit (2002).

[142] Shelke, N., Deshpande, S., & Thakare, V. (2017). Domain independent approach for aspect oriented sentiment analysis for product reviews. In Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications (pp. 651-659). Springer, Singapore.

[143] Li, J. J., Yang, H., & Tang, H. (2017). Feature Mining and Sentiment Orientation Analysis on Product. In Management Information and Optoelectronic Engineering: Proceedings of the 2016 International Conference on Management, Information and Communication (ICMIC2016) and the 2016 International Conference on Optics and Electronics Engineering (ICOEE2016) (p. 79). World Scientific.

[144] Akhtar, M. S., Gupta, D., Ekbal, A., & Bhattacharyya, P. (2017). Feature selection and ensemble construction: A two-step method for aspect based sentiment analysis. Knowledge-Based Systems, 125, 116-135.

[145] Piryani, R., Gupta, V., Singh, V. K., & Ghose, U. (2017). A Linguistic Rule-Based Approach for Aspect-Level Sentiment Analysis of Movie Reviews. In Advances in Computer and Computational Sciences (pp. 201-209). Springer, Singapore.

[146] Salas-Zárate, M. D. P., Medina-Moreira, J., Lagos-Ortiz, K., Luna-Aveiga, H., Rodríguez-García, M. Á., & Valencia-García, R. (2017). Sentiment Analysis on Tweets about Diabetes: An Aspect-Level Approach. Computational and mathematical methods in medicine, 2017.

[147] Yadollahi, A., Shahraki, A. G., & Zaiane, O. R. (2017). Current State of Text Sentiment Analysis from Opinion to Emotion Mining. ACM Computing Surveys (CSUR), 50(2), 25.

[148] Schouten, K., van der Weijde, O., Frasincar, F., & Dekker, R. (2017). Supervised and Unsupervised Aspect Category Detection for Sentiment Analysis With Co-Occurrence Data. IEEE Transactions on Cybernetics.

[149] Wang, Y., Huang, Y., & Wang, M. (2017). Aspect-Based Rating Prediction on Reviews Using Sentiment Strength Analysis. In International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (pp. 439-447). Springer, Cham.

[150] Akhtar, M. S., Kohail, S., Kumar, A., Ekbal, A., & Biemann, C. (2017). Feature Selection Using Multi-objective Optimization for Aspect Based Sentiment Analysis. In International Conference on Applications of Natural Language to Information Systems (pp. 15-27). Springer, Cham.

[151] Qazi, A., Qazi, A., Raj, R. G., Raj, R. G., Hardaker, G., Hardaker, G., & Standing, C. (2017). A systematic literature review on opinion types and sentiment analysis techniques: Tasks and challenges. Internet Research, 27(3), 608-630.

[152] Marstawi, A., Sharef, N. M., Aris, T. N. M., & Mustapha, A. (2017). Ontology-based Aspect Extraction for an Improved Sentiment Analysis in Summarization of Product Reviews. In Proceedings of the 8th International Conference on Computer Modeling and Simulation (pp. 100-104). ACM.

[153] Brambilla, M., Ceri, S., Della Valle, E., Volonterio, R., & Acero Salazar, F. X. (2017). Extracting emerging knowledge from social media. In Proceedings of the 26th International Conference on World Wide Web (pp. 795-804). International World Wide Web Conferences Steering Committee.

[154] Oostdijk, N., Hürriyetoglu, A., Puts, M., Daas, P., & van den Bosch, A. (2016). Information extraction from social media: A linguistically motivated approach. PARIS Inalco du 4 au 8 juillet 2016, 23.

[155] Shen, S., Murzintcev, N., Song, C., & Cheng, C. (2017). Information retrieval of a disaster event from cross-platform social media. Information Discovery and Delivery, (just-accepted), 00-00.

[156] Wang, Z., & Zhang, Y. (2016). A Text Information Retrieval Method by Integrating Global and Local Textual Information. In Computer Software and Applications Conference (COMPSAC), 2016 IEEE 40th Annual (Vol. 1, pp. 504-505). IEEE.

[157] Wang, S., Chen, Z., Liu, B., & Emery, S. (2016). Identifying Search Keywords for Finding Relevant Social Media Posts. In AAAI (pp. 3052-3058).

[158] Gao, L., Wang, Y., Li, D., Shao, J., & Song, J. (2017). Real-time social media retrieval with spatial, temporal and social constraints. Neurocomputing, 253, 77-88.

[159] Tolosa, G., Feuerstein, E., Becchetti, L., & Marchetti-Spaccamela, A. (2017). Performance improvements for search systems using an integrated cache of lists+ intersections lists+ intersections . Information Retrieval Journal. doi:10.1007/s10791-017-9299-5.

[160] Kim, Y., Callan, J., Culpepper, J. S., & Moffat, A. (2017). Efficient distributed selective search. Information Retrieval Journal. doi:10.1007/s10791-016-9290-6.

[161] Gagie, T., Hartikainen, A., Karhu, K., Kärkkäinen, J., Navarro, G., Puglisi, S. J., & Sirén, J. (2017). Document retrieval on repetitive string collections. Information Retrieval Journal. doi:10.1007/s10791-017-9297-7.

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

[163] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).

[164] Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In Proceedings of the 31st International Conference on Machine Learning (ICML-14) (pp. 1188-1196).

[165] Fleiss, J. L., & Cohen, J. (1973). The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. Educational and psychological measurement, 33(3), 613-619.

[166] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. Journal of machine Learning research, 3(Jan), 993-1022.

[167] Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., & Gatford, M. (1995). Okapi at TREC-3. Nist Special Publication Sp, 109, 109.

[168] Bird, S. (2006). NLTK: the natural language toolkit. In Proceedings of the COLING/ACL on Interactive presentation sessions (pp. 69-72). Association for Computational Linguistics.

[169] Shan, D., Zhao, W. X., He, J., Yan, R., Yan, H., & Li, X. (2011). Efficient phrase querying with flat position index. In Proceedings of the 20th ACM international conference on Information and knowledge management (pp. 2001-2004). ACM.

[170] Patil, M., Thankachan, S. V., Shah, R., Hon, W. K., Vitter, J. S., & Chandrasekaran, S. (2011). Inverted indexes for phrases and strings. In Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval (pp. 555-564). ACM.

[171] Raviv, H., Kurland, O., & Carmel, D. (2016). Document retrieval using entity-based language models. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval (pp. 65-74). ACM.

[172] Yamamura, E. (2016). Natural disasters and social capital formation: The impact of the Great Hanshin-Awaji earthquake. Papers in Regional Science, 95(S1).

[173] Halberstam, Y., & Knight, B. (2016). Homophily, group size, and the diffusion of political information in social networks: Evidence from Twitter. Journal of Public Economics, 143, 73-88.

[174] Nakov, P., Rosenthal, S., Kiritchenko, S., Mohammad, S. M., Kozareva, Z., Ritter, A., & Zhu, X. (2016). Developing a successful SemEval task in sentiment analysis of Twitter and other social media texts. Language Resources and Evaluation, 50(1), 35-65.

[175] Anoop, V. S., Asharaf, S., & Alessandro, Z. (2015). Generating and visualizing topic hierarchies from microblogs: An iterative latent dirichlet allocation approach. In Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on (pp. 824-828). IEEE.

[176] Sekara, V., Stopczynski, A., & Lehmann, S. (2016). Fundamental structures of dynamic social networks. Proceedings of the national academy of sciences, 113(36), 9977-9982.

[177] Aggarwal, C. C. (2011). An introduction to social network data analytics. Social network data analytics, 1-15.

[178] Boughanem, M. (2013). Information retrieval and social media. In Modeling Approaches and Algorithms for Advanced Computer Applications (pp. 7-7). Springer International Publishing.

[179] Wallach, H. M. (2006). Topic modeling: beyond bag-of-words. In Proceedings of the 23rd international conference on Machine learning (pp. 977-984). ACM.

[180] Hong, L.,& Davison, B. D. (2010). Empirical study of topic modeling in twitter. In Proceedings of the first workshop on social media analytics (pp. 80-88). ACM.

[181] Zhao, W. X., Jiang, J., Weng, J., He, J., Lim, E. P., Yan, H., & Li, X. (2011). Comparing twitter and traditional media using topic models. In European Conference on Information Retrieval (pp. 338-349). Springer, Berlin, Heidelberg.

[182] Lim, K. W., Chen, C., & Buntine, W. (2016). Twitter-network topic model: A full Bayesian treatment for social network and text modeling. arXiv preprint arXiv:1609.06791.

[183] Steinskog, A. O., Therkelsen, J. F., & Gambäck, B. (2017). Twitter Topic Modeling by Tweet Aggregation. In Proceedings of the 21st Nordic Conference on Computational Linguistics, NoDaLiDa, 22-24 May 2017, Gothenburg, Sweden (No. 131, pp. 77-86). Linköping University Electronic Press.

[184] Guha, R., McCool, R., & Miller, E. (2003, May). Semantic search. In Proceedings of the 12th international conference on World Wide Web (pp. 700-709). ACM.

[185] Lei, Y., Uren, V., & Motta, E. (2006, October). Semsearch: A search engine for the semantic web. In International Conference on Knowledge Engineering and Knowledge Management (pp. 238-245). Springer, Berlin, Heidelberg.

[186] Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R. S., Peng, Y., & Sachs, J. (2004, November). Swoogle: a search and metadata engine for the semantic web. In Proceedings of the thirteenth ACM international conference on Information and knowledge management (pp. 652-659). ACM.

[187] Fu, Z., Huang, F., Sun, X., Vasilakos, A., & Yang, C. N. (2016). Enabling semantic search based on conceptual graphs over encrypted outsourced data. IEEE Transactions on Services Computing.

[188] Robertson, S. (2019). A Brief History of Search Results Ranking. IEEE Annals of the History of Computing.

[189] Chatterjee, N., & Yadav, N. (2019). Hybrid Latent Semantic Analysis and Random Indexing Model for Text Summarization. In Information and Communication Technology for Competitive Strategies (pp. 149-156). Springer, Singapore.

[190] Bullock, B. N., Hotho, A., & Stumme, G. (2018). Accessing Information with Tags: Search and Ranking. In Social Information Access (pp. 310-343). Springer, Cham.

[191] Hao, S., Shi, C., Niu, Z., & Cao, L. (2018). Concept coupling learning for improving concept lattice-based document retrieval. Engineering Applications of Artificial Intelligence, 69, 65-75.

[192] Zou, F., Tang, X., Li, K., Wang, Y., Song, J., Yang, S., & Ling, H. (2018). Hidden semantic hashing for fast retrieval over large scale document collection. Multimedia Tools and Applications, 77(3), 3677-3697.

[193] Dragoni, M., Poria, S., & Cambria, E. (2018). OntoSenticNet: A commonsense ontology for sentiment analysis. IEEE Intelligent Systems, 33(3), 77-85.

[194] Majumder, N., Hazarika, D., Gelbukh, A., Cambria, E., & Poria, S. (2018). Multimodal sentiment analysis using hierarchical fusion with context modeling. Knowledge-Based Systems, 161, 124-133.

[195] Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., Mohammad, A. S., & Hoste, V. (2016). Semeval-2016 task 5: Aspect based sentiment analysis. In Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016) (pp. 19-30).

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

[197] Yih, S. W. T., Chang, M. W., He, X., & Gao, J. (2015). Semantic parsing via staged query graph generation: Question answering with knowledge base.

[198] Yih, W. T., Richardson, M., Meek, C., Chang, M. W., & Suh, J. (2016). The value of semantic parse labeling for knowledge base question answering. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (Vol. 2, pp. 201-206).

[199] Huang, X., Zhang, J., Li, D., & Li, P. (2019, January). Knowledge graph embedding based question answering. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (pp. 105-113). ACM.

[200] Zheng, W., Cheng, H., Yu, J. X., Zou, L., & Zhao, K. (2019). Interactive natural language question answering over knowledge graphs. Information Sciences, 481, 141-159.

[201] Sun, H., Dhingra, B., Zaheer, M., Mazaitis, K., Salakhutdinov, R., & Cohen, W. W. (2018). Open domain question answering using early fusion of knowledge bases and text. arXiv preprint arXiv:1809.00782.

[202] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. Journal of the American society for information science, 41(6), 391-407.

[203] Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. Discourse processes, 25(2-3), 259-284.

[204] Hofmann, T. (1999, July). Probabilistic latent semantic analysis. In Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence (pp. 289-296). Morgan Kaufmann Publishers Inc..

[205] Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. Machine learning, 42(1-2), 177-196.

[206] Zobel, J., & Moffat, A. (2006). Inverted files for text search engines. *ACM computing surveys (CSUR)*, 38(2), 6.

[207] Bahle, D., Williams, H. E., & Zobel, J. (2001). Compaction Techniques for Nextword Indexes. **In spire** (pp. 33-45).

[208] Bahle, D., Williams, H. E., & Zobel, J. (2001). Optimised phrase querying and browsing of large text databases. *Australian Computer Science Communications*, 23(1), 11-19.

[209] Williams, H. E., Zobel, J., & Bahle, D. (2004). Fast phrase querying with combined indexes. *ACM Transactions on Information Systems (TOIS)*, 22(4), 573-594.

[210] Loper, E., & Bird, S. (2002). NLTK: the natural language toolkit. arXiv preprint cs/0205028.

[211] Xu, Y., Rajpathak, D., Gibbs, I., & Klabjan, D. (2019). Automatic Ontology Learning from Domain-Specific Short Unstructured Text Data. arXiv preprint arXiv:1903.04360.

[212] Khovrichev, M. A., Chernykh, I. A., Mamaev, N. K., & Matveev, Y. N. (2019). Context-Dependent Synonym and Concept Extraction for Dialogue Systems Training. In 2019 International Conference" Quality Management, Transport and Information Security, Information Technologies"(IT&QM&IS) (pp. 421-423). IEEE.

[213] Li, H., Armiento, R., & Lambrix, P. (2019). Extending Ontologies in the Nanotechnology Domain using Topic Models and Formal Topical Concept Analysis on Unstructured Text. In 18th International Semantic Web Conference (pp. 5-8). CEUR Workshop Proceedings.

[214] Le, M., Roller, S., Papaxanthos, L., Kiela, D., & Nickel, M. (2019). Inferring concept hierarchies from text corpora via hyperbolic embeddings. arXiv preprint arXiv:1902.00913.

[215] Li, B., Xu, W., Tian, Y., & Chen, J. (2019,). A Phrase Topic Model for Large-scale Corpus. In 2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA) (pp. 634-639). IEEE.

[216] Kovalchuk, P., Proença, D., Borbinha, J., & Henriques, R. (2019). An unsupervised method for concept association analysis in text collections. In International Conference on Theory and Practice of Digital Libraries (pp. 18-32). Springer, Cham.

List of Publications

Peer Reviewed International Journals

- Anoop, V. S., & Asharaf, S. (2019). Aspect-Oriented Sentiment Analysis: A Topic Modeling-Powered Approach. Journal of Intelligent Systems, 29(1), 1166-1178. [*Indexed in Web of Science: Emerging Sources Citation Index (ESCI) Thomson Reuters, DBLP, Compendex, Scopus, SCImago, UGC Approved Journal No. 30264*]

- Anoop, V. S., & Asharaf, S. (2019). Conceptualized phrase clustering with distributed k-means. Intelligent Decision Technologies, 13(2), 153-160. [*Indexed in Web of Science: Emerging Sources Citation Index (ESCI) Thomson Reuters, DBLP, Compendex, Scopus, UGC Approved Journal No. 21318*]

- Anoop, V. S., & Asharaf, S. (2019). Extracting Conceptual Relationships and Inducing Concept Lattices from Unstructured Text. Journal of Intelligent Systems, 28(4), 669-681. [*Indexed in Web of Science: Emerging Sources Citation Index (ESCI) Thomson Reuters, DBLP, Compendex, Scopus, SCImago, UGC Approved Journal No. 30264*]

- Anoop, V. S., & Asharaf, S. (2017). A Topic Modeling Guided Approach for Semantic Knowledge Discovery in e-Commerce. International Journal of Interactive Multimedia & Artificial Intelligence, 4(6). [*Web of Science: Science Citation Index Expanded (SCIE) Thomson Reuters, DBLP*]

- Anoop, V. S., Asharaf, S., & Deepak, P. (2016). Learning concept hierarchies through probabilistic topic modeling. International Journal of Information Processing, I K International Publishing, ISSN : 0973-8215 [*UGC Approved Journal No. 22945*]

- "A Distributional Semantics based Information Retrieval Framework for Online Social Networks", Intelligent Decision Technologies, IOS Press, Netherlands, ISSN:

24056456, E-ISSN: 24056464 [*Indexed in Web of Science: Emerging Sources Citation Index (ESCI) Thomson Reuters, DBLP, Compendex, Scopus, UGC Approved Journal No. 32871*] - Under Peer Review

Peer Reviewed International Conferences

- Anoop, V. S., & Asharaf, S. (2017). Distributional Semantic Phrase Clustering and Conceptualization Using Probabilistic Knowledgebase. In International Conference on Next Generation Computing Technologies (pp. 526-534). Springer, Singapore. [*Indexed in Scopus, DBLP, SCImago, UGC Approved Journal No. 16246*]

- Anoop, V. S., Asharaf, S., & Deepak, P. (2017). Topic Modeling for Unsupervised Concept Extraction and Document Ranking. In The International Symposium on Intelligent Systems Technologies and Applications (pp. 123-135). Springer, Cham. [*Indexed in Scopus, DBLP, SCImago, UGC Approved Journal No. 49365*]

- Anoop, V. S., Asharaf, S., & Deepak, P. (2016). Unsupervised concept hierarchy learning: a topic modeling guided approach. Procedia Computer Science, 89, 386-394. [*Scopus Indexed, UGC Approved Journal No. 46138*]

- Anoop, V. S., Asharaf, S., & Alessandro, Z. (2015). Generating and visualizing topic hierarchies from microblogs: An iterative latent dirichlet allocation approach. In 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 824-828). IEEE. [*Indexed in Scopus*]

V.S. Anoop* and S. Asharaf

Extracting Conceptual Relationships and Inducing Concept Lattices from Unstructured Text

https://doi.org/10.1515/jisys-2017-0225
Received May 16, 2017.

Abstract: Concept and relationship extraction from unstructured text data plays a key role in meaning aware computing paradigms, which make computers intelligent by helping them learn, interpret, and synthesis information. These concepts and relationships leverage knowledge in the form of ontological structures, which is the backbone of semantic web. This paper proposes a framework that extracts concepts and relationships from unstructured text data and then learns lattices that connect concepts and relationships. The proposed framework uses an off-the-shelf tool for identifying common concepts from a plain text corpus and then implements machine learning algorithms for classifying common relations that connect those concepts. Formal concept analysis is then used for generating concept lattices, which is a proven and principled method of creating formal ontologies that aid machines to learn things. A rigorous and structured experimental evaluation of the proposed method on real-world datasets has been conducted. The results show that the newly proposed framework outperforms state-of-the-art approaches in concept extraction and lattice generation.

Keywords: Formal concept analysis, concept extraction, concept lattices, relation extraction, knowledge discovery.

1 Introduction

Text is considered to be one form of data that is very rapidly generating because of the number of text-producing and text-consuming applications. User applications and platforms such as online social networks, digital libraries, e-commerce websites, and blogs generate text data, and this caused the creation of large unstructured text archives in organizations. These repositories are gold mines for organizations, as they contain invaluable patterns that help them leverage knowledge that can be used as an input to strategic but intelligent decision-making process. As the complexity and quantity of text data being generated grows exponentially, the need for more intelligent, scalable, and text-understanding algorithms is indispensable. The advent of semantic web, an extension and a meaning aware version of current World Wide Web, leads way to the introduction of numerous tools and techniques for leveraging, organizing, and presenting knowledge. Ontologies are building blocks of any meaning aware or semantic computing paradigm that comprises a set of concepts and their hierarchies and relationships in a domain of interest. Thus, the automated concept hierarchy learning from unstructured text has gained significant attention among text mining and natural language processing (NLP) researchers and practitioners. Concept hierarchy learning algorithms extract concepts from text and connect those concepts using potential relations that exist among them. Such hierarchies may find useful applications in concept-based ontology generation [19], concept-guided document summarization [24], and concept-guided information retrieval [10] to name a few.

***Corresponding author: V.S. Anoop,** Data Engineering Lab, Indian Institute of Information Technology and Management-Kerala (IIITM-K), Thiruvananthapuram, India, e-mail: anoop.res15@iiitmk.ac.in
S. Asharaf: Indian Institute of Information Technology and Management-Kerala (IIITM-K), Thiruvananthapuram, India

Authenticated | anoop.res15@iiitmk.ac.in author's copy
Download Date | 12/31/18 11:53 AM

1.1 Contributions

This work proposes a framework for identifying commonly occurring relations that connect concepts in unstructured text data and then learn them using machine learning techniques. Specifically, the proposed approach can identify and learn subsumption ("is-a"), Hearst patterns [14] ("such as", "or other", "and other", "including", "especially", etc.), and other potential indications of relations among concepts. This approach makes use of formal concept analysis (FCA) [34], which is a well-established mathematical theory of analyzing data, to form context table and concept lattices [34] from the identified concept and relations. These lattices can then be used for generating ontologies that may be used by intelligent and meaning aware computing systems. The authors compared the performance of this proposed system to some state-of-the-art methods through rigorous experiment, and results indicate that this approach outperforms the chosen baselines.

1.2 Organization

The rest of this paper is organized as follows. Section 2 discusses some of the very recent state-of-the-art methods in relation extraction (RE) and knowledge representation using FCA. The research objective and a formal problem definition are given in Sections 3 and 4, respectively. The new method proposed in this paper is explained in Section 5, and our experimental setup is given in Section 6. A detailed evaluation of results is given in Section 7, and we draw conclusions and then discuss future work in Section 8.

2 Related Work

RE is a subtask information extraction (IE) that aims at extracting relevant and potentially useful patterns or information from humongous data that are being generated day by day. The sheer volume and heterogeneity of data makes it difficult to analyze and extract these patterns manually. Thus, we need automated techniques for this process. NLP is one of the major areas that address this issue by scanning natural language texts and extract useful patterns. IE tasks, specifically the RE, have a long history and go back to late 1970s. However, successful commercial systems were introduced in the 1990s. In this section, we discuss some of the recent approaches in RE. In addition, we also throw light on state-of-the-art methods in FCA-based concept lattice generation approaches.

Informally, we can group all the RE approaches introduced in the literature into five categories: hand-built patterns, bootstrapping methods, supervised methods, distant supervision, and unsupervised approaches. Hand-built patterns uses handcrafted rules for extracting potentially relevant relation words from text, and one very notable work was introduced by M.A. Hearst known as Hearst patterns [14]. One issue with this approach is that it is difficult to write all sets of possible rules, and for other tasks such as meronym extraction, the set of rules will be different. But still, a good number of extensions are being reported that use Hearst patterns as their foundation for RE [16, 27, 29].

Another category of RE is the bootstrapping-based approach in which a specific set of seed relation instances are created and these are used for searching for new tuples. One such approach for extracting "author book" relation was the DIPRE [8]. Later, another system was introduced that uses the idea of bootstrapping; Snowball [1] extracts "organization-location" relation pairs. The limitations with the above-said algorithms are the specific relations they can only deal with. Users have to specify the type of relation they need to work with such as "author-book" or "organization-location".

Later, TextRunner [36] was introduced in the domain of RE, which can learn relations, classes, and entities from a corpus in a self-supervised manner. This approach first tags the training data as positive and negative and then trains a classifier on the data to generate potential relations and entities. Another two-stage

Authenticated | anoop.res15@iiitmk.ac.in author's copy
Download Date | 12/31/18 11:53 AM

bootstrapping algorithm [33] was proposed by Sun. In the first step, the algorithm uses a bootstrapping method to scan the tuples, and in the second stage, it learns relation nominals and contexts.

More recently, supervised and semisupervised approaches are found to be promising, and a very good number of literatures have been reported in the area of RE that uses deep learning techniques for identifying relation patterns. Very recently, a neural temporal RE approach [12] has been introduced in which the authors experimented with neural architectures for temporal RE. They showed that neural models that take tokens only outperform state-of-the-art hand-engineered feature-based models in performance. They also reported that encoding relation arguments with XML tags performs superior than a traditional position-based encoding. Another notable approach was introduced that attempts neural RE with selective attention over instances [21]. This work employs convolutional neural networks (CNN) to embed the semantics of sentences. Experimental results show that this model could make full use of all informative sentences and achieved significant and consistent improvement on RE task.

An approach for extracting relationships from clinical text was introduced [28] that exploited CNN to learn features automatically that reduces the dependency on manual feature engineering. They showed that CNN can be effectively used for RE in clinical text without being dependent on expert knowledge on feature engineering. Our proposed RE method uses machine learning techniques to classify Hearst patterns [14] ("such as", "or other", "and other", "including", "especially", etc.) and other potential indications of relations such as "is-a" among textual concepts.

In recent years, FCA [34] has attained significant interest from research communities of various domains. FCA can analyze data that describe relationships that exist between a particular set of objects and their attributes. FCA is being widely used as a knowledge representation framework, especially in knowledge engineering and ontology generation tasks in information science. This proposed work also uses FCA to create concept lattices that incorporate a set of concepts and the relationships that connect them. Here, we discuss some of the very recent works on FCA, which use concept lattices for knowledge representation and ontology generation.

One of the recent notable works on extending FCA for association rule mining for knowledge representation is FCA-ARMM [15]. The authors integrated FCA and association rule mining model (ARMM) and developed a tool called FCA Miner, which is capable of generating association rules from real datasets. A portal retrieval engine based on FCA (PREFCA) [23] was introduced in which a portal's semantic data were collected and formed in a concept lattice. Later in the information retrieval phase, a ranking has been done to retrieve the best result. Another work on identifying and validating ontology mappings using FCA was reported very recently [37]. The authors proposed a method called FCA-Map, which constructs formal contexts and then extracts mappings from the lattices derived. Then, a relation-based formal context is built and used for discovering additional structure mappings. An interactive knowledge discovery and data mining on genomic data using FCA [13] was introduced recently. They used FCA-based biclustering methods to index external databases for observing the evolution of genes throughout the different biclusters.

Very recently, an approach was proposed by Monnin et al. [22] which builds an optimal lattice-based structure for classifying RDF resources with respect to their predicates. The authors introduced the notion of lattice annotation that enabled to compare their classification to an ontology schema for confirming axioms that exhibit the subsumption relation or for suggesting a completely new one. The authors used the DBpedia dataset for their experiments and the results showed that their proposed approach could strongly demonstrate the ability of FCA to guide a possible structuring of Lined Open Data [22].

An approach for concept lattice reduction using fuzzy k-means clustering was introduced by Kumar and Srinivas [17]. The authors took into consideration the complexity in computing all the concepts from a large incident matrix and used a fuzzy k-means clustering for reducing the size of concept lattices. They also showcased the usefulness of their proposed method on two real-world applications such as information retrieval and information visualization. This method performed well on large context tables and the authors could represent reduced concept lattices efficiently [17]. A fuzzy clustering-based FCA for association rules mining was proposed by Kumar [18]. The author has done an association rule mining on a reduced formal context using fuzzy k-means clustering approach introduced in their previous work [17]. The authors conducted

Authenticated | anoop.res15@iiitmk.ac.in author's copy
Download Date | 12/31/18 11:53 AM

experiments on two real-world healthcare datasets and showed that better association rule mining is possible on a reduced concept lattice [18].

Zhao and Zhang [37] introduced a novel method for identifying and validating ontology mappings using FCA in which the authors constructed three types of formal contexts and extracted mappings from the lattices derived. First, they showed that class names, labels, and synonyms share lexical tokens that may lead to lexical mappings across ontologies. Then, they proposed how this lattice can lead to validate the lexical matching to either positive or negative based on the lexical anchors. In the third phase, they showed how we can discover additional structural mappings from the positive relation-based context [37]. The authors conducted experiments and evaluated their methods on anatomy and large biomedical ontologies track of OAEI 2015 [37].

A very comprehensive survey on FCA and its research trends and applications was reported in the literature, which was compiled by Singh et al. [32]. The work is a torchbearer for researchers who wish to work on FCA and related areas. The authors summarized more than 350 recent research papers published after 2011 and indexed in major reputed indexing services. They specifically provided the mathematical foundations of each extension of FCA such as FCA with granular computing, intervalued FCA, and possibility theory [32]. Semenova and Smirnov [31] recently published a paper on building formal ontologies from incomplete data. They presented new models and methods for ontological data analysis, which facilitates the identification of conceptual structures or formal ontologies of a particular knowledge domain. They proposed an intelligent analysis of the incomplete data for building conceptual structures using FCA [31].

In this work, we make use of FCA for building context tables depicting various concepts and their associated relations and then transform this context into concept lattice. We show that efficient knowledge representation is thus possible and this may be extended into ontology engineering task.

2.1 Background: FCA

FCA is defined as a mathematical model or framework based on lattice theory [35], which is well suited for knowledge engineering and processing tasks. In recent years, the complexity and amount of data being produced across organizations has grown exponentially, and practitioners and researchers use FCA as an intelligent data analysis tool. In a basic setting, FCA generates two outputs for any given context table. The first one is called a concept lattice and the second one is called attribute implications. The former, the concept lattice, is a partially ordered collection of objects and its attributes and the latter, the attribute implications, describes particular attribute dependencies that are true in the context table [6]. One useful feature of FCA worth mentioning is that we can perform reasoning with dependencies in data, reasoning with concepts in data, and visualization of data with concepts and relationships. Some common examples are hierarchical arrangements of web search results, gene expression data analysis, analysis of organization of annotated taxonomies, etc. [6].

Definition 1: Formal context: In FCA, a formal context can be defined as a triplet $<X, Y, R>$, where X and Y are nonempty sets and R is a binary relation between X and Y. For a formal context, elements x from X are called objects and elements y from Y are called attributes.

Definition 2: Concept forming operators: For a formal concept, $<X, Y, R>$, operators $\uparrow: 2^X \to 2^Y$ and $\downarrow: 2^Y \to 2^X$ are defined for every $A \subseteq X$ and $B \subseteq Y$ by

$$A^{\uparrow} = \{y \in Y \mid \text{for each } x \in A : <x, y> \in R\} \text{ and}$$

$$B^{\downarrow} = \{x \in X \mid \text{for each } y \in B : <x, y> \in R\}.$$

Definition 3: Formal concept: In FCA, a formal concept in $<X, Y, R>$ is a pair $<A, B>$ of $A \subseteq X$ and $B \subseteq Y$ such that $A^{\uparrow} = B$ and $B^{\downarrow} = A$. For a formal concept $<A, B>$ in $<X, Y, R>$, A and B are called the extent and intent of $<A, B>$, respectively.

Authenticated | anoop.res15@iiitmk.ac.in author's copy
Download Date | 12/31/18 11:53 AM

Definition 4: Attribute implications: In FCA, an attribute implication can be defined as an expression $A \rightarrow B$, where $A, B \subseteq Y$, and it holds in a formal context if $A^{\downarrow} \subseteq B^{\downarrow}$. It means that any object that has all the attributes in A has also all the attributes in B. It is also well known that the sets of attribute implications that are satisfied by a context satisfy Armstrong's axioms [4].

FCA makes use of formal context (Definition 1) for data analysis in which each row corresponds to attributes and the filed value denotes the relationship between them. FCA takes this formal context as input and then outputs concept lattice that reflects generalization and specialization between the derived formal concepts from the incidence matrix [11]. These formal concepts are extensively used for knowledge processing tasks containing distinct extents and intents (set of objects and their relationships). These relations are represented in the form of a formal context, $F = (X, Y, R)$, where X is a set of objects, Y is a set of attributes, and R is a binary relation between them. From this given context, FCA creates a set of objects (A) and the set of all attributes (B) that are common for these objects.

Concept lattice: The concept lattice that is built from incidence matrix (context table) determines the hierarchy of formal concepts that follow a partial ordering principle as $(A1, B1) \leq (A2, B2)$ iff $A1 \leq A2$ $(B2 \leq B1)$ and then give generalization and specialization between the concepts. That is, $(A1, B1)$ is more specific than $(A2, B2)$. The attribute implications are represented in the form of $A \rightarrow B$ over the set Y. There are several algorithms developed for generating concept lattices [5, 9, 20, 25]. An example of a formal context showing airlines and their sector of operations and corresponding concept lattice visualization are shown in Figures 1 and 2, respectively. In the formal context (Figure 1), the rows represent the concepts or objects (in this case, "Air Canada", "Air New Zealand", and "Air India") and the columns represent the set of attributes (in this case, "Latin America", "Asia", "Europe", and "Middle East"). A cross ("X") in the intersection (cell) on the formal context denotes that the object has a corresponding property. In Figures 1 and 2, it denotes that an airline is operating in that particular sector. See Ref. [6] for a more detailed and comprehensive explanation of FCA and its related theory.

A	B	C	D	E
	Latin Amer...	Asia	Europe	Middle East
Air Canada	X		X	
Air New Zealand	X	X		
Air India		X		X

Figure 1: Formal Context Showing Airlines and their Sector of Operations.

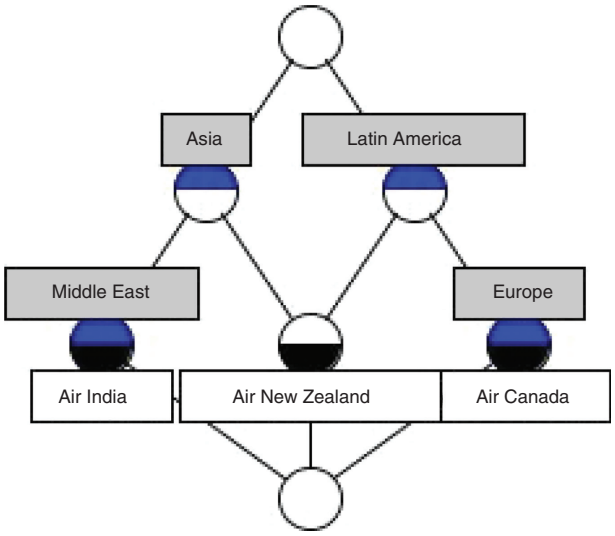


Figure 2: Concept Lattice Generated for Formal Context Given in Figure 1.

Authenticated | anoop.res15@iiitmk.ac.in author's copy
Download Date | 12/31/18 11:53 AM

3 Research Objective

The following are our main research objectives:

1. Introduce the task of RE from unstructured text and major approaches and categories for the same.
2. Propose a framework that uses machine learning approach for automatically extracting and learning subsumption relation ("is-a"), Hearst patterns ("such as", "or other", "and other", "including", "especially", etc.), and other potential indications of relations among concepts.
3. Represent the knowledge (concepts and extracted relationships) using FCA.
4. Verify experimentally the effectiveness of the method in extracting and representing real-world concepts and relationships.

4 Problem Definition

We now define the problem formally. RE problem is the task of detecting and classifying semantic relationships that connect entities or phrases or concepts in a corpus of interest. Given a static document corpus D, the relationship extraction task identifies valid relation words that connect two concepts together. For example, consider the sentence, "Alzheimer is a degenerative disease". The words "Alzheimer" and "degenerative disease" are potential concepts in a medical text document. The relationship extraction method identifies "is-a" as a potential relation that connects these two concepts. Given a static document corpus $D = d_1, d_2, \ldots, d_n$ that contains key-phrases or concepts $C = c_1, c_2, \ldots, c_n$, our problem is to identify semantically distinguishable relations that connect $c_1, c_2, \ldots, c_n$. We also address the problem of representing this knowledge using FCA, which is a widely used knowledge representation framework that comes with well-implemented mathematical models.

5 Proposed Approach

In this section, we outline our proposed approach for identifying and extracting relationships that connect entities or concepts that are extracted from unstructured text. Considering the complex sentence and language structures, concept extraction as well as RE is an extremely difficult task in NLP and information retrieval. Although there are many attempts reported on how to extract them, majority of the works are heavily dependent on the specific corpus chosen for the experiment. In previous works, we have also attempted the concept extraction task, which is guided by a topic modeling process that suits well on any plain text corpus [2, 3]. In this work, our main focus is on the RE task; thus, the process of concept extraction is not emphasized. Here, we are using an off-the-shelf tool for identifying potential entities, phrases, and concepts from our static document corpus and then implement our relationship extraction algorithm on top of it to extract relation patterns. An overall workflow of the proposed approach is shown in Figure 3.

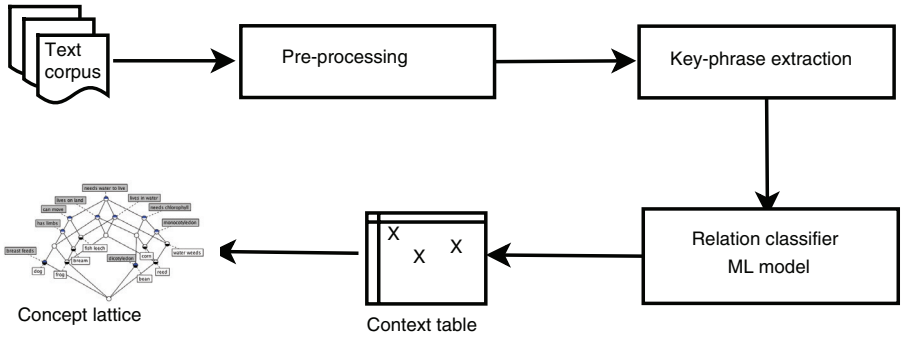


Figure 3: Overall Workflow of the Proposed Approach.

Authenticated | anoop.res15@iiitmk.ac.in author's copy
Download Date | 12/31/18 11:53 AM

6 Experimental Setup

In this section, we describe our experimental setup and a detailed evaluation of the results. Two separate experiments are conducted. The first one is for extracting semantically valid relationships that connect concepts extracted. The second one is for validating the usefulness of FCA for representing concepts and relationships extracted from a plain text corpus. The entire dataset, noun phrases, and the Python code are available in an open repository and can be freely accessed at https://github.com/anoop-research/relation-extraction.

6.1 Dataset Description

For the experiment, we have used disease description data that are publically available in unstructured form from the Medscape website (http://www.medscape.com), which is an online global destination for physicians and other healthcare professionals. This website offers the latest medical news, expert opinions, and disease details. We have crawled the website for diseases and treatment descriptions for two categories (cardiology and neurology) and collected data stored in plain text files. Some of the concepts or medical phrases extracted from those plain text files are shown in Table 1. For the cardiology category, we have collected descriptions of 45 diseases, such as acute coronary syndrome, alcoholic cardiomyopathy, heart failure, and hypertension, and for neurology, there are 42 descriptions for diseases, such as Parkinson's disease, depression, and Alzheimer's disease. A snapshot of such a description (for Parkinson's disease) is shown in Figure 4.

Table 1: Some of the Concepts/Medical Phrases Extracted for Cardiology and Neurology Categories.

Cardiology	Neurology
Acute aortic dissection	Central nervous system
Marfan syndrome	Potential toxic metabolites
Type III dissections	Pathological processes
β-Adrenergic blocker	Progressive disability
Thoracic aortic dissections	Vascular parkinsonism
Atherosclerotic disease	Thalamocortical pathway
Lymphocyte activation	Painful muscular contractions
Urine microalbumin	Umbilical cord contamination
Rheumatogenic strains	Neonatal tetanus
Streptococcal infections	Elastic membrane

Parkinson disease (PD) is one of the most common neurologic disorders, affecting approximately 1% of individuals older than 60 years and causing progressive disability that can be slowed, but not halted, by treatment. The 2 major neuropathologic findings in Parkinson disease are loss of pigmented dopaminergic neurons of the substantia nigra pars compacta and the presence of Lewy bodies and Lewy neurites. See the images below. Initial clinical symptoms of Parkinson disease include the following: Onset of motor signs includes the following: Nonmotor symptoms Nonmotor symptoms are common in early Parkinson disease. Recognition of the combination of nonmotor and motor symptoms can promote early diagnosis and thus early intervention, which often results in a better quality of life. See Clinical Presentation for more detail. Parkinson disease is a clinical diagnosis. No laboratory biomarkers exist for the condition, and findings on routine magnetic resonance imaging and computed tomography scans are unremarkable……..

Figure 4: Snapshot of Disease Description Collected from Medscape (http://www.medscape.com).

Authenticated | anoop.res15@iiitmk.ac.in author's copy
Download Date | 12/31/18 11:53 AM

6.2 Dataset Preprocessing

Dataset preprocessing concentrates on tidying up the data by removing unwanted characters, words, special symbols, and links from external sources. We have not removed stop-words from the corpus for this experiment as some words in the stop-words list may be found useful in identifying a particular relation word, for example, "is-a". Special symbols and other irrelevant characters are removed using regular expressions, and we used Snowball stemmer for reducing the words to its root word, say, "affecting" to "affect". Then, we have vectorized these words to feed into our proposed machine learning model.

6.3 Building a Machine Learning Model

Our next step is to create a machine learning classifier that learns the Hearst patterns and other potential indications of relation patterns. We have used many classification algorithms to build the model, such as decision tree, random forest, Adaboost, and support vector machine (SVM). The entire dataset has been split into 70–30, where 70% is for training and 30% for testing the model. For training and testing, we have used a server machine configured with AMD Opteron 6376 with 2.3 GHz speed and 16 core processor and 16 GB of main memory. For implementing the classifier, Python 2.7 version is used along with "sci-kit learn" [26] library. The input to these classifiers is in the form of a sentence-relation word matrix where the concept/key-phrase tagged sentences are in rows of the matrix and the relation words, such as "is-a" and "such as", are along the column. The cell contains the total count of a specific relation word that occurred in that sentence. As this is a binary classifier, "0" or "1" is given as the label based on the absence or presence of the desired relation. For this experiment, the input matrix contains 10,000 such rows, which are given as input to the four relation classifier models. After training, for testing, we present a new sentence to the trained model, and the model will output either "1" or "0" based on the presence or absence of the relation that will be converted to a formal context for building concept lattices.

For decision tree classifier, "gini" criterion is used as a measure of quality of split, and the maximum depth parameter has been chosen as 15 by a trial-and-error method. Second, random forest classifier has been implemented with the number of estimators as 300, maximum depth as 15, and random state as 42. Our Adaboost classifier used decision tree classifier as the base estimator that has a maximum depth of 8 and random state of 42. The learning rate parameter, the number of estimators, and the random state were chosen to 0.9, 500, and 1332, respectively. For our SVM classifier, we have set the number of random state as 22 and maximum number of iterations as 100.

6.4 Creating Concept Lattices

Once major concepts, phrases, and relationships have been extracted, the knowledge may be represented in an intuitive and informative way so that meaning aware applications can be built on top of the knowledge. We used FCA for deriving concept hierarchy or formal ontology that represents concepts and associated relationships we have leveraged. To build the lattice, a table with logical attributes represented as a triplet $<X, Y, R>$ should be created first. In the triplet $<X, Y, R>$, R denotes a binary relation between the objects X and Y. In our case, X denotes a set of disease names and Y denotes a set of attributes of the disease. For example, consider "hypertension" as the case; then, we have "blood pressure", "breathing disorder", "cortisol stress reactivity", etc., as its attributes. The binary relation R has a value of 1 if "hypertension" has a particular attribute or else the value will be 0. The size of the entire formal context and the concept lattice is so big. For the "cardiology" domain, the formal context contained 45 objects (disease names) and 7746 attributes (symptoms). For the "neurology" domain, there are 42 objects (disease names) and 9156 attributes (symptoms). Due to space constraints, it is impossible to show the entire context table and concept lattice that has been generated for the entire dataset. A part of such a context table created is shown in Figure 5 and the corresponding concept lattice is shown in Figure 6.

Authenticated | anoop.res15@iiitmk.ac.in author's copy
Download Date | 12/31/18 11:53 AM

A	B	C	D	E	F
	blood pres...	breathing ...	cytotoxic eff...	fulminant ...	genetic trait
hypertension	X	X			
myocarditis			X		
nephrosclerosis				X	
pulmonic stenosis					
brown syndrome					X

Figure 5: Part of a Formal Context Generated from the Original Dataset Using Our Proposed Method.

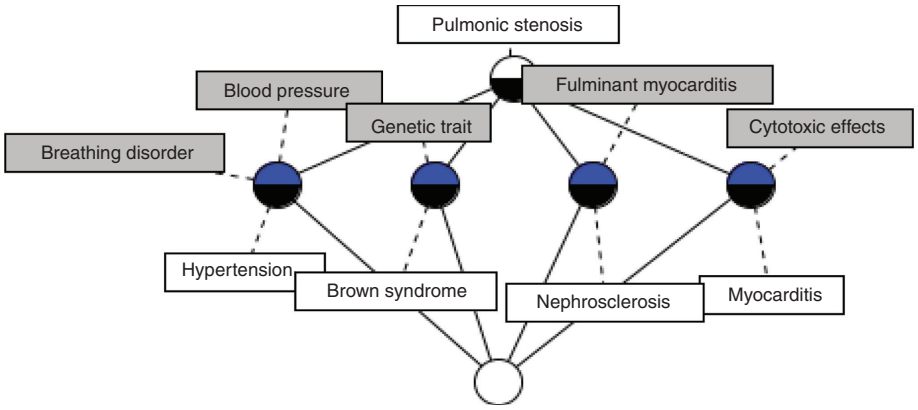


Figure 6: Part of a Concept Lattice Generated from the Original Dataset Using Our Proposed Method.

6.5 Baselines Chosen

We have chosen two baselines for comparing our proposed method. Of all classifier algorithms chosen (decision tree, random forest, Adaboost, and SVM), SVM showed better accuracy on relation classification. Thus, we compared the SVM model to the chosen baselines. We have chosen two baselines.

- Baseline 1 [7]: This approach extracts facts from natural language texts with conceptual modeling [7]. This work shows the application of FCA for extracting facts from natural language text. Their approach combines concept graphs and concept lattices for leveraging facts, which is closely associated with our proposed approach. They use concept lattices to model relationships that connect words and then these relationships for interpreting formal concepts as possible facts.
- Baseline 2 [30]: The second baseline, on the contrary, attempts to create a public dataset containing more than 400 million hypernymy relations from CommonCrawl web corpus. Although we are not considering all the relations used in their work, we have chosen this as our second baseline, as our proposed work is aligned to their workflow in a greater extent.

A comparison of the results of these baselines to our proposed framework and a detailed evaluation are discussed in Section 7.

7 Results and Evaluation

This section describes the results of our rigorous and systematic experiment on relation classification and knowledge representation using FCA. As explained in Section 6, for the relation classification, we have chosen four different algorithms, such as Adaboost, random forest, decision tree, and SVM. The precision, recall, and F1 score reported by our machine learning classifier are shown in Tables 2–4. Of the four different

Authenticated | anoop.res15@iiitmk.ac.in author's copy
Download Date | 12/31/18 11:53 AM

Table 2: Precision, Recall, and F1 Score of Different Classifier Algorithms on Our Dataset.

Algorithm	Precision	Recall	F1 score
Adaboost	0.95	0.85	0.90
Random forest	0.97	0.86	0.91
Decision tree	0.95	0.84	0.89
SVM	0.98	0.87	0.92

Table 3: Precision, Recall, and F1 Score Comparison of Baselines and Our Proposed Method for RE.

Algorithm	Precision	Recall	F1 score
Baseline (for RE) [7]	0.79	0.81	0.79
Proposed	0.91	0.87	0.88

Table 4: Precision, Recall, and F1 Score Comparison of Baselines and Our Proposed Method for Concept Lattice Generation.

Algorithm	Precision	Recall	F1 score
Baseline (for lattice generation) [30]	0.80	0.78	0.78
Proposed	0.88	0.87	0.87

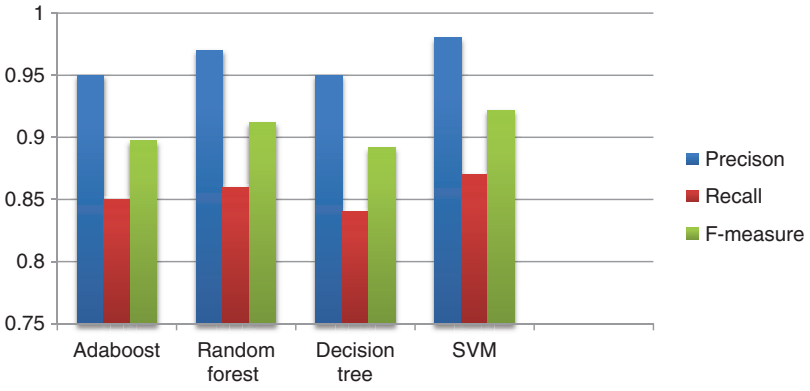


Figure 7: Graph Representation of Classifier Performance on the Chosen Dataset.

algorithms chosen with optimal parameters, SVM is found to show better classification accuracy; thus, this model has been chosen for comparing the performance of our proposed method to the chosen baselines. The normalized confusion matrix for all the four classification algorithms is shown in Figure 7 and the classification accuracy comparison in terms of precision, recall, and F1 score in a graph is shown in Figure 8.

8 Conclusions and Future Work

This paper proposed a framework for extracting relationships that connect concepts and phrases found in unstructured text documents. We make use of a machine learning-based approach for learning commonly occurring relations such as "is-a" and other patterns discussed in Hearst patterns such as "such as", "or other", "and other", "including", and "especially". This approach employs different machine learning algorithms to classify potential relationships, such as Adaboost, random forest, decision

Authenticated | anoop.res15@iiitmk.ac.in author's copy
Download Date | 12/31/18 11:53 AM

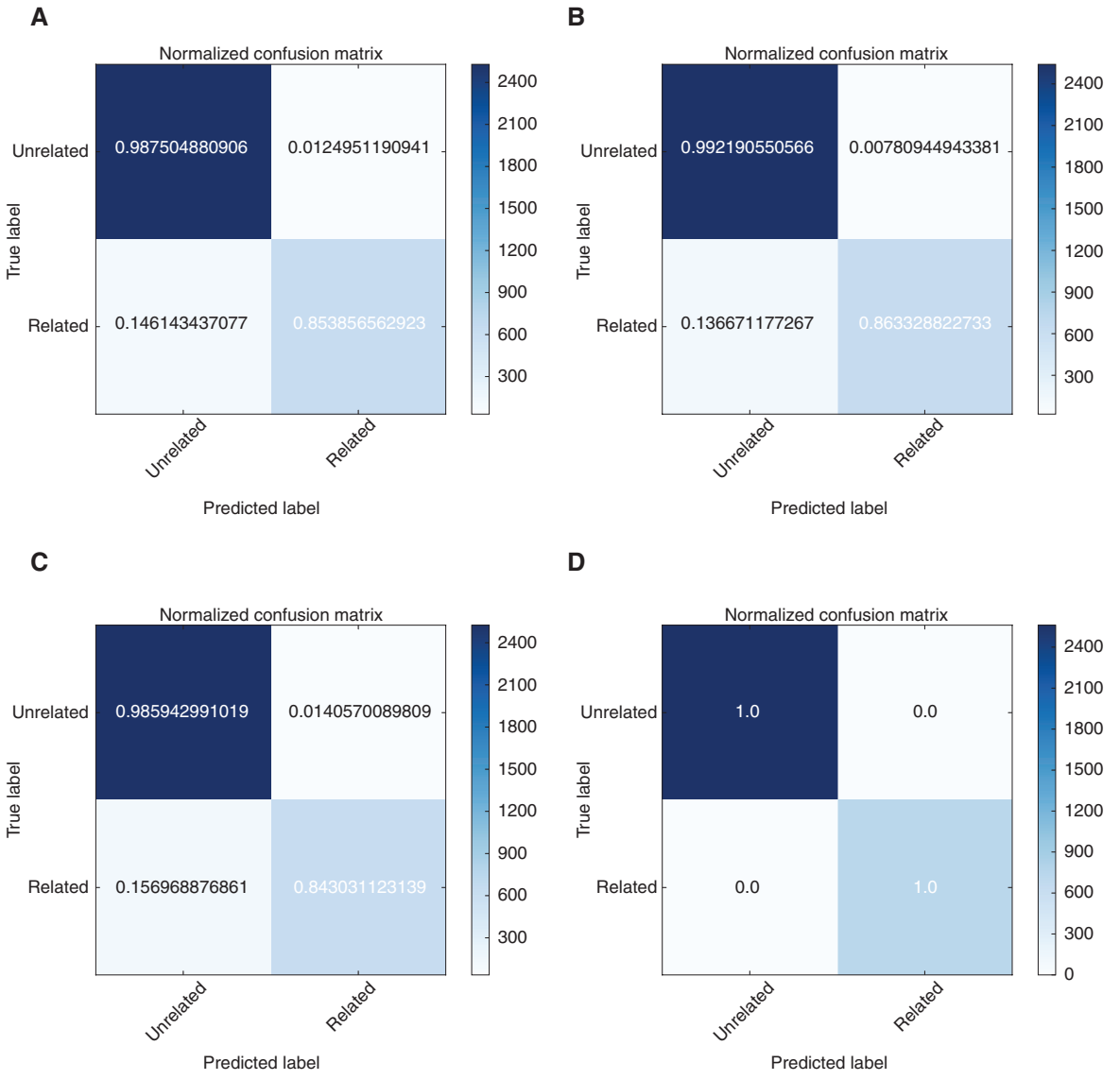


Figure 8: Normalized Confusion Matrix for (A) Adaboost, Random Forest (B), Decision Tree (C), and SVM (D) Classifiers.

tree, and SVM. This work makes use of FCA to represent noun-phrases and relations that are leveraged using our proposed RE algorithm. Experiments on real-world medical dataset collected from public web shows that this proposed method extracts better conceptual structures when compared to the baselines [7, 30].

As the end results are promising, our future work will be mainly on the direction of improving the accuracy of our classification engine and extracting more semantically valid relation patterns. This may generate more fine-grained facts from unstructured text and may aid ontology enrichment process in semantic computing paradigms. This current experimental setup will only work with static unstructured text corpus for extracting facts and building concept lattices. In the future, we may extend this for dynamically generated text contents from platforms such as social networks.

Acknowledgments: The authors thank all researchers from the Data Engineering Lab at the Indian Institute of Information Technology and Management-Kerala (IIITM-K) for their suggestions that improved the quality of this paper. The authors also acknowledge the anonymous reviewers for their constructive comments.

Authenticated | anoop.res15@iiitmk.ac.in author's copy
Download Date | 12/31/18 11:53 AM

Bibliography

[1] E. Agichtein and L. Gravano, Snowball: extracting relations from large plain-text collections, in: *Proceedings of the 5th ACM Conference on Digital Libraries*, pp. 85–94, San Antonio, TX, USA: ACM, 2000.

[2] V. S. Anoop, S. Asharaf and P. Deepak, Learning concept hierarchies through probabilistic topic modeling, *Int. J. Inf. Process.* **10** (2016), 1–11.

[3] V. S. Anoop, S. Asharaf and P. Deepak, Unsupervised concept hierarchy learning: a topic modeling guided approach, *Proc. Comput. Sci.* **89** (2016), 386–394.

[4] W. W. Armstrong, Dependency structures of data base relationships, in: *IFIP Congress*, vol. 74, pp. 580–583, 1974.

[5] E. Bartl, H. Rezankova and L. Sobisek, Comparison of classical dimensionality reduction methods with novel approach based on formal concept analysis, in: *International Conference on Rough Sets and Knowledge Technology*, pp. 26–35, Springer, Berlin/Heidelberg, 2011.

[6] R. Belohlavek, *Introduction to Formal Concept Analysis*, Department of Computer Science, Palacky University, Olomouc, 2008.

[7] M. Bogatyrev, Fact extraction from natural language texts with conceptual modeling, in: *International Conference on Data Analytics and Management in Data Intensive Domains*, pp. 89–102, Moscow, Russia: Springer, 2016.

[8] S. Brin, Extracting patterns and relations from the world wide web, in: *International Workshop on the World Wide Web and Databases*, pp. 172–183, Springer, Berlin/Heidelberg, 1998.

[9] V. Codocedo, C. Taramasco and H. Astudillo, Cheating to achieve formal concept analysis over a large formal context, in: *The 8th International Conference on Concept Lattices and Their Applications-CLA 2011*, pp. 349–362, LORIA Nancy, France, 2011.

[10] C. Cui, J. Shen, Z. Chen, S. Wang and J. Ma, Learning to rank images for complex queries in concept-based search, *Neurocomputing*, Elsevier (2017, In Press).

[11] B. A. Davey and H. A. Priestley, *Introduction to Lattices and Order*, Cambridge University Press, Cambridge, UK, 2002.

[12] D. Dligach, T. Miller, C. Lin, S. Bethard and G. Savova, Neural temporal relation extraction, *European Chapter of the Association for Computational Linguistics*, p. 746, Valencia, Spain, 2017.

[13] J. M. Gonzalez-Calabozo, F. J. Valverde-Albacete and C. Pelaez-Moreno, Interactive knowledge discovery and data mining on genomic expression data with numeric formal concept analysis, *BMC Bioinform.* **17** (2016), 374.

[14] M. A. Hearst, Automatic acquisition of hyponyms from large text corpora, in: *Proceedings of the 14th Conference on Computational Linguistics*, vol. 2, pp. 539–545, Association for Computational Linguistics, Nantes, France, 1992.

[15] T. Herawan, M. M. Deris and A. R. Hamdan, FCA-ARMM: a model for mining association rules from formal concept analysis, in: *Recent Advances on Soft Computing and Data Mining: The Second International Conference on Soft Computing and Data Mining (SCDM-2016), Bandung, Indonesia, August 18–20, 2016 Proceedings*, vol. 549, p. 213, Springer, 2017.

[16] T. Kawaumra, M. Sekine and K. Matsumura, Hyponym/hypernym detection in science and technology thesauri from bibliographic datasets, in: *Semantic Computing (ICSC), 2017 IEEE 11th International Conference on*, pp. 180–187, San Diego, CA, USA: IEEE, 2017.

[17] C. A. Kumar and S. Srinivas, Concept lattice reduction using fuzzy k-means clustering, *Expert Syst. Appl.* **37** (2010), 2696–2704.

[18] C. A. Kumar, Fuzzy clustering-based formal concept analysis for association rules mining, *Appl. Artif. Intell.* **26** (2012), 274–301.

[19] N. Kumar, M. Kumar and M. Singh, Automated ontology generation from a plain text using statistical and NLP techniques, *Int. J. Syst. Assur. Eng. Manage.* **7** (2016), 282–293.

[20] S. O. Kuznetsov and S. A. Obiedkov, Comparing performance of algorithms for generating concept lattices, *J. Exp. Theor. Artif. Intell.* **14** (2002), 189–216.

[21] Y. Lin, S. Shen, Z. Liu, H. Luan and M. Sun, Neural relation extraction with selective attention over instances, in: *Proceedings of ACL*, vol. 1, pp. 2124–2133, 2016.

[22] P. Monnin, M. Lezoche, A. Napoli and A. Coulet, Using formal concept analysis for checking the structure of an ontology in LOD: the example of DBpedia, in: *23rd International Symposium on Methodologies for Intelligent Systems, ISMIS*, 2017.

[23] E. Negm, S. AbdelRahman and R. Bahgat, PREFCA: a portal retrieval engine based on formal concept analysis, *Inf. Process. Manage.* **53** (2017), 203–222.

[24] H. Oliveira, R. Lima, R. D. Lins, F. Freitas, M. Riss and S. J. Simske, A concept-based integer linear programming approach for single-document summarization, in: *Intelligent Systems (BRACIS), 2016 5th Brazilian Conference on*, pp. 403–408, Recife, Pernambuco, Brazil: IEEE, 2016.

[25] J. Outrata and V. Vychodil, Fast algorithm for computing fixpoints of Galois connections induced by object-attribute relational data, *Inf. Sci.* **185** (2012), 114–127.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel and J. Vanderplas, Scikit-learn: machine learning in Python, *J. Mach. Learn. Res.* **12** (2011), 2825–2830.

[27] S. Roller and K. Erk, Relations such as hypernymy: identifying and exploiting Hearst patterns in distributional vectors for lexical entailment, arXiv preprint arXiv:1605.05433 (2016).

[28] S. K. Sahu, A. Anand, K. Oruganty and M. Gattu, Relation extraction from clinical texts using domain invariant convolutional neural network, arXiv preprint arXiv:1606.09370 (2016).

Authenticated | anoop.res15@iiitmk.ac.in author's copy
Download Date | 12/31/18 11:53 AM

[29] J. Seitner, C. Bizer, K. Eckert, S. Faralli, R. Meusel, H. Paulheim and S. Ponzetto, A large database of hypernymy relations extracted from the web, in: *Proceedings of the 10th edition of the Language Resources and Evaluation Conference*, Portoroz, Slovenia, 2016.
[30] J. Seitner, C. Bizer, K. Eckert, S. Faralli, R. Meusel, H. Paulheim and S. Ponzetto, A large database of hypernymy relations extracted from the web, in: *Proceedings of the 10th Edition of the Language Resources and Evaluation Conference*, Portoroz, Slovenia, 2016.
[31] V. A. Semenova and S. V. Smirnov, Intelligent analysis of incomplete data for building formal ontologies, in: *CEUR Workshop Proceedings*, vol. 1638, pp. 796–805, 2016.
[32] P. K. Singh, C. A. Kumar and A. Gani, A comprehensive survey on formal concept analysis, its research trends and applications, *Int. J. Appl. Math. Comput. Sci.* **26** (2016), 495–516.
[33] A. Sun, A two-stage bootstrapping algorithm for relation extraction, in: *Proceedings of Recent Advances in Natural Language Processing*, pp. 76–82, Borovets, Bulgaria, 2009.
[34] R. Wille, Restructuring lattice theory: an approach based on hierarchies of concepts, in: *Ordered Sets*, pp. 445–470, Springer, The Netherlands, 1982.
[35] R. Wille, Concept lattices and conceptual knowledge systems, *Comput. Math. Appl.* **23** (1992), 493–515.
[36] A. Yates, M. Cafarella, M. Banko, O. Etzioni, M. Broadhead and S. Soderland, TextRunner: open information extraction on the web, in: *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pp. 25–26, Association for Computational Linguistics, Rochester, New York, 2007.
[37] M. Zhao and S. Zhang, Identifying and validating ontology mappings by formal concept analysis, in: *Proceedings of the 15th International Semantic Web Conference*, pp. 61–72, Kobe, Japan, 2016.

Authenticated | anoop.res15@iiitmk.ac.in author's copy
Download Date | 12/31/18 11:53 AM

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 89 (2016) 386 – 394

Procedia Computer Science

Twelfth International Multi-Conference on Information Processing-2016 (IMCIP-2016)

Unsupervised Concept Hierarchy Learning: A Topic Modeling Guided Approach

V. S. Anoop[a,*], S. Asharaf[a] and P. Deepak[b]

[a] *Indian Institute of Information Technology and Management, Kerala (IIITM-K), Thiruvananthapuram, India*
[b] *Queen's University Belfast, UK*

Abstract

This paper proposes an efficient and scalable method for concept extraction and concept hierarchy learning from large unstructured text corpus which is guided by a topic modeling process. The method leverages "concepts" from statistically discovered "topics" and then learns a hierarchy of those concepts by exploiting a subsumption relation between them. Advantage of the proposed method is that the entire process falls under the unsupervised learning paradigm thus the use of a domain specific training corpus can be eliminated. Given a massive collection of text documents, the method maps topics to concepts by some lightweight statistical and linguistic processes and then probabilistically learns the subsumption hierarchy. Extensive experiments with large text corpora such as BBC News dataset and Reuters News corpus shows that our proposed method outperforms some of the existing methods for concept extraction and efficient concept hierarchy learning is possible if the overall task is guided by a topic modeling process.

© 2016 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of organizing committee of the Organizing Committee of IMCIP-2016

Keywords: Concept Learning; Latent Dirichlet Allocation; Subsumption Hierarchy; Text Mining; Topic Modeling.

1. Introduction

The overwhelming amount of text data being produced and consumed by applications show that text is increasingly considered as an important and most widely used data type. Due to the unprecedented growth of web enabled applications in the recent past, the relevance of extracting useful patterns from huge dumps of text has become crucial for knowledge discovery and decision making. There are still a lot of avenues where text data is yet to be exploited fully and we need new algorithms and techniques to mine such data. As the amount of available data grows, the traditional computational methods became inefficient to process such huge amount of data.

As the volume of data increases, existing systems may take much time and computational resources for the processing and we can't rely on such systems for efficient results. The volume of data may be in such a way that it can't even be stored in traditional storage devices and thus leveraging useful insights from such data is not practical using already developed algorithms. Platforms such as social networks, e-commerce websites, blogs and research journals generate such data in the form of unstructured text and it is essential to analyze, synthesis and process such

*Corresponding author. Tel.: +91 9747091417.
E-mail address: anoop.res15@iiitmk.ac.in

1877-0509 © 2016 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of organizing committee of the Organizing Committee of IMCIP-2016
doi:10.1016/j.procs.2016.06.086

data for efficient retrieval of useful information. Thus we need more efficient and scalable processing mechanisms for such huge kind of data archives from which potentially useful patterns and knowledge can be leveraged.

Concept mining and extraction tasks play a key role in any text mining activity because potential knowledge can be leveraged using these concepts and thus useful for many tasks such as information retrieval, classification and opinion mining to name a few. In this era of data explosion, organizations are flooded with data and striving hard to store, process, manage and most importantly to extract knowledge out of it. The traditional algorithms can be improved a lot when it comes to the extraction of entities, concepts, aspects and their interrelationships. To overcome this shortfall, a significant amount of research has been carried out in the recent past for leveraging underlying thematic and semantic structure from text archives. As a result a good number of algorithmic techniques were introduced which are proved to be efficient for the discovery of themes and semantics underlying high dimensional data. One such algorithm is the probabilistic topic model[1] which can bring out latent themes from huge text dumps.

Contributions: In this work, we propose a new approach which can efficiently leverage concepts from huge amount of unstructured text data. The proposed hybrid approach comprises topic modeling, a newly introduced statistical weighting scheme called $tf - itf$ (term frequency – inverse topic frequency) and basic linguistic process such as parts-of-speech tagging. The advantage of this proposed method is that it is completely unsupervised and works well with huge archives of text data and highly scalable.

Organization: The rest of this paper is organized as follows. We briefly review related works in Section 2. Section 3 introduces the novel approach we have proposed. Detailed explanation of the implementation details is presented in Section 4, and the evaluation of the proposed method is discussed in Section 5 and finally we draw a conclusion and discuss future work in Section 6.

2. State of the Art

Concepts can be defined as a sequence of words that may represent real or imaginary ideas or entities expressed in plain text. Concept extraction is the process of extracting such concepts for document summarization and related tasks and extraction of relevant concept has got wider recognition in the recent past. This is due to the wide variety of applications which are mainly dealing with text data such as e-commerce websites. Due to the potential applications, a large number of research literatures are available in the field of concept extraction or concept mining which proposed many algorithms with varying degrees of success. In this section, we give emphasis on past literatures in automated concept extraction algorithms and briefly discuss works closely related to our proposed framework.

A system which extracts concepts from user tag and query log dataset is proposed by Parameswaran *et al.*[2] which uses techniques similar to association rule mining. This method uses features such as frequency of occurrences and the popularity among users for extracting core concepts and attempts to build a web of concepts. Even though this algorithm can be applied to any large dataset, a lot of additional processing is required when dealing with web pages. A bag-of-word approach was proposed by Gelfand *et al.*[3] for concept extraction from plain text and used these to form a closely tied semantic relations graph for representing relationships between them. They have applied this technique specifically for some classification tasks and found that their method produces better concepts than the Naive Bayes text classifier.

Dheeraj Rajagopal *et al.*[4] introduced another graph based approach for commonsense concept extraction and detection of semantic similarity among those concepts. They used a manually labeled dataset of 200 multi-word concept pairs for evaluating their parser capable of detecting semantic similarity and showed that their method was capable of effectively finding syntactically and semantically related concepts. The main disadvantage of that method is the use of manually labeled dataset and the creation of such dataset is time consuming and requires human effort. Another work reported in this domain is the method proposed by Krulwich and Burkey[5] which uses a simple heuristics rule based approach to extract key phrases from document by considering visual clues such as the usage of bold and italic characters as features. They have shown that this technique can be extended for automatic document classification experiments.

Ramirez and Mattmann[6] proposed a system named Automatic Concept Extractor (ACE), specifically designed for extracting concepts from HTML pages and making use of the text body and some visual clues on HTML tags for identifying potential concepts. Their method could outperform some of the reported algorithms that was prevalent at

that time. Another system named GenEx, proposed by Turney[7] which employed a genetic algorithm supported rule learning mechanism for concept extraction.

A key phrase extraction system called Automatic Keyphrase Extraction (KEA) developed by Witten *et al.*[8] was reported in the concept extraction literatures which creates a Naive Bayes learning model with known key phrases extracted from training documents and uses this model for inferring key phrases from new set of documents. As an extension to this KEA framework, Song *et al.*[9] proposed a method which uses the information gain measure for ranking candidate key phrases based on some distance and tf-idf features which was first introduced in[8]. Another impressive and widely used method was introduced by Frantzi *et al.*[10] which extracts multi-word terms from medical documents and named as C/NC method. The algorithm uses a POS tagger POS patten filter for collecting noun phrases and then uses some statistical measures for determining the termhood of candidate multi-words.

The proposed method in this paper is a hybrid approach incorporating statistical methods such as topic modeling and tf-itf weighting and some lightweight linguistic processes such as POS tagging and analysis for leveraging concepts from text. We expect the learnt concept hierarchy to be close to the real world understanding of concepts which we will quantify using evaluation measures such as precision, recall and f-measure.

3. Latent Dirichlet Allocation

A good number of topic modeling algorithms are introduced in the recent past which varies in their method of working mainly with the assumptions they adopt for the statistical processing. An automated document indexing method based on a latent class model for factor analysis of count data in the latent semantic space has been introduced by Thomas Hofman[11]. This generative data model called Probabilistic Latent Semantic Indexing (PLSI), considered as an alternative to the basic Latent Semantic Indexing has a strong statistical foundation. The basic assumption of PLSI is that each word in a document corresponds to only one topic. Later, Blei *et al.*[12] introduced a new topic modeling algorithm known as Latent Dirichlet Allocation (LDA) which is more efficient and attractive than PLSI. This model assumes that a document contain multiple topics and such topics are leveraged using a Dirichlet Prior process. In the following section, we will briefly describe the underlying principle of LDA.

Even though a LDA works well on broad ranges of discrete datasets, the text is considered to be a typical example to which the model can be best applied. The process of generating a document with n words by LDA can be described as follows[12]:

1. Choose the number of words, n, according to Poisson Distribution;
2. Choose the distribution over topics, θ, for this document by Dirichlet Distribution;
   (a) Choose a topic $T^{(i)} \sim \text{Multinomial}(\theta)$
   (b) Choose a word $W^{(i)}$ from $P(W^{(i)}|T^{(i)}, \beta)$

Thus the marginal distribution of the document can be obtained from the above process as:

$$P(d) = \int_{\theta} \left(\prod_{i=1}^{n} \sum_{T^{(i)}} P(W^{(i)}|T^{(i)}, \beta) P(T^{(i)}|\theta) \right) P(\theta|\alpha) d\theta \tag{1}$$

where, $P(\theta|\alpha)$ is derived by Dirichlet Distribution parameterized by α, and $P(W^{(i)})|T^{(i)}, \beta)$ is the probability of $W^{(i)}$ under topic $T^{(i)}$ parameterized by β. The parameter α can be viewed as a prior observation counting on the number of times each topic is sampled in a document, before we actually seen any word from that document. The parameter β is a hyper-parameter determining the number of times words are sampled from a topic[12], before any word of the corpus is observed. At the end, the probability of the whole corpus D can be derived by taking the product of all documents' marginal probability as given below:

$$P(D) = \prod_{i=1}^{M} P(d_i) \tag{2}$$

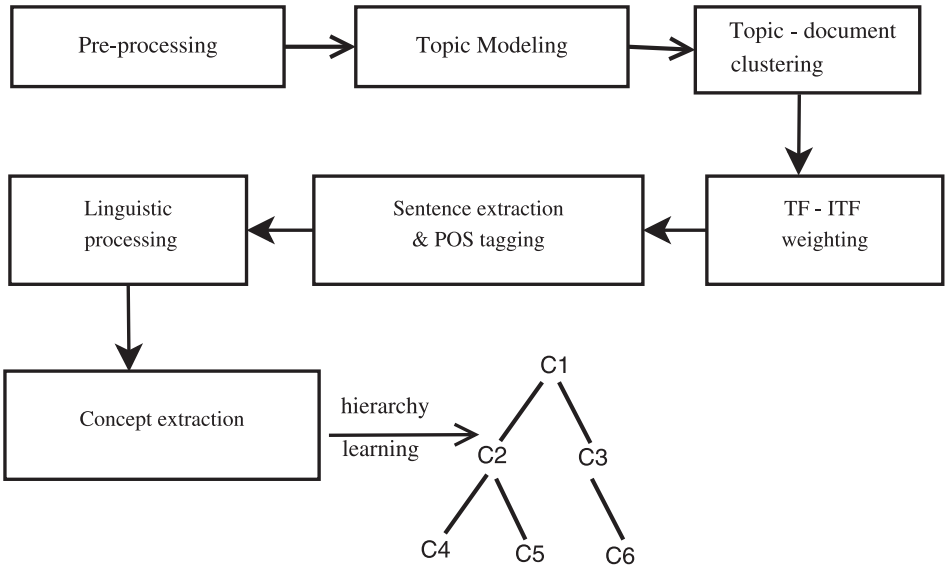


Fig. 1. Workflow of the Proposed Framework, where C1, C2, C3, C4, C5 and C6 represent Concepts.

4. Proposed Approach

In the area of text mining, topic models or specifically probabilistic topic models are suite of algorithms which got wider recognition for its ability to leverage hidden thematic information from huge archives of text data. Text mining researchers are making use of topic modeling algorithms such as Latent Semantic Indexing (LSI), Probabilistic Latent Semantic Indexing (pLSI), Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA) etc., extensively for bringing out the themes or so called "topics" from high dimensional unstructured data. Among all these algorithms, LDA has got lot of attention in the recent past and is widely using because of its easiness of implementation and potential applications. Even though the power of LDA algorithm has been extensively used for leveraging topics, very few studies have been reported for mapping these statistically outputted topics to semantically rich concepts. Our proposed framework is an attempt to address this issue by making use of LDA algorithm to generate topics and we leverage concepts from such topics by using a new statistical weighting scheme and some lightweight linguistic processes. The overall work flow of the proposed approach is depicted in Fig. 1.

Our framework can be divided into 2 modules (i) concept extraction and (ii) concept hierarchy learning. The concept extraction module extract concepts from topics generated by LDA algorithm and the concept hierarchy learning module learns a hierarchy of extracted concepts by inducing a subsumption hierarchy learning algorithm. Detailed explanation of these modules are given below.

4.1 Concept extraction

In this module, we introduce a topic to concept mapping procedure for leveraging potential concepts from statistically computed topics which are generated by the LDA algorithm. The first step of the proposed framework deals with the preprocessing of data which is meant for removing unwanted and irrelevant data and noises. Latent Dirichlet Allocation algorithm is executed on top of this preprocessed data which in turn generate topics through the statistical process. A total of 50 topics have been extracted by tuning the parameters of LDA algorithm. Once we got the sufficient topics for the experiment, for each topic, we have created a topic - document cluster by grouping the documents which generated such a topic and the same process has been executed for all topics under consideration.

Now, we introduce a new weighting scheme called $tf - itf$ (term frequency – inverse topic frequency) which is used for finding out highly contributing topic word in each topic. We bring this weighting scheme to filter out the relevant candidate topic words. Term frequency (tf) is the total number of times that particular topic word comes in the

topic - document clusters. Normalized term frequency, N_{tf} of a topic word T_w can be calculated as:

$$N_{tf} = \frac{\text{number of times term } T_w \text{ appears in the mapped collection}}{\text{total number of terms in that collection}} \tag{3}$$

Inverse topic frequency I_{tf} is calculated as:

$$I_{tf} = \frac{\text{total number of documents in the mapped collection}}{\text{number of documents with term } tw \text{ in that collection}} \tag{4}$$

$tf - itf$ is calculated using the following equation:

$$tf - itf = N_{tf} * I_{tf} \tag{5}$$

This step is followed by a sentence extraction process in which all the sentences which contain the topic words which have high tf-itf weight are extracted. Next, we apply a parts of speech tagging on these sentences and extract only noun and adjective tags as we are only concentrating on the extraction of concepts. In linguistic pre-processing step, we take Noun + Noun, Noun + Adjective and (Adjective/Noun) + Noun combinations of words from the tagged collection. Concept identification is the last step in the process flow in which we find out the term count of all the combinations of Noun + Noun, Noun + Adjective and (Adjective/Noun) + Noun. A positive term count implies that the current multi word can be a potential "concept" and if we get a zero term count, then that multi word can be ignored. The newly proposed algorithm for extracting the concepts is shown in Algorithm 1.

```
1:  procedure ExtractConcepts (TopicCollection)
2:      Create topic-document clusters for each topic
3:      Compute tf − itf weight for topic words in every topic-document cluster
4:      Filter n topic words having highest tf − itf weight for every topic
5:      Extract sentences from the corpus which contain highest tf − itf weighted words
6:      Execute POS tagging on the extracted sentences
7:      Extract those words tagged as nouns (NNP, NNS or NN) and adjective (JJ)
8:      Find out combinations of Noun + Noun or Adjective + Noun and create a multi-word collection, MW_c
9:      while |MW_c| ≠ 0 do
10:         Calculate termCount(T_c) for each multi-word MW in MW_c
11:         if T_c > 0 then
12:             Add MW into concept repository
13:             Remove MW from MW_c
14:             Fetch next MW from MW_c
15:         else
16:             Remove MW from MW_c
17:             Fetch next MW from MW_c
18:         end if
19:     end while
20: end procedure
```

Algorithm 1. Algorithm for Concept Extraction

4.2 Concept hierarchy learning

In this module we derive hierarchical organization of leveraged concepts using a type of co-occurrence called "subsumption" relation. Subsumption relation is found to be simple but very effective way of inferring relationships between words and phrases without using any training data or clustering methods. The basic idea behind subsumption relation is very simple: for any two concepts C_a and C_b, C_a is said to be subsume C_b if 2 conditions hold. $P(C_a|C_b) = 1$ and $P(C_b|C_a) < 1$. To be more specific, C_a subsumes C_b if the documents which C_b occurs in are a subset of the documents which C_a occurs in. Because C_a subsumes C_b and because it is more frequent, in the hierarchy, C_a is the parent of C_b.

```
1: procedure LearnConceptHierarchy (ConceptCollection)
2:     Consider any pair of concepts, say C_a and C_b
3:     Compute P(C_a|C_b) and P(C_b|C_a)
4:     if P(C_a|C_b) = 1 and P(C_b|C_a) < 1 then
5:         Assign C_a as the parent of C_b
6:     else
7:         Fetch next concept pairs
8:     end if
9:     Goto step 2 and repeat the same for all concept pairs
10: end procedure
```

Algorithm 2. Algorithm for Learning Concept Hierarchies

5. Experimental Setup

This section concentrates on the implementation details of our proposed framework and concept extraction and hierarchy learning procedures are discussed in detail.

5.1 Concept extraction

Here, concept extraction module of the framework is discussed. This module concentrates on tasks such as data collection and pre-processing, topic modeling, topic-document clustering, tf-itf weighting, sentence extraction and POS tagging, linguistic pre-processing etc., for identifying concepts and a detailed explanation of each step is given below.

5.1.1 Dataset collection and pre-processing

We are using publicly available datasets such as Reuters Corpus Volume 1 dataset[14] and BBC News Dataset[15] for the experiment. Reuters is the world's biggest international news agency and cater different news and related information through their website, video, interactive television and mobile platforms. Reuters Corpus Volume 1 is in XML format and is freely available for research purpose. Text messages are extracted by a thorough pre-processing such as removing XML tags, URLs and other special symbols and then created a new dataset exclusively for our experiment. BBC provides two benchmarked news article datasets which is freely available for machine learning research. The general BBC dataset consist of 2225 text documents directly from their website corresponding to stories in five areas such as business, entertainment, politics, sports and technology, from 2004 to 2005. A thorough pre-processing such as stemming, and removal of stop-word, URLs and special characters on this dataset and made an experiment ready copy of the original dataset.

5.1.2 Topic modeling

Latent Dirichlet Allocation (LDA) algorithm has been applied on the pre-processed dataset to leverage topics for this experiment. The number of iterations is set to 300 as Gibbs sampling method usually approaches the target distribution after 300 iterations. The number of topics is set to 50 and a snapshot of 5 topics we have randomly chosen is shown in Table 1.

5.1.3 Topic - document clustering

In this step, we consider each topic and then grouped and clustered top 50 documents which contributed the creation of that specific topic. This has been done for all the 50 topics of our choice. As an outcome, we have got 50 such clusters that contain documents which generated the topics.

Table 1. Top 10 Topic Words from First 5 Topics.

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
web [0.0048]	set [0.0047]	system [0.0064]	site [0.0042]	technology [0.0035]
search [0.0048]	software [0.0032]	music [0.0045]	net [0.0038]	mobile [0.0022]
online [0.0047]	virus [0.0028]	devices [0.0043]	spam [0.0035]	people [0.0021]
news [0.0046]	users [0.0027]	players [0.0035]	mail [0.0028]	using [0.0021]
google [0.0033]	firms [0.0025]	media [0.0032]	firm [0.0027]	data [0.0017]
people [0.0032]	microsoft [0.0025]	digital [0.0027]	data [0.0024]	phone [0.0013]
information [0.0032]	security [0.0022]	market [0.0024]	attacks [0.0019]	phones [0.0012]
internet [0.0029]	windows [0.0022]	technology [0.0022]	network [0.0018]	mobiles [0.0012]
website [0.0027]	file [0.0013]	consumer [0.0021]	web [0.0016]	camera [0.0012]
users [0.0020]	programs [0.0011]	technologies [0.0018]	research [0.0014]	operators [0.0011]

Table 2. Top Concepts Returned Against each Topic.

Concepts from Topic 1	Concepts from Topic 2	Concepts from Topic 3	Concepts from Topic 4	Concepts from Topic 5
web search	software users	music players	spam mail	mobile technology
search engine	virus programs	digital media	spam web site	mobile camera
google news	windows security	digital technology	network research	mobile operators
online news search	software firms	consumer devices	research firm	phone data
google search engine	microsoft programs	market system	web site attacks	mobile phone operators

5.1.4 TF-ITF weighting

Here, we compute the $tf - itf$ *(term frequency - inverse topic frequency)* weight of each word in every topic using Eq. (3), Eq. (4) and Eq. (5) to find out highly used topic words in the collection. Table 1 also shows topic words along with their tf-itf weight.

5.1.5 Sentence extraction & POS tagging

In sentence extraction step, we consider topic words having highest tf-itf weight and then extract sentences containing these topic words from the topic - document clusters. Then a parts of speech tagging has been done to identify words tagged as nouns and adjectives from these sentences as our aim is to extract potential "concepts" from the repository. For this experiment, Natural Language Toolkit (NLTK)[13] has been used which contains libraries for Natural Language Processing for Python programming language.

5.1.6 Linguistic processing & Concept identification

We have collected all words which are tagged as Nouns (NN/NNP/NNS) and Adjectives (JJ) and then taken all possible combinations of Noun + Noun, Adjective + Noun and (Noun/Adjective) + Noun. The outcome of this process is shown in Table 3. The term count for each of these multi word term is then calculated against the original corpus and a positive term count implies that the corresponding multi-word term can be a potential concept and we eliminate the term if we get a zero term count. This process has been repeated for all the multi-words and the result is shown in Table 4.

5.2 Concept hierarchy learning

Concept hierarchy learning module concentrates on leveraging a subsumption hierarchy[16] depicting an "is-a" relation between the concepts identified by the proposed algorithm. Subsumption relation is simple but considered as an important relationship type in any ontological structure and we calculate two probability conditions for the same. For any given two concepts, we first calculate $P(C_1|C_2)$ and then $P(C_2|C1)$, in order to establish a subsumption

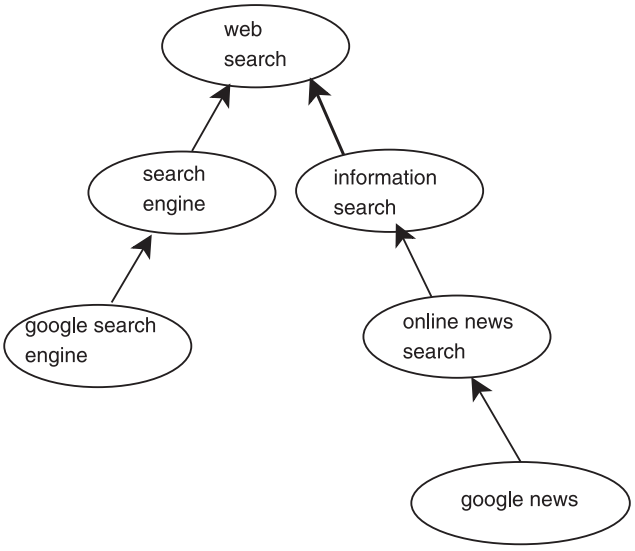


Fig. 2. Part of a Subsumption Hierarchy Learned for "Topic 1" using Algorithm 2.

Table 3. Comparison of ACE, ICE and our Proposed Method.

Algorithm	Precision	Recall	F1
ACE	0.2372	0.2689	0.2517
ICE	0.7113	0.8147	0.7595
Our approach	**0.8165**	**0.8901**	**0.8516**

relation, the former probability must be 1 and the latter should be less than 1. In other words, C_1 subsumes C_2 if the documents in which C_2 occurs is a subset of the documents which C_1 occurs in.

For instance, consider two concepts *search engine* and *google search engine*, we have computed *P(search engine — google search engine)* and *P(google search engine—search engine)* and found that the number of documents in which *google search engine* occurs is a subset of number of documents in which *search engine* occurs. That means there exists a subsumption relation between these two concepts and *google search engine* concept may be subsumed by *search engine* concept. This process has been repeated for all concepts in the collection, and a part of such a hierarchy generated for concepts from Topic 1 is shown in Fig. 2.

6. Evaluation of Results

Here we evaluate the results produced by our proposed method and precision and recall measures are used for evaluating the quality of concepts leveraged. We have first created a human generated concept repository and kept for verifying against the machine generated concepts. Precision computes the fraction of machine extracted concepts that are also human generated, and recall measures concepts which are generated by proposed algorithm that are also human authored. In information retrieval, it is estimated that achieving high precision and recall at same time is difficult and using a measure called F1, we can balance these two. Here, true positive is defined as the number of overlapped concepts between human authored concepts and concepts generated by the algorithm, false positive is the number of extracted concepts that are not truly human authored concepts and false negative is the human authored concepts that are missed by the concept extraction method. Using these measures, we have compared our proposed method against some of the existing concept extraction algorithms and the result is shown in Table 3.

For further confirmation of the quality of concepts generated by our method, we have used DBPedia precision search and find service and each of the machine generated concepts are passed to the service as query string and then evaluated

Table 4. DBPedia Link Counts Returned Against Concepts.

Concept	Count	Concept	Count	Concept	Count
web search	172	software users	25	music players	76
search engine	1386	virus programs	10	digital media	1481
google news	59	windows security	25	digital technology	261
online news search	43	software firms	10	consumer devices	39
google search engine	22	microsoft programs	8	market system	57

the number of links returned by the search service. Service returns a positive count shows that the corresponding concept is a valid one and otherwise the concept cannot be considered. Concepts and their corresponding link count returned by the DBPedia service is shown in Table 4.

7. Conclusions and Future Work

We introduced a novel framework for extracting core concepts from large collection of text documents which is guided by a topic modeling process. One module of this framework also deals with learning a subsumption hierarchy which exploits "is-a" relationships among identified concepts. Extensive but systematic experiments with large datasets such as Reuters and BBC news corpus shows that the proposed method outperforms some of the already available algorithms and better concept identification is possible with this framework. As the end results are promising, the future work will be mainly on the directions of measuring the scalability of proposed framework by using large datasets for further experiments. Apart from the basic "is-a" relationships so far identified, we would like to leverage more relationships among concepts so that a complete ontology generation will be possible.

References

[1] D. M. Blei, Probabilistic Topic Models, *Communications of the ACM*, vol. 55(4), pp. 77–84, (2012).
[2] A. Parameswaran, H. Garcia-Molina and A. Rajaraman, Towards the Web of Concepts: Extracting Concepts from Large Datasets, *Proceedings of the VLDB Endowment*, vol. 3(1–2), pp. 566–77, (2010).
[3] B. Gelfand, M. Wulfekuler and W. F. Punch, Automated Concept Extraction from Plain Text, *In AAAI 1998 Workshop on Text Categorization*, pp. 13–17, (1998).
[4] D. Rajagopal, E. Cambria, D. Olsher and K. Kwok, A Graph-based Approach to Commonsense Concept Extraction and Semantic Similarity Detection, In *Proceedings of the 22nd International Conference on World Wide Web Companion*, pp. 565–570, (2013).
[5] B. Krulwich and C. Burkey Learning User Information Interests through Extraction of Semantically Significant Phrases, In *Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access*, pp. 100–112, (1996).
[6] P. M. Ramirez and C. A. Mattmann, ACE: Improving Search Engines Via Automatic Concept Extraction, *Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration*, 2004, IRI 2004, pp. 229–234, (2004).
[7] P. D. Turney, Learning Algorithms for Keyphrase Extraction, *Information Retrieval*, vol. 2(4), pp. 303–36, (2000).
[8] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin and C. G. Nevill-Manning KEA: Practical Automatic Keyphrase Extraction, In *Proceedings of the fourth ACM conference on Digital Libraries*, pp. 254–255, (1999).
[9] M Song, I. Y. Song and X. Hu, KP Spotter: A Flexible Information Gain-Based Keyphrase Extraction System, In *Proceedings of the 5th ACM International Workshop on Web Information and Data Management*, pp. 50–53, (2003).
[10] K. Frantzi, S. Ananiadou and H. Mima, Automatic Recognition of Multi-Word Terms, The c-value/nc-value Method, *International Journal on Digital Libraries*, vol. 3(2), pp. 115–30, (2000).
[11] T. Hofmann, Probabilistic Latent Semantic Indexing, In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 50–57, (1999).
[12] D. M. Blei, A. Y. Ng and M. I. Jordan, Latent Dirichlet Allocation, *The Journal of Machine Learning Research*, vol. 3, pp. 993–1022, (2003).
[13] S. Bird, NLTK: The Natural Language Toolkit. In *Proceedings of the COLING/ACL on Interactive Presentation Sessions*, pp. 69–72, (2006).
[14] D. D. Lewis, Y. Yang, T. Rose and F. Li, RCV1: A New Benchmark Collection for Text Categorization Research, *Journal of Machine Learning Research*, vol. 5, pp. 361–397, (2004).
[15] D. Greene and P. Cunningham, Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering, In *Proceedings of the 23rd International Conference on Machine Learning*, pp. 377–384, (2006).
[16] M. Sanderson and B. Croft, Deriving Concept Hierarchies from Text. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 206–213, (1999).

Topic Modeling for Unsupervised Concept Extraction and Document Ranking

V.S. Anoop[1(✉)], S. Asharaf[2], and P. Deepak[3]

[1] Data Engineering Lab, Indian Institute of Information Technology and Management - Kerala, Thiruvananthapuram, India
anoop.res15@iiitmk.ac.in
[2] Indian Institute of Information Technology and Management - Kerala, Thiruvananthapuram, India
asharaf.s@iiitmk.ac.in
[3] Queens University, Belfast, UK
d.padmanabhan@qub.ac.uk

Abstract. This paper proposes a framework which induces semantically rich concepts from probabilistically generated topics by a topic modeling algorithm. In this method an off-the-shelf tool has been used to extract noun-phrases as word bi-grams and tri-grams from the static document corpus and then models the topics using Latent Dirichlet Allocation algorithm. Additionally, we show that a small extension to our proposed framework can better rank documents in a large collection, which is a well studied area in information retrieval. Experiments conducted on three real world datasets show that this proposed framework outperforms state-of-the-art methods used for extracting concepts and ranking documents. When compared with the baselines chosen, our proposed concept extraction method showed an increased f-measure in the range of 16.65% to 22.04% and the proposed topic modeling guided document retrieval method showed 7.6%–16.61% increase in f-measure.

Keywords: Topic modeling · Concept extraction · Document ranking · Latent Dirichlet Allocation

1 Introduction

Topic modeling algorithms such as Latent Dirichlet Allocation (LDA) [1], Probabilistic Latent Semantic Indexing (PLSI) [2] and Probabilistic Latent Semantic Analysis (PLSA) [3] are proved to be extensively useful in bringing out hidden themes from textual content which are used for further analysis. Given a large collection of text documents, these algorithms generate such themes by representing each document as a random mixture over latent topics and assumes each topic be a probabilistic distribution over words. This idea is used in various scenarios such as document analysis and pattern identification in text mining,

© Springer International Publishing AG 2018
S.M. Thampi et al. (eds.), *Intelligent Systems Technologies and Applications*,
Advances in Intelligent Systems and Computing 683,
https://doi.org/10.1007/978-3-319-68385-0_11

because text is considered to be the best data on which topic modeling can be easily applied. Majority of the topic modeling algorithms work on the assumption called "bag-of-words" to generate "topics". These statistically generated "topics" are sequences of word unigrams. The interpretation of such topics still remain as a hindrance to most of the text mining practitioners [4]. In real life scenarios, humans understand the concepts and key-phrases as a combination of words such as bi-grams and tri-grams. Splitting these bigrams and trigrams into unigrams cause its semantic meaning to be lost and that would generate ambiguous and irrelevant topics. For example, the concept "network security attack" is more interpretable for humans compared to the word unigram - "attacks". Chopping them into "network", "security" and "attack" unigrams cause its semantic meaning to be lost.

Concept mining and extraction plays a key role in text mining where each concept is a combination of words which are real or imaginary. Those concepts are used for tasks such as document retrieval [5,6], classification [7], concept based sentiment analysis [8], semantic product search in e-commerce [9,10] and concept hierarchy learning [21]. Search engines and document databases also use such concepts to locate related information from huge text archives. Thus concept identification is a key process in leveraging potential knowledge out of the data. Understanding, analyzing and summarizing key concepts from large volumes of text data is non-trivial and crucial in knowledge discovery process.

We address the problem of interpreting statistically generated topics by inducing concepts from large document collection using probabilistic topic modeling. Different from previous studies, we attempt to extract semantically rich "concepts" directly from LDA [1] generated "topics" using a simple framework which is completely unsupervised and easy to implement. Exploration in these dimensions may enrich existing systems in better understanding of text data and can be applied to tasks such as document summarization [14], automated ontology generation [15] etc. We also show that our framework is fit for the task of searching and retrieving documents in large archives and outperforms state of the art probabilistic and phrase based mechanisms.

The main contributions of the paper are summarized as follows:

1. Proposes a novel framework for extracting semantically rich and close to real-world concepts directly from statistically generated topics using a lightweight scoring algorithm.
2. Rigorous experimental comparison of the proposed concept extraction method with state-of-the-art approaches, establishes the effectiveness of the proposed method.
3. Demonstrates the usefulness of the method in improving document retrieval performance in information retrieval tasks.

Organization: The rest of this paper is organized as follows. We review related works in Sect. 2. Section 3 briefly outline the background on probabilistic topic model and LDA algorithm we use in this paper. Section 4 specifies the research objective and a detailed explanation of our proposed topic modeling guided

concept extraction algorithm is presented in Sect. 5. In Sect. 6, we show the usefulness of our proposed method in improving the document retrieval and ranking process. We discuss our experimental setup in Sect. 7, the results and detailed evaluation is described in Sect. 8. Finally we draw conclusions and discuss future work in Sect. 9.

2 Related Work

Topic models such as Latent Dirichlet Allocation (LDA) [1] and Probabilistic Latent Semantic Analysis (PLSA) [3] come up with well established mathematical and statistical model to inspect unstructured text documents for bringing out hidden themes called "topics". Such topics are probability distributions over words in the vocabulary. In this section we evaluate related researches on the dimension of inferring concepts from text data using topic modeling and critically review those works which are closely similar to our proposed method.

The first notable work which explored beyond the traditional "bag-of-word" approach in topic modeling is the Bigram topic model [16]. In this model each topic word is generated from the distribution of words over a context which is given by a latent topic and the previous word. Later Wang et al. extended the basic Bigram topic model and proposed a new model called Topical n-gram [11] by incorporating a switching variable at each word position to denote the starting of a new n-gram. If this switching variable is not triggered, then the word will be considered as the continuation of a previously identified n-gram. The major shortfall of this model is that a post-processing is required to get the topic of a final word in a n-gram as the topic of the entire n-gram. This is because, in practical, words within an n-gram will not share same topic normally.

Identifying these shortcomings of the Topical n-gram model [11], Lindsay et al. proposed PDLDA (Phrase discovering topic model) [12] which used the hierarchical Pitman-Yor processes [17] for creating topic-word matrix. A topic segmentation model which incorporates word order [13] is later proposed by Jameel et al. Apart from topic detection, these models performs phrase segmentation which is computationally expensive when dealing with large text archives. More recent work in this dimension is reported in 2014 by El-Kishky et al. in which they proposed a topical phrase mining method called TopMine [18]. This framework uses a two step process - one for discovering phrases from text and next for training a traditional LDA model on these phrases and their assumption is that words in the same phrase should be assigned with the same topic.

Other recent work reported is a framework proposed by Yulan He for extracting topical phrases from clinical documents [19]. In this two step work, the author first extracts medical phrases using an off-the-shelf tool and then train a topic model which takes a hierarchy of Pitman-Yor processes [17] as prior for modeling the generation of phrases of arbitrary length. Other notable work in this dimension is the Hierarchical Concept Topic Model (HCTM) [20] and Concept Topic Model (CTM) [22] which incorporates concepts into probabilistic topic models. HCTM employs a manual tagging mechanism to tag concepts in a document and involves assigning concepts to each word in a document. Then the semantic

themes of a document content are revealed as a probability distribution over concepts and combine a hierarchy of human defined concepts with statistical topic models to combine the best of both. The major disadvantage of this work is that the tagging process is cumbersome and cannot be done without human annotators. A cluster based iterative topical phrase mining framework [31] was recently introduced that present a novel framework for topical phrase mining. Their approach treats corpus as a mixture of clusters and each cluster is characterized by documents sharing similar topical distributions. Then this method iteratively performs phrase mining, topical inferring and cluster updating until a satisfactorily final result is obtained. Another notable work in this dimension was introduced by Li and Jin [32]. They proposed a semantic concept latent dirichlet allocation and semantic concept hierarchical dirichlet process based approaches by representing text as meaningful concepts rather than words. The authors implemented the algorithms in discovering new semantic relation between concepts from text documents. The method improved the search quality when compared with other LDA or HDP based approaches. Another very recent work was introduced by Xu et al. [33] that incorporates Wikipedia concepts and categories as prior knowledge into topic models. Their work utilizes entity knowledge, concepts and categories in Wikipedia as prior knowledge into topic models so that it discover more coherent topics. Their method not only modeled the relationship between words and topics, but also utilizes knowledge of concept and category to model semantic relationship between them.

The method proposed in this paper introduces a framework which extracts semantically rich concepts directly from a collection of probabilistically generated topics using Latent Dirichlet Allocation (LDA) [1] algorithm. This method first generates a bag of word bi-grams and tri-grams from the static document corpus and rank them using our new scoring function. We also show that our proposed method outperforms already existing phrase based document retrieval methods for retrieving and ranking relevant documents from a large text archive.

3 Background: Latent Dirichlet Allocation (LDA)

Inspired from previous topic models, Blei et al. introduced a new topic modeling algorithm known as Latent Dirichlet Allocation (LDA) [1]. This model assumes that a document contain multiple topics and such topics are extracted using a Dirichlet Prior process. In the following section, we will briefly describe the underlying principle of LDA [1]. Even though LDA works well on broad ranges of discrete datasets, the text is considered to be a typical example to which the model can be best applied. The process of generating a document with n words by LDA can be described as follows [1]:

1. Choose the number of words, n, according to Poisson Distribution;
2. Choose the distribution over topics, θ, for this document by Dirichlet Distribution;
   (a) Choose a topic $T^{(i)} \sim$ Multinomial (θ)
   (b) Choose a word $W^{(i)}$ from $P\left(W^{(i)}|T^{(i)}, \beta\right)$

Thus the marginal distribution of the document can be obtained from the above process as:

$$P\left(d\right) = \int_{\theta} \left(\prod_{i=1}^{n} \sum_{T^{(i)}} P(W^{(i)})|T^{(i)}, \beta) P(T^{(i)}|\theta) \right) P(\theta|\alpha) d\theta \tag{1}$$

where, $P(\theta|\alpha)$ is derived by Dirichlet Distribution parameterized by α, and $P(W^{(i)})|T^{(i)}, \beta)$ is the probability of $W^{(i)}$ under topic $T^{(i)}$ parameterized by β. The parameter α can be viewed as a prior observation counting on the number of times each topic is sampled in a document, before we actually seen any word from that document. The parameter β is a hyperparamter determining the number of times words are sampled from a topic [1], before any word of the corpus is observed. At the end, the probability of the whole corpus D can be derived by taking the product of all documents' marginal probability as:

$$P(D) = \prod_{i=1}^{M} P(d_i) \tag{2}$$

4 Research Objective

The following are our main research objectives:

1. Introduce the task of topical phrase discovery from unstructured text corpus and its applications in real life scenarios.
2. Propose a framework that uses a lightweight scoring algorithm for extracting concepts from statistically generated topics using Latent Dirichlet Allocation (LDA).
3. Verify experimentally the effectiveness of the method in extracting real-world concepts. We compare our proposed algorithm with state-of-the-art phrase discovery topic models to establish the fitness of our method for concept extraction.
4. Show the usefulness of our approach in phrase based document retrieval task, where given a user query relevant documents are retrieved from a large static text archival.

5 Topic Modeling Guided Concept Extraction

In this section, we present our approach for extracting close to real-world and semantically interpretable concepts from relatively larger static document corpus. Firstly we present our proposed concept scoring function and the topic modeling guided concept extraction algorithm. Secondly we describe how our proposed method can better rank documents in a large text archival and how our method is efficient in retrieving relevant documents. Firstly, we outline our proposed scoring function that we use to score real-world concepts that are latent in large text document corpora. We introduce a scoring function where the score

of a noun-phrase n in a topic t is computed as the probability of a topic over the document multiplied by the frequency of noun-phrase in that document. Notationally,

$$score(n,t) = \sum_{document_d} p(t|d) freq(n,d) \tag{3}$$

where $p(t|d)$ is the scoring for the document-topic pair obtained from the LDA and $freq(n,d)$ is the frequency of noun-phrase n in document d. After calculating score for each noun-phrase, we filter top k noun-phrases according to the score. Now we propose our algorithm, Algorithm 1, which takes a set of topics which are probability distribution over words, generated by Latent Dirichlet Allocation (LDA) [1] and a set of static corpus of documents D. Using an off-the-shelf noun-phrase tagger, the algorithm first tags all the noun-phrases from static corpus D. The frequencies of noun-phrase in documents are then computed. The probability distribution of topic over documents, $p(t|d)$ is computed from LDA output. Then the score of already tagged noun-phrases in documents, $score(n,t)$ calculated as the product of $p(t|d)$ and $freq(n,d)$. Finally, we find top - k noun-phrases according to the score, $score(n,t)$ calculated using Eq. 3.

Algorithm 1. Algorithm for topic modeling guided concept extraction

function TM-ExtractConcepts (D,T);
Input : A static document collection, D and a topic collection, T
Output: top-k phrases for each topic
for *each document d from D* **do**
| run through a noun-phrase tagger, and collect noun-phrases
end
for *topic t and noun-phrase n* **do**
| score(n,t) = $\sum_{document_d} p(t|d) freq(n,d)$
end
for each topic t, find the top-k noun-phrases according to the $score(n,t)$

6 Topic Modeling Guided Document Retrieval

Given a multi-word query Q, computing the top k relevant documents from a large collection of documents D is a fundamental problem in Information Retrieval (IR). The relevance between a given query and a document d is determined by using similarity functions such as BM25 [24] and such functions generate scores for every document d in the collection D. Then the system reports k top scored documents as the ranked result. Earlier, vector based models were introduced in which documents are represented as a vector of terms, $\vec{d_j} = w_{1j}, w_{2j}, ...w_{tj}$ where t is the total number of words in the document and each $w_{1j} > 0$ if and only if the word i is present in document d_j. In this model, the term weights are not binary as in boolean model since vector space model do

not consider the presence or absence of terms. In the same fashion, query is also represented as vector and the similarity between a query vector and a document vector is calculated for measuring the relevance and the same is used as a ranking score. In vector space model, there are two widely used factors for calculating term weights; namely, term frequency (tf) and inverse document frequency (idf). The term weight is computed by multiplying these two factors and widely known as $tf - idf$ measure [23]. Then the similarity measure between the query vector and document vector is calculated using widely used similarity measures such as Okapi BM25 [24] as:

$$score(q,d) = \sum_{i=1}^{|q|} idf(q_i).\frac{tf(q_i,d).(k_1+1)}{tf(q_i,d)+k_1.(1-b+b.\frac{|d|}{avgdl})} \tag{4}$$

where, $tf(q_i, d)$ is q_i's term frequency in the document d, $|d|$ is the length of the document d in words, and $avgdl$ is the average document length in the text collection from which documents are drawn. When dealing with large collection of documents, calculating tf, itf and then computing the similarity measures between query and document vectors are often difficult.

Here, we propose a new method for retrieving top - k documents from a large collection of text corpus using [concept - topic - document] triplet where concepts are extracted using the scoring function we have introduced in Sect. 6. Given a query, we first find relevant topics using the function score(n, t) = $\sum_{document_d} p(t|d)freq(n, d)$ where $p(t|d)$ is the probability of topic over document and $freq(n, d)$ is the frequency of phrase over document, which are precomputed. Using topic distribution over documents information given by LDA, we then obtain a set of documents having highest topic probability and present top-k documents as the ranked collection.

Algorithm 2. Algorithm for topic modeling guided document retrieval

function TM-RetrieveDocuments(q);
Input : A phrase query q
Output: Ranked set of documents
retrieve the significant topics for q using scoring function, score(q,t), where score(q,t) = $\sum_{document_d} p(t|d)freq(q, d)$
use score(q,t), to get topic - document distribution (given by LDA)
list top - k documents according to the distribution

7 Experimental Setup

7.1 Dataset Description

We use three different datasets in our experimental evaluation - the BBC dataset, StackExchange dataset and a much larger Reuters 21578 dataset. A short description of these datasets are given below.

- **BBC News Dataset:**[1] The general BBC dataset consist of 2225 text documents directly from their website corresponding to stories in five areas such as business, entertainment, politics, sports and technology, from 2004 to 2005.
- **StackExchange Dataset:**[2] This dataset comprise of anonymized, user-contributed contents on the StackExchange website in different categories. Since our proposed approach uses a crowd-sourcing experiment we have carefully chosen categories in such a way that it matches with the general knowledge of the users participating. Specifically we have chosen academia, android, cooking, gaming and travel categories for this experiment. All post under a single thread is merged in one document and thus the final dataset consists of 2769 documents.
- **Reuters 21578 Dataset:**[3] Reuters 21578 dataset is a collection of newswire articles and is popular among data mining communities. The collection was made available for research purposes by Reuters in 1990.

7.2 Experimental Testbed for Topic Induced Concept Extraction

This section describes the experimental setup we have used for our proposed concept extraction experiment. All methods described in this paper were implemented in Python 2.7. The experiments were run on a server configured with AMD Opteron 6376 @ 2.3 GHz/16 core processor and 16 GB of main memory. Firstly, we pre-processed each document for removing common words such as "the", "and" etc. We then used TextBlob [27] library in Python for tagging noun-phrases from each document. We considered noun-phrases that contain at least two words and for all datasets, we have filtered noun-phrases that are word n-grams where n $\geq$ 2. We then modeled topics using LDA algorithm and the score of each noun-phrase is calculated using Eq. 3 and selected top -k noun-phrases for each topic. We use MALLET[4] implementation of the LDA model to generate topics and the number of iterations in Gibbs sampling used in this paper is set as 300, as we find that Gibbs sampling usually approaches the target distribution after 300 rounds of iterations. The parameters α and β are set as $\alpha = 50/Z$ and $\beta = 0.01$, respectively.

Baselines - We compare our proposed concept extraction approach with the following baselines. For all the hyper-parameters, we use the default settings and optimizes these parameters every 100 Gibbs sampling iterations.

- **TNG** [11] - The topical n-gram model automatically decides whether to form an n-gram or not by considering its surrounding text. The TNG model provides a systematic way to model topical phrases and simultaneously detect n-grams and topics. The MALLET implementation of the TNG is used with all the default settings for hyperparameters.

[1] http://mlg.ucd.ie/datasets/bbc.html.
[2] https://archive.org/details/stackexchange.
[3] http://www.daviddlewis.com/resources/testcollections/reuters21578/.
[4] http://mallet.cs.umass.edu/.

– **TopMine** [18] - This algorithm first extracts phrases using a method similar to frequent pattern mining and then train a modified LDA model on the "bag-of phrases" input.

7.3 Experimental Testbed for Document Retrieval

In this section, we describe the experimental setup used for evaluating document retrieval using our proposed framework. We compare our newly devised model to BM25 [24] which is a term-matching based baseline and two phrase based retrieval baselines that uses a flat position index [28] and inverted index [29] as their data structure. For the BM25 setup, we have used the parameters $k1 = 1$, $k2 = 0$, $k3 = 1$ and $b = 0.5$ for initial experiment and later finalized the values of $k1$ and b as 1.7 and 0.95 respectively as these values gave optimum retrieval results for the current experiment. We implemented the inverted index and flat position index using Python 2.7 version and used public code libraries at https://www.rosettacode.org/wiki/Inverted_index_Python and https://github.com/matteobertozzi/blog-code/blob/master/py-inverted-index/ for developing some of the components of the indexes. Our experiments were run on a server configured with AMD Opteron 6376 with 2.3 GHz with 16 core processor and 16 GB of main memory.

8 Result and Evaluation

8.1 Qualitative Evaluation

For measuring closeness of the concepts generated using our proposed method with the real-world concepts, we have conducted a crowd-sourcing experiment which is widely used in text mining and processing tasks to validate and verify the computer generated outputs with human annotated contents. Crowdsourcing has been applied to tasks such automated question-answering systems [25] and ontology alignment [26]. For this experiment, we have created a web interface that present the users with the concepts extracted by our algorithm and asked them to positively reward every concept which seems to be semantically valid. We used Fleiss Kappa score [30] for the inter-annotator agreement. Measures such as $precision(P)$, $recall(R)$ and $F - measure(F)$ is used for computing the quality of extracted concepts to the human identified concepts. Here, true positive is defined as the number of overlapped concepts between human authored concepts and concepts generated by the algorithm, false positive is the number of extracted concepts that are not truly human authored concepts and false negative is the human authored concepts that are missed by the concept extraction method. Our precision, recall and f-measure values are shown in Table 1 and from the values we draw the following conclusions. We have got precision values ranging from 90.12% to 94.58%. This shows our method extracts quality concepts which are close to real world when compared with the results of a crowd-sourcing experiment (Fig. 1).

Table 1. Comparison of proposed topic modeling guided concept extraction method with Topical N-gram, TopMine baselines. Proposed method outperforms significantly better than all the baselines on all datasets.

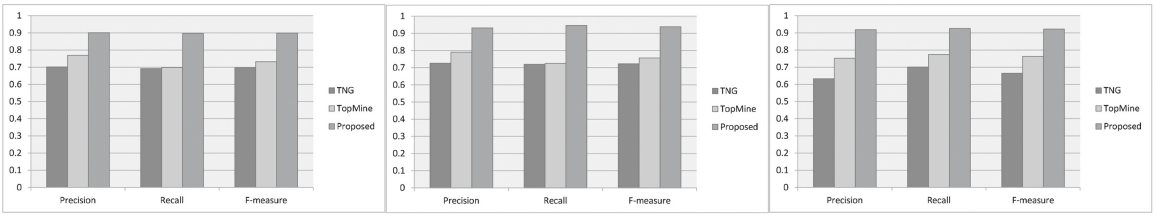
Method	Dataset								
	Reuters			BBC			StackExchange		
	P	R	F	P	R	F	P	R	F
TNG [11]	0.7015	0.6922	0.6968	0.7259	0.7199	0.7228	0.6325	0.7011	0.6650
TopMine [18]	0.7692	0.6988	0.7323	0.7903	0.7254	0.7564	0.7521	0.7741	0.7629
Proposed	**0.9012**	**0.8966**	**0.8988**	**0.9318**	**0.9458**	**0.9387**	**0.9187**	**0.9258**	**0.9222**

Table 2. Comparison of proposed document retrieval method with BM25, flat position index and inverted index baselines.

Method	Dataset								
	Reuters			BBC			StackExchange		
	P	R	F	P	R	F	P	R	F
BM25 [24]	0.6432	0.5920	0.6165	0.5177	0.4852	0.5009	0.6129	0.6233	0.6180
FPI [28]	0.7012	0.6903	0.6957	0.6911	0.7230	0.7066	0.6709	0.6237	0.6464
Inv. idx [29]	0.8126	0.8701	0.8403	0.7752	0.7659	0.7705	**0.8129**	**0.8788**	**0.8445**
Proposed	**0.9127**	**0.9201**	**0.9163**	**0.9322**	**0.9412**	**0.9366**	0.8099	0.8123	0.8110

8.2 Evaluation of Document Retrieval

Here we evaluate our proposed document retrieval method and compares it with the baselines such as [24,28,29]. Table 2 shows the performance comparison in terms of precision, recall and f-measure. From the table it is evident that our proposed method significantly outperforms all the three baselines on two out of three datasets we have chosen. The recorded precision values for our proposed method are ranging from 91.27% to 94.12% for BBC and Reuters dataset respectively. For the StackExchange dataset, we have noted that our closest competitor [29] achieved better precision than the proposed method. This is due to comparatively less number of concepts extracted for different categories we



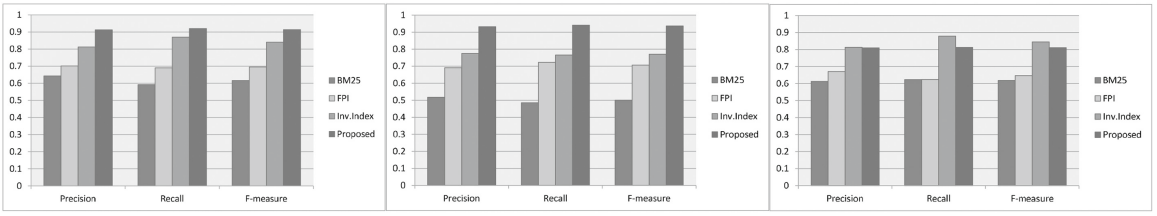
(a) Precision, Recall and F-measure comparison on Reuters dataset

(b) Precision, Recall and F-measure comparison on BBC dataset

(c) Precision, Recall and F-measure comparison on StackExchange dataset

Fig. 1. Precision, Recall and F-measure values of concept extraction when compared with baseline methods on Reuters, BBC and StackExchange datasets

have considered for the StackExchange dataset such as academia, android and cooking. We found that the noun-phrase tagger was unable to tag phrases from those less popular categories (Fig. 2).



(a) Precision, Recall and F-measure comparison on Reuters dataset

(b) Precision, Recall and F-measure comparison on BBC dataset

(c) Precision, Recall and F-measure comparison on StackExchange dataset

Fig. 2. Precision, Recall and F-measure values obtained for document retrieval when compared with baseline methods on Reuters, BBC and StackExchange datasets

9 Conclusions and Future Work

In this paper, we proposed a novel framework for leveraging semantically rich concepts from statistically computed topics using a widely used LDA algorithm. Rigorous experiments with three real-world datasets show that our proposed method produce close to the real-world concepts which we verifies with a crowdsourcing experiment in terms of precision, recall and f-measure. We also demonstrate the usage of our framework on document retrieval and ranking which is a well studied area in information retrieval. The experimental results show that our method can significantly outperform phrase based document retrieval baselines. There are further scopes for future work. By constantly improving the quality of concepts by tuning the parameters of our proposed algorithm, the extracted concepts can be used for building ontologies by exploiting the relations between them. Concept hierarchy learning can also be done using this and we plan to extend our framework further in these dimensions.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
2. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 50–57. ACM (1999)
3. Hofmann, T.: Probabilistic latent semantic analysis. In: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, pp. 289–296. Morgan Kaufmann Publishers Inc. (1999)
4. Chang, J., Boyd-Graber, J.L., Gerrish, S., Wang, C., Blei, D.M.: Reading tea leaves: how humans interpret topic models. In: NIPS, vol. 31, pp. 1–9 (2009)
5. Arvanitis, A., Wiley, M. T., Hristidis, V.: Efficient Concept-based Document Ranking. In: EDBT, pp. 403–414 (2014)

6. Egozi, O., Gabrilovich, E., Markovitch, S.: Concept-based feature generation and selection for information retrieval. In: AAAI, pp. 1132–1137 (2008)
7. Celikyilmaz, A., Hakkani-Tr, D.: Concept-based classification for multi-document summarization. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5540–5543 (2011)
8. Cambria, E.: An introduction to concept-level sentiment analysis. In: MICAI, vol. 2, pp. 478–483 (2013)
9. Asharaf, S., Anoop, V.S., Afzal, A.L.: A framework for meaning aware product discovery in e-commerce. In: Encyclopedia of e-Commerce Development, Implementation, and Management, pp. 1386–1398. IGI Global (2016)
10. Anoop, V.S., Asharaf, S.: A topic modeling guided approach for semantic knowledge discovery in e-commerce. Int. J. Interact. Multimedia Artif. Intell. **4**, 1–8 (2017)
11. Wang, X., McCallum, A., Wei, X.: Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In: Proceedings of Seventh IEEE International Conference on Data Mining, ICDM 2007, pp. 697–702 (2007)
12. Lindsey, R.V., Headden III, W.P., Stipicevic, M.J.: A phrase-discovering topic model using hierarchical pitman-yor processes. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 214–222 (2012)
13. Jameel, S., Lam, W.: An unsupervised topic segmentation model incorporating word order. In: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 203–212 (2013)
14. Yang, G., Wen, D., Chen, N.S., Sutinen, E.: A novel contextual topic model for multi-document summarization. Exp. Syst. Appl. **42**(3), 1340–1352 (2015)
15. Sleeman, J., Finin, T., Joshi, A.: Topic modeling for RDF graphs. In: LD4IE@ ISWC, pp. 48–62 (2015)
16. Wallach, H.M.: Topic modeling: beyond bag-of-words. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 977–984 (2006)
17. Pitman, J., Yor, M.: The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. Ann. Probab. **25**(2), 855–900 (1997)
18. El-Kishky, A., Song, Y., Wang, C., Voss, C.R., Han, J.: Scalable topical phrase mining from text corpora. Proc. VLDB Endow. **8**(3), 305–316 (2014)
19. He, Y.: Extracting topical phrases from clinical documents. In: AAAI, pp. 2957–2963 (2016)
20. Chemudugunta, C., Smyth, P., Steyvers, M.: Combining concept hierarchies and statistical topic models. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 1469–1470 (2008)
21. Anoop, V.S., Asharaf, S., Deepak, P.: Unsupervised concept hierarchy learning: a topic modeling guided approach. Procedia Comput. Sci. **89**, 386–394 (2016)
22. Chemudugunta, C., Holloway, A., Smyth, P., Steyvers, M.: Modeling documents by combining semantic concepts with unsupervised statistical learning. In: International Semantic Web Conference, pp. 229–244 (2008)
23. Ramos, J.: Using tf-idf to determine word relevance in document queries. In: Proceedings of the First Instructional Conference on Machine Learning (2003)
24. Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M.M., Gatford, M.: Okapi at TREC-3, p. 109. NIST Special Publication, Gaithersburg (1995)
25. Mrozinski, J., Whittaker, E., Furui, S.: Collecting a why-question corpus for development and evaluation of an automatic QA-system. In: 46th Annual Meeting of the Association of Computational Linguistics: Human Language Technologies, pp. 443–451 (2008)

26. Sarasua, C., Simperl, E., Noy, N.F.: Crowdmap: Crowdsourcing ontology alignment with microtasks. In: International Semantic Web Conference, pp. 525–541 (2012)
27. Loria, S.: TextBlob: simplified text processing (2014)
28. Shan, D., Zhao, W.X., He, J., Yan, R., Yan, H., Li, X.: Efficient phrase querying with flat position index. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, pp. 2001–2004 (2011)
29. Patil, M., Thankachan, S.V., Shah, R., Hon, W.K., Vitter, J.S., Chandrasekaran, S.: Inverted indexes for phrases and strings. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 555–564 (2011)
30. Fleiss, J.L., Cohen, J.: The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. Educ. Psychol. Measur. **33**(3), 613–619 (1973)
31. Li, B., Wang, B., Zhou, R., Yang, X., Liu, C.: CITPM: a cluster-based iterative topical phrase mining framework. In: International Conference on Database Systems for Advanced Applications, pp. 197–213 (2016)
32. Li, X., Jin, W.: Cross-document knowledge discovery using semantic concept topic model. In: 15th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 108–114 (2016)
33. Xu, K., Qi, G., Huang, J., Wu, T.: Incorporating Wikipedia concepts and categories as prior knowledge into topic models. Intell. Data Anal. **21**(2), 443–461 (2017)

International Journal of Information Processing, 10(3), 1-11, 2016
ISSN : 0973-8215
IK International Publishing House Pvt. Ltd., New Delhi, India

Learning Concept Hierarchies through Probabilistic Topic Modeling

V S Anoop[a], S Asharaf[b], Deepak P[c]

[a]Data Engineering Lab, Indian Institute of Information Technology and Management - Kerala (IIITM-K), Thiruvananthapuram 695 581, India, Contact: anoop.res15@iiitmk.ac.in

[b]Indian Institute of Information Technology and Management - Kerala (IIITM-K), Thiruvananthapuram 695 581, India, Contact: asharaf.s@iiitmk.ac.in

[c]Queen's University, Belfast, UK, Contact: deepaksp@acm.org

With the advent of semantic web, various tools and techniques have been introduced for presenting and organizing knowledge. Concept hierarchies are one such technique which gained significant attention due to its usefulness in creating domain ontologies that are considered as an integral part of semantic web. Automated concept hierarchy learning algorithms focus on extracting relevant concepts from unstructured text corpus and connect them together by identifying some potential relations exist between them. In this paper, we propose a novel approach for identifying relevant concepts from plain text and then learns hierarchy of concepts by exploiting subsumption relation between them. To start with, we model topics using a probabilistic topic model and then make use of some lightweight linguistic process to extract semantically rich concepts. Then we connect concepts by identifying an "is-a" relationship between pair of concepts. The proposed method is completely unsupervised and there is no need for a domain specific training corpus for concept extraction and learning. Experiments on large and real-world text corpora such as BBC News dataset and Reuters News corpus shows that the proposed method outperforms some of the existing methods for concept extraction and efficient concept hierarchy learning is possible if the overall task is guided by a probabilistic topic modeling algorithm.

Keywords : Concept Extraction, Natural Language Processing, Probabilistic Topic Models, Semantic Web, Subsumption Hierarchy Learning, Text Mining.

1. INTRODUCTION

Due to rapid growth of text producing and consuming applications, numerous tools and techniques were introduced in the recent past for extracting useful patterns from unstructured text. These patterns are crucial for organizations to discover knowledge out of it and aid in making intelligent decisions. As the amount of such data grows exponentially, already available algorithms performs poor on the scalability and performance aspects. But there are still a lot of avenues where text data is yet to be exploited fully and thus we need new and efficient algorithms to tackle this situation. Platforms such as social networks, e-commerce websites, blogs and research journals generate such data in the form of unstructured text and it is essential to analyze, synthesis and process such data for efficient retrieval of useful information.

In text mining, concepts are defined as a sequence of words that constitute real or imaginary entities. Extraction of such entities are non-trivial for applications such as automated ontology generation [1], document summarization [2] and aspect oriented sentiment analysis [3] to name a few. This is the era of data explosion thus it is very difficult to store, process, manage and most importantly to extract knowledge out of it. To overcome this shortfall, a significant amount of research has been carried out in the recent past for leveraging underlying thematic and semantic structure from

text archives. As a result a good number of algorithmic techniques were introduced which are proved to be efficient for the discovery of themes and semantics underlying high dimensional data.

Topic Models are suite of text understanding algorithms which statistically generate latent themes pervade a large collection of unstructured text. Since its inception, text mining researchers and practitioners are using it extensively to analyze and organize large document collections. They are unsupervised learning algorithms thus it does not require user tagged corpus to work with. A large number of topic modeling algorithms have been reported in the past with the difference in the assumption they make for modeling topics. Models such as Probabilistic topic models [4] and Latent Dirichlet Allocation (LDA) are some such flavors of topic modeling that attained significant attention.

Contributions: This work proposes a novel unsupervised approach for learning concept hierarchies from large unstructured text corpus which is guided by a probabilistic topic modeling approach. To begin with, we model topics from the corpus using Latent Dirichlet Allocation (LDA) algorithm and then uses a lightweight linguistic process to identify concepts which are close to the real world understanding. Then we make use of a subsumption relation [5] ("is-a") to connect concepts which are related thus forms a hierarchy of concepts.

Organization: The rest of this paper is organized as follows. We briefly review related works in Section 2. Section 3 introduces the novel approach we have proposed. Detailed explanation of the implementation details is presented in Section 4 and the evaluation of the proposed method is discussed in Section 5 and finally, we draw a conclusion and discuss future work in Section 6.

2. PROBLEM DEFINITION AND RELATED WORK

2.1. Problem Definition

Here, we define the problem formally. Given a large corpus containing unstructured text documents, our problem is to automatically generate concept hierarchies which are close to human understanding. In a nutshell, this paper aims to answer the following research questions :

1. Is it possible to automatically extract human interpretable concepts from statistically generated topics using a lightweight linguistic process ?
2. Can our proposed method learn a hierarchy of such concepts incorporating a subsumption relation between them, which are important in automated ontology generation ?
3. Given a large but unstructured text corpus, can our topic modeling guided method better extracts and learns concept hierarchies compared to existing algorithms ?

Many recent works have been reported in this direction which proposed many algorithms to extract semantically rich concepts from plain text. In the following section, we due acknowledge some past literatures that discusses methods which are close to our proposed algorithm.
Notations used in this paper: To help narrative, some commonly used notations are shown in Table 1 that are used in the rest of this paper.

2.2. Related Work

Concept extraction is the process of extracting real or imaginary entities from plain text that has got wider recognition in the recent past. This is due to the wide variety of applications which are mainly dealing with text data such as e-commerce websites, research articles etc. Thus a significant number of research literatures are available in the field of concept extraction and mining which proposes many algorithms with varying degrees of success. In

Table 1
Notations used in subsequent sections

Notation	Meaning
tf	term frequency
itf	inverse topic frequency
N_{tf}	normalized term frequency
tc	topic collection
C_{td}	topic-document cluster
MW	muti-word
MW_c	multi-word collection

this section, we give emphasis on past literatures in automated concept extraction and hierarchy learning algorithms and briefly discuss works closely related to our proposed framework.

Phrase discovering topic model [6] that uses pitman-yor process and TopMine [7] were two notable works that proposed algorithms for mining topical phrases from text documents. The former constructs a topic-word matrix before modeling topics but disadvantage of the approach was that creating such a matrix for large volume of text is often difficult. The latter approach makes use of a two stage process for modeling topics and mainly works with clinical documents. First it identifies phrases using some off-the-shelf tools and then trains a topic model with the identified phrases. Another work which uses topic models for generating multi-word phrases was the topical n-gram [8]. This makes use of some switching variable for identifying a new n-gram. The assumption of this method was that the words within an n-gram usually won't share same topic, which may not be true all the time.

Automatic Concept Extractor (ACE), a system specifically designed for extracting concepts from HTML pages and making use of the text body and some visual clues on HTML tags for identifying potential concepts was proposed by Ramirez and Mattmann [9]. Even though this method could outperform some state of the art methods, dependency with HTML was a major drawback. Turney[10] proposed another system named GenEx, which employed a genetic algorithm supported rule learning mechanism for concept extraction.

A system which extracts concepts from user tag and query log dataset is proposed by Parameswaran *et al.*,[11] which uses techniques similar to association rule mining. This method uses features such as frequency of occurrences and the popularity among users for extracting core concepts and attempts to build a web of concepts. Even though this algorithm can be applied to any large dataset, a lot of additional processing is required when dealing with web pages.

A bag-of-word approach was proposed by Gelfand *et al.*,[12] for concept extraction from plain text and used these to form a closely tied semantic relations graph for representing relationships between them. They have applied this technique specifically for some classification tasks and found that their method produces better concepts than the Naive Bayes text classifier.

Dheeraj Rajagopal *et al.*,[13] introduced another graph based approach for commonsense concept extraction and detection of semantic similarity among those concepts. They used a manually labeled dataset of 200 multiword concept pairs for evaluating their parser capable of detecting semantic similarity and showed that their method was capable of effectively finding syntactically and semantically related concepts. The main disadvantage of that method is the use of manually labeled dataset and the creation of such dataset is time consuming and requires human effort. Another work reported in this domain is the method proposed by Krulwich and Burkey [14] which uses a simple heuristics rule based approach to extract key phrases from document by considering visual clues such as the usage of bold and italic characters as features. They have shown that this technique can be extended for automatic document classification experiments.

A key phrase extraction system called Automatic Keyphrase Extraction (KEA) developed by Witten *et al.*,[15] was reported in the concept extraction literatures which creates a

Naive Bayes learning model with known key phrases extracted from training documents and uses this model for inferring key phrases from new set of documents. As an extension to this KEA framework, Song *et al.,*[16] proposed a method which uses the information gain measure for ranking candidate key phrases based on some distance and tf-idf features which was first introduced in [15]. Another impressive and widely used method was introduced by Frantzi *et al.,*[17] which extracts multi-word terms from medical documents and named as C/NC method. The algorithm uses a POS tagger POS patten filter for collecting noun phrases and then uses some statistical measures for determining the termhood of candidate multi-words.

The proposed method in this paper is a hybrid approach incorporating statistical methods such as topic modeling and tf-itf weighting and some lightweight linguistic processes such as POS tagging and analysis for leveraging concepts from text. We expect the learnt concept hierarchy to be close to the real world understanding of concepts which we will quantify using evaluation measures such as precision, recall and f-measure.

3. BACKGROUND: LATENT DIRICHLET ALLOCATION (LDA)

A good number of topic modeling algorithms are introduced in the recent past which varies in their method of working mainly with the assumptions they adopt for the statistical processing. An automated document indexing method based on a latent class model for factor analysis of count data in the latent semantic space has been introduced by Thomas Hofman [18]. This generative data model called Probabilistic Latent Semantic Indexing (PLSI), considered as an alternative to the basic Latent Semantic Indexing has a strong statistical foundation. The basic assumption of PLSI is that each word in a document corresponds to only one topic.

Later, Blei *et al.,*[19] introduced a new topic modeling algorithm known as Latent Dirichlet Allocation (LDA) which is more efficient and attractive than PLSI. This model assumes that a document contain multiple topics and such topics are leveraged using a Dirichlet Prior process. In the following section, we will briefly describe the underlying principle of LDA. Even though a LDA works well on broad ranges of discrete datasets, the text is considered to be a typical example to which the model can be best applied. The process of generating a document with n words by LDA can be described as follows[19]:

1. Choose the number of words, n, according to Poisson Distribution;
2. Choose the distribution over topics, θ, for this document by Dirichlet Distribution;
   (a) Choose a topic $T^{(i)} \sim$ Multinomial(θ)
   (b) Choose a word $W^{(i)}$ from $P\left(W^{(i)}|T^{(i)}, \beta\right)$

Thus the marginal distribution of the document can be obtained from the above process as :

$$P\left(d\right) = \int_{\theta} \left(\prod_{i=1}^{n} \sum_{T^{(i)}} P(W^{(i)}|T^{(i)}, \beta) P(T^{(i)}|\theta) \right) + P(\theta|\alpha) d\theta$$

where, $P(\theta|\alpha)$ is derived by Dirichlet Distribution parameterized by α, and $P(W^{(i)})|T^{(i)}, \beta)$ is the probability of $W^{(i)}$ under topic $T^{(i)}$ parameterized by β. The parameter α can be viewed as a prior observation counting on the number of times each topic is sampled in a document, before we actually seen any word from that document. The parameter β is a hyperparameter determining the number of times words are sampled from a topic [19], before any word of the corpus is observed. At the end, the probability of the whole corpus D can be derived by taking the product of all documents' marginal probability as given below:

$$P(D) = \prod_{i=1}^{M} P(d_i) \quad (1)$$

4. PROPOSED APPROACH

In the area of text mining, topic models or specifically probabilistic topic models are suite of algorithms which got wider recognition for its ability to leverage hidden thematic information from huge archives of text data. Text mining researchers are making use of topic modeling algorithms such as Latent Semantic Analysis (LSA) [20], Probabilistic Latent Semantic Indexing (PLSI) [21], Latent Dirichlet Allocation (LDA) [22] etc extensively for bringing out the themes or so called "topics" from high dimensional unstructured data.

Among all these algorithms, LDA has got lot of attention in the recent past and is widely using because of its easiness of implementation and potential applications. Even though the power of LDA algorithm has been extensively used for leveraging topics, very few studies have been reported for mapping these statistically outputted topics to semantically rich concepts. Our proposed framework is an attempt to address this issue by making use of LDA algorithm to generate topics and we leverage concepts from such topics by using a new statistical weighting scheme and some lightweight linguistic processes. The overall work flow of the proposed approach is depicted in Figure 1.

Our framework can be divided into 2 modules (i) concept extraction and (ii) concept hierarchy learning. The concept extraction module extract concepts from topics generated by LDA algorithm and the concept hierarchy learning module learns a hierarchy of extracted concepts by inducing a subsumption hierarchy learning algorithm. Detailed explanation of these modules are given below.

4.1. Concept Extraction

In this module, we introduce a topic to concept mapping procedure for leveraging potential concepts from statistically computed topics which are generated by the LDA algorithm. The first step of the proposed framework deals with the preprocessing of data which is meant for removing unwanted and irrelevant data and noises. Latent Dirichlet Allocation algorithm is executed on top of this preprocessed data which in turn generate topics through the statistical process. A total of 50 topics have been extracted by tuning the parameters of LDA algorithm. Once we got the sufficient topics for the experiment, for each topic, we have created a topic - document cluster by grouping the documents which generated such a topic and the same process has been executed for all topics under consideration.

Now, we introduce a new weighting scheme called $tf - itf$ (term frequency - inverse topic frequency) which is used for finding out highly contributing topic word in each topic. We bring this weighting scheme to filter out the relevant candidate topic words. Term frequency (tf) is the total number of times that particular topic word comes in the topic - document clusters. Normalized term frequency, N_{tf} of a topic word T_w can be calculated as:

$$N_{tf} = \frac{count(T_w)\ in\ C_{td}}{count(total\ terms\ in\ C_{td})} \tag{2}$$

Inverse topic frequency I_{tf} is calculated as:

$$I_{tf} = \frac{count(total\ terms\ in\ C_{td})}{count(documents\ with\ T_w)} \tag{3}$$

$tf - itf$ is calculated using the following equation:

$$tf - itf = N_{tf} * I_{tf} \tag{4}$$

This step is followed by a sentence extraction process in which all the sentences which contain the topic words which have high tf-itf weight are extracted. Next, we apply a parts of speech tagging on these sentences and extract only noun and adjective tags as we are only concentrating on the extraction of concepts. In linguistic pre-processing step, we take Noun + Noun, Noun + Adjective and (Adjective / Noun) + Noun combinations of words from the tagged collection. Concept identification is the last step in the process flow in which we find out the term count of all the combinations of Noun + Noun, Noun + Adjective and

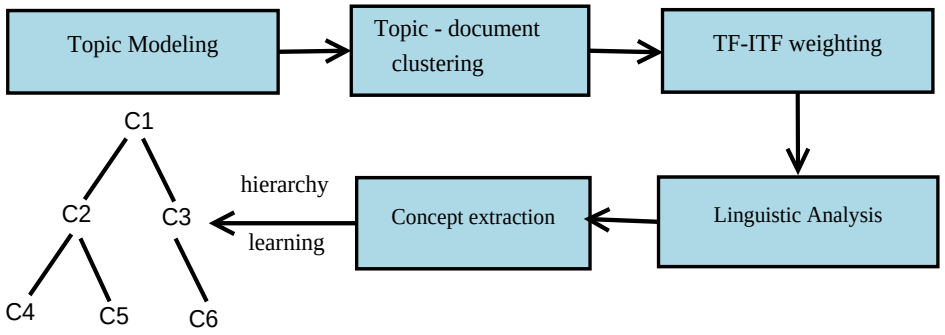


Figure 1. Workflow of the proposed framework, where C1, C2, C3, C4, C5 and C6 represent concepts

(Adjective / Noun) + Noun. A positive term count implies that the current multi word can be a potential "concept" and if we get a zero term count, then that multi word can be ignored. The newly proposed algorithm for extracting the concepts is shown in Algorithm 1.

Algorithm 1 Concept Extraction

procedure EXTRACTCONCEPTS(tc)
  $\forall\ t$, create C_{td}
  $\forall C_{td}$, compute $tf - itf$ weight
  $\forall\ t$, choose n words with highest $tf-itf$
  $S[\,] =$ sentences with top $tf-itf$ words
  $POS\ tag(S)$
  $W[\,] = (NNP, NNS, NN, JJ)$
  $MW_c[\,] = noun + noun|adj + noun$
  while $|MW_c| \neq 0$ **do**
    $termCount(MW)\ \forall\ MW$ in MW_c
    if $T_c > 0$ **then**
      Add MW into C
      Remove MW from MW_c
      Fetch next MW from MW_c
    else
      Remove MW from MW_c
      Fetch next MW from MW_c
    end if
  end while
end procedure

4.2. Concept Hierarchy Learning

In this module we derive hierarchical organization of leveraged concepts using a type of co-occurrence called "subsumption" relation. Subsumption relation is found to be simple but very effective way of inferring relationships between words and phrases without using any training data or clustering methods. The basic idea behind subsumption relation is very simple : for any two concepts C_a and C_b, C_a is said to be subsume C_b if 2 conditions hold. $P(C_a|C_b) = 1$ and $P(C_b|C_a) < 1$. To be more specific, C_a subsumes C_b if the documents which C_b occurs in are a subset of the documents which C_a occurs in. Because C_a subsumes C_b and because it is more frequent, in the hierarchy, C_a is the parent of C_b.

Algorithm 2 Concept Hierarchy Learning

procedure LEARNHIERARCHY(C)
  Choose pair of concepts, say C_a and C_b
  Compute $P(C_a|C_b)$ and $P(C_b|C_a)$
  if $P(C_a|C_b) = 1$ and $P(C_b|C_a) < 1$ **then**
    Assign C_a as the parent of C_b
  else
    Fetch next concept pairs
  end if
  Goto step 2, repeat $\forall\ C_a, C_b$
end procedure

5. EXPERIMENTAL SETUP

This section concentrates on the implementation details of our proposed framework and concept extraction and hierarchy learning procedures are discussed in detail.

5.1. Concept Extraction

Here, concept extraction module of the framework is discussed. This module concentrates on tasks such as data collection and pre-processing, topic modeling, topic-document clustering, tf-itf weighting, sentence extraction and POS tagging, linguistic pre-processing etc for identifying concepts and a detailed explanation of each step is given below.

5.1.1. Dataset Collection and Pre-processing

We are using publicly available datasets such as Reuters Corpus Volume 1 dataset [24] and BBC News Dataset [25] for the experiment. Reuters is the world's biggest international news agency and cater different news and related information through their website, video, interactive television and mobile platforms. Reuters Corpus Volume 1 is in XML format and is freely available for research purpose. Text messages are extracted by a thorough pre-processing such as removing XML tags, URLs and other special symbols and then created a new dataset exclusively for our experiment. BBC provides two benchmarked news article datasets which is freely available for machine learning research. The general BBC dataset consist of 2225 text documents directly from their website corresponding to stories in five areas such as business, entertainment, politics, sports and technology, from 2004 to 2005. A thorough pre-processing such as stemming, and removal of stop-word, URLs and special characters on this dataset and made an experiment ready copy of the original dataset.

5.1.2. Topic Modeling

Latent Dirichlet Allocation (LDA) algorithm has been applied on the pre-processed dataset to leverage topics for this experiment. The number of iterations is set to 300 as Gibbs sampling method usually approaches the target distribution after 300 iterations. The number of topics is set to 50 and a snapshot of 5 topics we have randomly chosen is shown in Table 2.

Table 2
Top 10 topic words from first 4 topics along with their TF-ITF weight

Topic 1	**Topic 3**
web [0.0048]	set [0.0047]
search [0.0048]	software [0.0032]
online [0.0047]	virus [0.0028]
news [0.0046]	users [0.0027]
google [0.0033]	firms [0.0025]
people [0.0032]	microsoft [0.0025]
information [0.0032]	security [0.0022]
internet [0.0029]	windows [0.0022]
website [0.0027]	file [0.0013]
users [0.0020]	programs [0.0011]
Topic 2	**Topic 4**
system [0.0064]	site [0.0042]
music [0.0045]	net [0.0038]
devices [0.0043]	spam [0.0035]
players [0.0035]	mail [0.0028]
media [0.0032]	firm [0.0027]
digital [0.0027]	data [0.0024]
market [0.0024]	attacks [0.0019]
technology [0.0022]	network [0.0018]
consumer [0.0021]	web [0.0016]
technologies [0.0018]	research [0.0014]

5.1.3. Topic - Document Clustering

In this step, we consider each topic and then grouped and clustered top 50 documents which contributed the creation of that specific topic. This has been done for all the 50 topics of our choice. As an outcome, we have got 50 such clusters that contain documents which generated the topics.

5.1.4. TF-ITF Weighting

Here, we compute the $tf-itf$ (term frequency-inverse topic frequency) weight of each word in every topic using Eq. (3), Eq. (4) and Eq. (5) to find out highly used topic words in the collection. Table 2 also shows topic words along with their tf-itf weight.

Table 3
Top concepts extracted against first 4 topics

Concepts_Topic 1	**Concepts_Topic 3**
web search	music players
search engine	digital media
google news	digital technology
online news search	consumer devices
google search engine	market system
Concepts_Topic 2	**Concepts_Topic 4**
software users	spam mail
virus programs	spam website
windows security	network research
software forms	research firm
microsoft programs	website attacks

5.1.5. Sentence Extraction and POS Tagging

In sentence extraction step, we consider topic words having highest tf-itf weight and then extract sentences containing these topic words from the topic - document clusters. Then a parts of speech tagging has been done to identify words tagged as nouns and adjectives from these sentences as our aim is to extract potential "concepts" from the repository. For this experiment, Natural Language Toolkit (NLTK) [23] has been used which contains libraries for Natural Language Processing for Python programming language.

5.1.6. Linguistic Processing and Concept Identification

All words which are tagged as Nouns(NN/NNP/NNS) and Adjectives (JJ) are filtered out and all possible combinations of $Noun + Noun$, $Adjective + Noun$ and $(Noun/Adjective) + Noun$[24][25]. The results are shown in Table 3. The term count for each of these multi word term is then calculated against the original corpus and a positive term count implies that the corresponding multi-word term can be a potential concept and we eliminate the term if we get a zero term count. This process has been repeated for all the multi-words we have filtered out.

5.2. Concept Hierarchy Learning

Concept hierarchy learning module concentrates on leveraging a subsumption hierarchy[5] depicting an "is-a" relation between the concepts identified by the proposed algorithm. Subsumption relation is simple but considered as an important relationship type in any ontological structure and we calculate two probability conditions for the same. For any given two concepts, we first calculate $P(C_1|C_2)$ and then $P(C_2|C1)$, in order to establish a subsumption relation, the former probability must be 1 and the latter should be less than 1. In other words, C_1 subsumes C_2 if the documents in which C_2 occurs is a subset of the documents which C_1 occurs in.

For instance, consider two concepts *dial-up internet* and *network connection*, the proposed method computes $P(dialup\ internet|network\ connection)$ and $P(network\ connection|dial-up\ internet)$ and found that the number of documents in which *dialup internet* occurs is a subset of number of documents in which *network connection* occurs. That means there exists a subsumption relation between these two concepts and *dialup internet* concept may be subsumed by *network connection* concept. This process has been repeated for all concepts in the collection, and a part of such a hierarchy generated using our proposed algorithm is shown in Figure 2.

6. EVALUATION OF RESULTS

Here we evaluate the results produced by our proposed method and precision and recall measures are used for evaluating the quality of concepts leveraged. We have first created a human generated concept repository and kept for verifying against the machine generated concepts. Precision computes the fraction of machine extracted concepts that are also human generated, and recall measures concepts which are extracted by proposed algorithm that are also human authored. In information retrieval, it is estimated that achieving high precision and recall at same time is difficult and using a measure called F1, we can balance these two. Here, true positive is defined as the number of overlapped concepts between human generated concepts and concepts extracted by our pro-

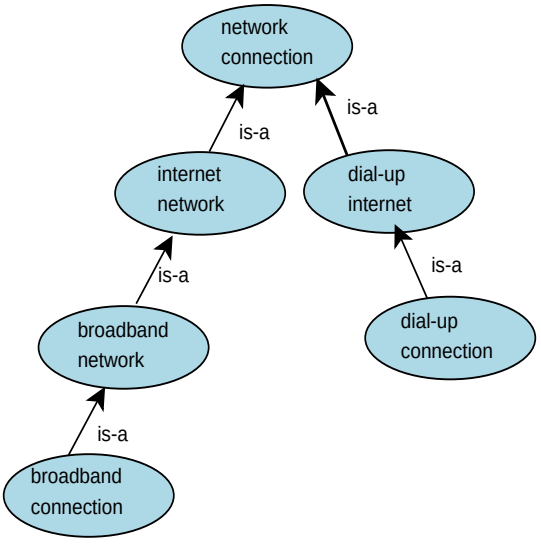


Figure 2. Part of a subsumption hierarchy learned using Algorithm 2

posed algorithm, false positive is the number of extracted concepts that are not truly human authored concepts and false negative is the human authored concepts that are missed by the concept extraction method. Using these measures, we have compared our proposed method against some of the existing concept extraction algorithms and the result is shown in Table 4.

From the performance graph shown in Figure 4, it is clear that our proposed algorithm extracts more concepts as the number of topics are increasing. The other baseline algorithms such as ACE and ICE performs poor when the number of topics are increased randomly. This shows that the proposed algorithm outperforms the baseline algorithms when extracting real-world concepts from large number of statistically generated topics.

Table 4
Comparison of ACE, ICE and our proposed method

Algorithm	Precision	Recall	F1
ACE	0.2372	0.2689	0.2517
ICE	0.7113	0.8147	0.7595
Proposed	**0.8165**	**0.8901**	**0.8516**

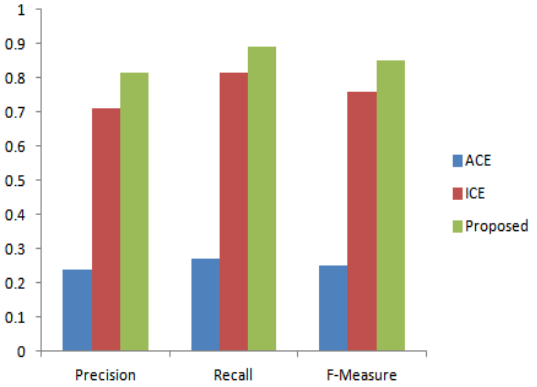


Figure 3. Precision, Recall and F-measure comparison of ACE, ICE and proposed method

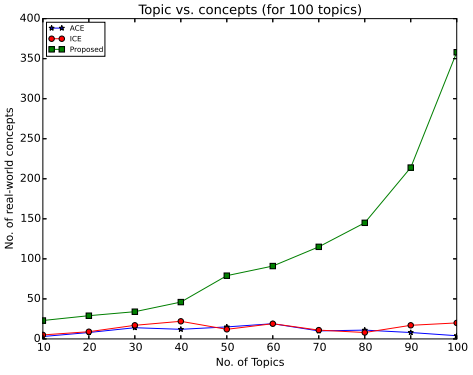


Figure 4. Performance of algorithms on 100 topics (No. of human interpretable concepts generated)

7. CONCLUSIONS AND FUTURE WORK

This paper proposed a novel framework for extracting close to real world concepts from large collection of unstructured text documents which is guided by a probabilistic topic modeling algorithm. Proposed method also deals with learning a subsumption hierarchy which exploits "is-a" relationships among identified concepts which is extensively used in ontology generation. Experiments conducted on large datasets such as Reuters and BBC news corpus shows that the proposed method outperforms

some of the already available algorithms and better concept identification is possible with this framework.

Because of the promising end results, we are interested to work mainly on the directions of measuring the scalability of proposed framework by using more large datasets. Apart from the basic subsumption hierarchy which depicts "is-a" relation, our future work will be on leveraging other relations that exist between concepts we would like to so that a this framework can automate the complete ontology generation process.

REFERENCES

1. Pospiech, Sebastian, Martin Pelke and Robert Mertens. Semi-automated Ontology Creation for Semantic Search in Business Process Exploration *IEEE Tenth International Conference on Semantic Computing (ICSC)*, 2016.
2. Marujo, Lus, *et al.*,. Exploring Events and Distributed Representations of Text in Multi-Document Summarization. *Knowledge-Based Systems*, 94:33–42, 2016.
3. Manek AS, Shenoy P D, Mohan MC and Venugopal K R. Aspect Term Extraction for Sentiment Analysis in Large Movie Reviews using Gini Index Feature Selection Method and SVM Classifier. *World Wide Web*, pages 1–20, 2016.
4. Steyvers M and Griffiths T. Probabilistic Topic Models. *Handbook of Latent Semantic Analysis*, 427(7):424–440, 2007.
5. Sanderson M and Croft B. Deriving Concept Hierarchies from Text. *In Proceedings of the* 22^{nd} *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 206–213, 1999.
6. Lindsey RV, Headden III WP and Stipicevic MJ. A Phrase-Discovering Topic Model using Hierarchical Pitman-yor Processes. *In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 214–222, 2012.
7. El-Kishky A, Song Y, Wang C, Voss CR and Han J. Scalable Topical Phrase Mining from Text Corpora. *Proceedings of the VLDB Endowment*, 8(3):305–316, 2014.
8. Wang X, McCallum A and Wei X. Topical n-grams: Phrase and Topic Discovery, with an Application to Information Retrieval. *In Proceedings of Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 697–702, 2007.
9. Ramirez P M, Mattmann C A. ACE: Improving Search Engines *via* Automatic Concept Extraction. *In Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration, IRI 2004,* pages 229–234, 2004.
10. Turney P D. Learning Algorithms for Key Phrase Extraction. *Information Retrieval.*, 2(4):303–316, 2000.
11. Parameswaran A, Garcia-Molina H and Rajaraman A. Towards the web of Concepts: Extracting Concepts from Large Datasets. *In Proceedings of the VLDB Endowment.*, 3(1-2):566-577, 2010.
12. Gelfand B, Wulfekuler M and Punch WF. Automated concept extraction from plain text. *In AAAI 1998 Workshop on Text Categorization.*, pages 13-17, 1998.
13. Rajagopal D, Cambria E, Olsher D and Kwok K. A graph-based approach to commonsense concept extraction and semantic similarity detection. *In Proceedings of the 22nd international conference on World Wide Web companion*, pages 565-570, 2013.
14. Krulwich B and Burkey C. Learning User Information Interests through Extraction of Semantically Significant Phrases. *In Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access*, pages 100–112, 1996.
15. Witten I H, Paynter G W, Frank E, Gutwin C and Nevill-Manning CG. KEA: Practical Automatic Key Phrase Extraction. *In Proceedings of the Fourth ACM Conference on Digital Libraries.*, pages 254-255, 1999.
16. Song M, Song IY and Hu X. KPSpotter: A Flexible Information Gain-based Key Phrase Extraction System. *In Proceedings of the* 5^{th} *ACM International Workshop on Web Information and Data Management.*, pages 50–53, 2003.
17. Frantzi K, Ananiadou S and Mima H. Automatic Recognition of Multi-word Terms: the c-value/nc-value Method. *International Journal on Digital Libraries.*, 3(2):115–130, 2000.
18. Hofmann T. Probabilistic Latent Semantic Indexing. *In Proceedings of the* 22^{nd} *Annual International ACM SIGIR Conference on Research and Development in Information Re-*

trieval, pages 50–57, 1999.
19. Blei DM, Ng AY and Jordan MI. Latent Dirichlet Allocation. *In Journal of Machine Learning Research.*, 3:993–1022, 2003.
20. Dumais ST. Latent Semantic Analysis. *Annual Review of Information Science and Technology.*, 38(1):188–230, 2004.
21. Hofmann T. Probabilistic Latent Semantic Indexing. *In Proceedings of the* 22^{nd} *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 50–57, 1999.
22. Blei DM, Ng AY and Jordan MI. Latent Dirichlet Allocation. *Journal of Machine Learning Research.*, 993-1022, 2003.
23. Bird S. NLTK: The Natural Language Toolkit. *In Proceedings of the COLING/ACL on Interactive Presentation Sessions*, pages 69–72, 2006.
24. Lewis D, Yang Y, Rose T and Li F. RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research*, 5:361-397, 2004.
25. Greene D and Cunningham P. Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering. *In Proceedings of the* 23^{rd} *International Conference on Machine Learning*, pages 377–384, 2006.

Anoop V S is a full time Ph.D Research Scholar at Data Engineering Lab, Indian Institute of Information Technology and Management - Kerala (IIITM-K), Thiruvananthapuram, India. He received his Masters in Computer Applications (MCA) - from IGNOU and Master of Philosophy in Computer Science from Cochin University of Science and Technology (CUSAT), Kerala in 2014. He has several publications in international journals, conference proceedings and book chapters. His research interests include Information Retrieval, Text Mining and NLP.

Asharaf S is an Associate Professor at Indian Institute of Information Technology and Management - Kerala (IIITM-K), Thiruvananthapuram, India. He received his Ph.D and Master of Engineering degrees in Computer Science and Engineering - from Indian Institute of Science, Bangalore. His areas of interest include Algorithms, Business Models and Software Systems related to Data Mining, Data Analytics, Information Retrieval, Computational Advertising, Soft Computing and Machine Learning.

Deepak Padmanabhan is a Lecturer (Asst. Professor) in Computer Science at Queens University Belfast, UK. He completed his M.Tech and Ph.D from Indian Institute of Technology Madras, all in Computer Science. His current research interests include Data Analytics, Similarity Search, Information Retrieval and Natural Language Processing. He has published over 40 research papers across major venues in Information and Knowledge Management. He is a Senior Member of the IEEE and ACM.

V.S. Anoop* and S. Asharaf

Aspect-Oriented Sentiment Analysis: A Topic Modeling-Powered Approach

https://doi.org/10.1515/jisys-2018-0299
Received July 9, 2018.

Abstract: Because of exponential growth in the number of people who purchase products online, e-commerce organizations are vying for each other to offer innovative and improved services to its customers. Current platforms give its customers innovative services such as product recommendations based on their purchase histories and location, product comparison, and most importantly, a platform for expressing their experience and feedback. It is important for any e-commerce organization to analyze this feedback and to find out the sentiment of the customers for giving them better products and services. As large reviews may contain feedback in a mixed manner where a customer gives his opinion on different product features in the same review, finding out the exact sentiment is tedious. This work proposes aspect-specific sentiment analysis of product reviews using a well-sophisticated topic modeling algorithm called latent Dirichlet allocation (LDA). The topic words, thus, extracted are mapped with various aspects of an entity to perform the aspect-specific sentiment analysis on product reviews. Experiments with synthetic and real dataset show promising results compared to existing methods of sentiment analysis.

Keywords: Latent Dirichlet allocation, sentiment analysis, topic modeling, e-commerce, aspect extraction, text mining.

1 Introduction

Platforms such as social networks, micro-blogs, online reviews, and discussion forums are growing very fast, and thus, the need for analyzing the sentiments of the users are also increasing. Sentiment analysis proves to be very effective in businesses and social domains because opinions matter and is critical for all human activities, and thus, they have become the key influence of human behavior. There are two broadly categorized types of the textual representation: one is fact; another is opinion. While facts are objective expressions about entities, events, and their properties, opinions are subjective expressions that describe people's sentiments toward entities or events. Product reviews in e-commerce platforms help the customer to analyze the product and take better judgments based on its previous usage history.

Sentiment analysis is the process of extracting subjective information from natural language text, and it also expresses the opinion or view of a user toward a topic. Sentiment means feelings, and it includes attitudes, emotions, and opinions, and sentiment is a subjective impression, not a fact. Sentiment analysis can be done by applying natural language processing, statistical methods, or machine learning techniques to identify, extract, or characterize the sentiment content of a text unit. The insights and applications from sentiment analysis were useful in other areas, like politics, law, policy making, sociology, and psychology. In general, humans are subjective creatures, and opinions are important, and being able to interact with people on that level has many advantages for the information system.

The main challenges in sentiment analysis are that people express their opinions in complex ways and the relevance of lexical content in text. There are many possibilities for classification such as users, texts, words,

***Corresponding author: V.S. Anoop,** Data Engineering Lab, Indian Institute of Information Technology and Management – Kerala (IIITM-K), IIITM-K Building, Technopark Campus, Thiruvananthapuram, Kerala 695581, India, e-mail: anoop.res15@iiitmk.ac.in
S. Asharaf: Indian Institute of Information Technology and Management – Kerala (IIITM-K), IIITM-K Building, Technopark Campus, Thiruvananthapuram, Kerala 695581, India

Authenticated | anoop.res15@iiitmk.ac.in author's copy
Download Date | 12/31/18 11:54 AM

sentences, etc. The words or short phrases are the building blocks of the sentiment expressed. The next step is to find the polarity keywords to identify the sentiment behind them. For example, there will be some relation between the words and positive reviews, and such keywords can be found out by data-driven methods for better accuracy. In the case of social networks like Twitter and Facebook, sentiment is most succinctly represented with emotions or smileys, and in such scenario, emotion recognition is performed to capture the sentiment expressed in them [35]. Therefore, customers are not only interested in the whole quality of the product but also some specific aspects of the product also. For example, considering a laptop as an example, customers are not only interested in the whole product but also into some other aspects such as "screen quality", "processor generation", or "random-access memory (RAM) speed". This leads to the extraction of various aspects of a product [31] to perform sentiment analysis of product reviews. For each class of products, there is a set of very general categorical aspects that are common to all products in that class. Aspects can be identified either manually or through an appropriate automated method. This work aims to identify aspects and then perform the sentiment analysis of product reviews using topic modeling. The main contributions of this paper are summarized as follows:

1. Outlines a brief explanation of aspect-oriented sentiment analysis and topic modeling, which are well-studied and addressed areas in text mining.
2. Proposes a topic modeling-powered approach for aspect extraction and aspect-based sentiment analysis from unstructured product reviews expressed by users of various products.
3. Verify experimentally the effectiveness and usefulness of proposed algorithms in extracting aspect-oriented sentiments from real-world and dynamic datasets.

Organization of this paper: The remainder of this paper is organized as follows. Section 2 describes the state-of-the-art in sentiment analysis; Section 3 throws light on some background of latent Dirichlet allocation (LDA), which is a widely used topic modeling algorithm. The experimental setup used in this work is explained in Section 5, and Section 6 presents the results and a detailed evaluation of the same. The author concludes the paper, and some interesting future dimensions are given in Section 7.

2 Related Work

In this section, we discuss some of the very recent works on sentiment analysis from unstructured text specifically focusing on aspect-specific sentiment analysis. Sentiment analysis is an ever-evolving research area, and this caused rapid generation of a vast array of research publications. Because of space constraints, we, here, mention some of the prominent works that are closely related to our proposed method. Shelke et al. [30] proposed an approach for domain-independent aspect-oriented sentiment analysis for product reviews. This unsupervised approach uses SentiWordNet lexical resource to determine the polarity of identified features. A client perception-based product feature mining algorithm was introduced by Li et al. [14] that makes use of sentiment orientation. The feature and sentiment orientation of products are sorted by the weights of user interest that help customers access review orientation more effectively. Another notable approach in aspect-oriented sentiment analysis was introduced in the literature by Akhtar et al. [1] that is composed of a two-step approach, viz., aspect term extraction and sentiment classification. Their pruned, compact set of features performed better compared to the chosen baselines that uses a complete set of features for sentiment classification. An ensemble-based particle swarm optimization they construct in the second phase using maximum entropy, conditional random field and support vector machine outperformed the baseline.

A linguistic rule-based approach for aspect-level sentiment analysis for movie reviews was reported in the literature, which was devised by Piryani et al. [25]. Their method used a linguistic rule-based approach, which identifies the aspects from movie reviews, locates opinion about that aspect, and computes the sentiment polarity of that opinion using linguistic approaches. The system then generates an aspect-level opinion summary, and the authors claim that the system shows good accuracy, and this can be deployed in an integrated opinion profiling system. Another interesting work that finds out aspect-level sentiment analysis on tweets about diabetes was introduced by Salas-Zárate et al. [28]. The authors calculate the sentiment of aspects

Authenticated | anoop.res15@iiitmk.ac.in author's copy
Download Date | 12/31/18 11:54 AM

by considering the words around the aspects using n-gram-based approach. They claim that this approach showed a precision of 81.93 % and an f-measure of 81.24 %.

A very recent comprehensive survey on text sentiment analysis from opinion to emotion mining by Yadollahi et al. [34] was reported in the literature very recently. The authors present the state-of-the-art methods in this area and proposes a taxonomy of sentiment analysis, a survey on polarity classification methods, emotion theories, and emotion mining research. Schouten et al. [29] proposed a novel approach for supervised and unsupervised aspect category detection for sentiment analysis with co-occurrence data. Their proposed approach applies association rule mining on co-occurrence frequency data to find aspect categories. An approach for aspect-based rating prediction on reviews using sentiment strength analysis was introduced by Wang et al. [32]. Previous works on sentiment analysis used reviewer's sentiment orientation, and in this work, the authors focus on the sentiment strength users expressed in the reviews.

A feature selection approach using multi-objective optimization for aspect-based sentiment analysis was reported by Akhtar et al. [2] in which the authors incorporated the concepts of multi-objective optimization, distributional thesaurus, and unsupervised lexical induction. This work uses support vector machines for sentiment classification and conditional random fields for aspect term and opinion target expression extraction tasks. Their experiments on benchmark setups of SemEval-2014 and SemEval-2016 shared tasks show that the proposed method achieved the state-of-the-art on aspect-based sentiment analysis for several languages. Another comprehensive and systematic literature review on opinion types and sentiment analysis techniques was contributed in the literature by Qazi et al. [27] in which the authors identified nine practices of review types, eight standard machine learning classification techniques, and seven practices of concept learning Sentic computing techniques. Their review offers insights on promising concept-based approaches to SA, which leverage commonsense knowledge and linguistics for tasks such as polarity detection. In addition to this, the practical implications of the methods are also explained in this review.

Ontology-based approach for enhancing aspect extraction process by identifying features pertaining to implicit entities was reported recently, which was proposed by Marstawi et al. [19]. Their Ontology-Based Product Sentiment Summarization (OBPSS) method outperformed other existing summarization systems in terms of aspect extraction and sentiment scoring. Our proposed method devise algorithms for aspect-oriented sentiment analysis, which are guided by topic modeling algorithm. This method model topics from product reviews and the obtained "topics" are mapped to corresponding aspects. Then, the polarity of these aspects are computed to summarize the overall sentiment of customers toward a product.

Very recently, Hazarika et al. proposed an approach for modeling inter-aspect dependencies for aspect-based sentiment analysis [9]. Majority of the present neural-based approaches largely ignore the inter-aspect dependencies and only capture aspect and its contextual information. In their proposed work, authors incorporated the inter-aspect pattern by simultaneous classification of all aspects in a sentence along with temporal dependency processing of their corresponding sentence representations using recurrent networks [9]. Another recent work that proposed a general two-stage approach for target-based sentiment analysis was introduced by Wang et al. [33]. In this approach, the first stage extract and group target-related words, and the second stage separates aspect and opinions from these identified groups. The proposed method showed better performance when there were no sufficient labeled data was available.

3 Background: Latent Dirichlet Allocation (LDA)

Topic modeling is a method for analyzing large quantities of unlabeled text data. A topic is a probability distribution over a collection of words, and a topic model is a formal statistical relationship between a group of observed and latent random variables that specifies a probabilistic procedure to generate the topics – a generative model [5]. LDA is the most popular and simplest topic model. It is a generative for text and other collections of discrete data that generalizes or improves on several previous models including PLSI (probabilistic latent semantic indexing). LDA is a generative probabilistic model for collections of discrete data such as text corpora. It is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is, in turn, modeled as an infinite mixture over

Authenticated | anoop.res15@iiitmk.ac.in author's copy
Download Date | 12/31/18 11:54 AM

an underlying set of topic probabilities, and the topic probabilities provide an explicit representation of a document in the context of topic modeling [6]. In LDA, we represent a word as the basic unit of discrete data, defined to be an item from a vocabulary indexed by {1. . .V}. A document is a sequence of N words denoted by $w = (w_1, \ldots w_n)$, where w_n is the nth word in the sequence, and a corpus is a collection of "M" documents denoted by $D = W_1 \ldots W_m$. The generative process for each document "w" in a corpus "D" is represented as follows [5]:

1. Choose $N \sim$ Poisson (ξ)
2. Choose $\theta \sim$ Dir (α)
3. For each of the N words w_n:
   (a) Choose a topic $z_n \sim$ Multinomial (θ)
   (b) (Choose a word w_n from $p(w_n \mid z_n, \beta)$, a multinomial probability conditioned on the topic z_n.

A k-dimensional Dirichlet random variable θ can take values in the $(k-1)$ simplex (a k-vector θ lies in the $(k-1)$ simplex if $\theta_i \geq 0, \sum_{i=1}^{k} \theta_i = 0$ and has the following probability density on this simplex:

$$P(\theta|\alpha) = \frac{\Gamma\left(\sum_{i=1}^{k} \alpha_i\right)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \ldots \theta_1^{\alpha_k - 1} \tag{1}$$

where the parameter α is a k-vector with components $\alpha_i > 0$, and where $\Gamma(x)$ is the γ function. Given the parameters α and β, the joint distribution of a topic mixture θ, a set of N topics z, and a set of N words w, is given by: $P(\theta, z, w|\alpha, \beta) = P(\theta|\alpha) \prod_{n=1}^{N} P(z_n|\theta)$

$$+P(w_n|z_n, \beta) \tag{2}$$

where $P(z_n|\theta)$ is simply θ_i for the unique i such $z_n = 1$. Integrating over θ and summing over z, we obtain the marginal distribution of a document as follows: $P(w|\alpha, \beta) = \int P(\theta|\alpha)\left(\prod_{n=1}^{N} \sum_{z_n} P(z_n|\theta\right)$

$$+P(w_n|z_n, \beta))d\theta \tag{3}$$

Finally, taking the product of the marginal probabilities of single documents, we obtain the probability of a corpus as:

$$(D|\alpha, \beta) = \prod_{d=1}^{M} \int P(\theta_d|\alpha)\left(\prod_{n=1}^{N} {}_d \sum_{z_{dn}} P(z_{dn}\theta_d) + P(w_{dn}|z_{dn}, \beta)d\theta_d\right) \tag{4}$$

4 Aspect-Oriented Sentiment Analysis: A Topic Modeling-Powered Approach

In this section, we describe our proposed framework for finding aspect-oriented sentiment of product reviews expressed by customers. Those reviews mostly contain mixed opinions so that individual word-level polarity computation may not be useful. Our topic modeling-powered approach first models topics from customer reviews, which are a collection of word uni-grams and then extracts aspects from it. Based on the probability, we map these topics to corresponding aspects, and then aspect-level polarity is computed, and using Naive Bayes classifier, we classify sentiments. The overall work flow of the proposed approach is shown in Figure 1. In this work, we used Amazon product review dataset, which consists of a large collection of review texts on mobile phones. Pre-processing is performed for converting raw text file into a well-defined sequence of linguistically meaningful units. This is an essential part of this system, as the characters, words, and sentences identified at this stage are the basic elements passed to all other processing stages. The words such as "and", "the", "but", and other punctuation and hyperlinks are removed from the reviews. Then, the LDA algorithm is executed on top of this cleansed dataset to generate topics. In the case of hundreds of review documents

Authenticated | anoop.res15@iiitmk.ac.in author's copy
Download Date | 12/31/18 11:54 AM

from this dataset, topic modeling will be a good approach to understand something of what the archive contains without reading every document and this will be helpful for extracting aspects from the review texts. Table 1 represents some topics that are extracted by topic modeling.

Algorithm 1 takes the product review corpus as the input, and the output will be aspect-specific sentences. First of all, the review corpus is pre-processed by removing stop-words and symbols. Then, LDA is performed

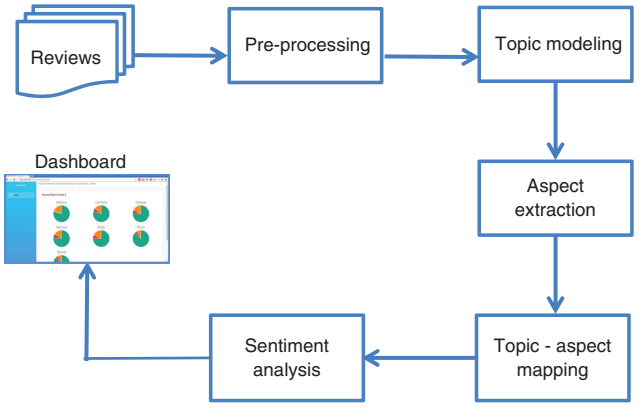


Figure 1: Overall Workflow of the Proposed Approach for Topic Modeling-Guided Aspect-Oriented Sentiment Analysis.

Table 1: Topics Extracted by LDA Algorithm (Because of Space Constraints, We Show Only the First Five Topics Generated).

Topic	Topic keywords
Topic 1	Software treo palm keyboard blackberry device pda data email pc sync computer windows web internet mobile version support contacts
Topic 2	Charger car cable work usb works plug charging power fine cord adapter charge worked device port doesn plugged connection
Topic 3	Buy money worth love cheap thing blue bought quot buying don people dont color waste paid cost cool wanted
Topic 4	Phone phones cell motorola nokia razr samsung lg reception features flip dropped calls drop basic sanyo vx razor krzr
Topic 5	Small size pocket design makes bit feel find big large nice hand easily carry fact easier smaller style buttons

Algorithm 1: Topic Modeling & Topic-Aspect Mapping

Data: "Product Review Corpus"
Result: "Aspect specific sentences"
Initialization;
Pre-processed Review Corpus ← Stopword Removal() & Symbols Removal()
Topics ← LDA(Pre-processed Review Corpus)
Aspect ← Product Aspects
for *each "Topic"* **do**
  Topic-Aspect Mapping;
end
for *each "Aspect"* **do**
  Scan the Review text lines;
  if *line contains "Topic words"* **then**
    File ←Write(line);
  else
    Skip the line;
  end
end

Authenticated | anoop.res15@iiitmk.ac.in author's copy
Download Date | 12/31/18 11:54 AM

Algorithm 2: Aspect specific Sentiment Analysis

```
Data: File ["Aspect specific sentences"]
Result: "Sentiment Polarity score & Sentiment"
Sentiment Analysis() ← File
while not the end of lines do
    Calculate Sentiment Polarity Score;
    For each 'line'
    if Sentiment Polarity Score(line) >0 then
        Sentiment ← Positive;
    else
        if Sentiment Polarity Score(line) <0 then
            Sentiment ← Negative;
        else
            if Sentiment Polarity Score(line) =0 then
                Sentiment ← Neutral;
            else
            end
        end
    end
end
```

on the pre-processed corpus to do topic modeling, and thus, topics are extracted. Each topic is then mapped to a particular aspect, and for each aspect, the algorithm scans the pre-processed corpus to find a match with topic words to extract aspect-specific sentences. Thus, for each aspect, the aspect-specific sentences will be written to a text file.

Algorithm 2 takes the file containing aspect-specific sentences as the input, and it calculates the polarity score for each such sentence. The polarity score determines the sentiment of the sentence. If the polarity score

Algorithm 3: Finding sentiment on "Aspect"

```
Data: "Polarity count of aspect specific sentences"
Result: "Sentiment on "Aspect"
for each "Aspect" do
    Positive ← Total Positive polarity count
    Negative ← Total Negative polarity count
    Neutral ← Total Neutral count
    if Positive > (Negative & Neutral) then
        Opinion ← Positive;
    else
        if Negative > (Positive & Neutral) then
            Opinion ← Negative;
        else
            if Neutral > (Positive & Negative) then
                Opinion ← Neutral
            else
            end
        end
    end
end
```

Authenticated | anoop.res15@iiitmk.ac.in author's copy
Download Date | 12/31/18 11:54 AM

of a sentence is greater than zero, the sentiment is "Positive", and if the polarity score is less than zero, then the sentiment is "Negative". If the polarity score is equal to zero, then, the sentiment toward the particular sentence is said to be "Neutral".

Algorithm 3 takes the total polarity count of aspect-specific sentences as the input, and it expresses the sentiment on that particular aspect. If the total positive polarity count is higher than the negative and neutral polarity counts, then the sentiment toward the particular aspect is said to be "Positive". Similarly, if the total negative polarity count is higher than the positive and neutral polarity counts, then the sentiment toward a particular aspect is said to be "Negative". If the total neutral count is higher than the positive and negative polarity counts, then the sentiment toward the particular aspect is said to be "Neutral".

5 Experimental Setup

This section details the experimental setup we have used for implementing proposed algorithms on Amazon product reviews available on public web. A brief description of datasets used in this experiment and the experimental testbed details are discussed below.

5.1 Dataset Description

We use Amazon product review dataset available at the url http://jmcauley.ucsd.edu/data/amazon/links.html. This dataset contains product reviews and associated metadata from Amazon, including 142.8 million reviews spanning for the period May 1996 to July 2014. This dataset includes reviews (ratings, text, helpfulness votes), product metadata (descriptions, category information, price, brand, and image features), and links (also viewed/also bought). It contains reviews for a variety of product categories such as books, electronics, beauty, automotive, etc. For this current experiment, we used "Cell Phones and Accessories" category and used 194,439 reviews for our experiment.

5.2 Experimental Testbed

All algorithms discussed in this paper were implemented on a server configured with AMD Opteron 6376 @ 2.3 GHz having 16 core processor and 16 GB of main memory. These algorithms were programmed using Python 2.7 and removal of stopwords and other pre-processing were done using Python NLTK (Natural Language Tool Kit) [4], and stopword list for English language was used. We also removed punctuation, URLs, and other special characters to make the experimental ready copy of our dataset. We observed that the user reviews contain emojis, which plays an important role in expressing the user sentiment, but those emojis cannot be handled by our topic modeling algorithm. We also removed emojis from the dataset and considered only text as our input to the LDA algorithm. For topic modeling, we used MALLET (MAchine Learning for LanguagE Toolkit), which is freely available at the url http://mallet.cs.umass.edu/. For modeling topics, we chose 300 as the number of iterations in Gibbs sampling as by trail and error method we found that Gibbs sampling approaches the target distribution after 300 rounds of iterations. The α and β parameters of LDA was set to $50/Z$ and 0.01, respectively.

5.3 Baselines

We compare our proposed algorithms with two state-of-the-art methods available in the literature for aspect-oriented sentiment analysis. The first baseline [19] is an approach that uses ontology for extraction of aspects and eliminates lexicon-based sentiment scoring issues. Thus, they claim that this will improve the accuracy of sentiment analysis. The second baseline [25] is a linguistic rule-based approach for aspect-level sentiment analysis of movie reviews. The authors devised a linguistic rule-based approach, which identifies the aspects from movie reviews, locates opinion about that aspect, and computes the sentiment polarity of that opinion

Authenticated | anoop.res15@iiitmk.ac.in author's copy
Download Date | 12/31/18 11:54 AM

using linguistic approaches [25]. We chose these two baselines as we found these methods closely associated with our proposed algorithms. A comparison result in terms of Precision, Recall, and Accuracy is given in the experimental results section.

6 Results and Evaluation

In this section, we present the results obtained from the experimental setup discussed in Section 5. The topics generated by the LDA algorithm is shown in Table 1. Because of space constraints, we show only the first five topics generated by the LDA, which are word unigrams. Each of these topics are then mapped to aspects identified from the product descriptions such as Network, Battery, Price, etc., and the results of such mappings are shown in Table 2. In Table 3, we show the polarity score and sentiment of the review text for the "Battery" aspect of the product. Various polarity counts for such identified aspects are given in Table 4. In Table 5, we outline the result of our aspect-oriented sentiment analysis algorithm. For further analysis and interpretation, we plot the true positive rate (TPR) and false positive rate (FPR) using the receiver operating characteristics (ROC) plot, and such graphs for the aspects "Battery and Body", "Display and Network", "Price and Service" are shown in Figures 2–4, respectively. Figures 5 and 6 shows the charts depicting the precision, recall, and f-measure comparisons of our proposed method

Table 2: Result of Topic–Aspect Mapping.

Topic	Topic words	Aspect
Topic 1	Software treo palm keyboard blackberry device pda data email pc sync computer windows web internet mobile version support contacts	Network
Topic 2	Charger car cable work usb works plug charging power fine cord adapter charge worked device port doesn plugged connection	Battery
Topic 3	Buy money worth love cheap thing blue bought quot buying don people dont color waste paid cost cool wanted	Price
Topic 4	Phone phones cell motorola nokia razr samsung lg reception features flip dropped calls drop basic sanyo vx razor krzr	Body
Topic 5	Small size pocket design makes bit feel find big large nice hand easily carry fact easier smaller style buttons	Battery

Based on the number of words that are related to an aspect, we map the entire topic to the corresponding aspect.

Table 3: Polarity Score and Sentiment of Review Texts for the Aspect "Battery".

1	The first battery, from a different vendor was not good and needed charging daily	−0.025	Neg	Correct
2	The battery gets hot when charging so I would not use this as my main battery	0.20	Pos	Wrong
3	The battery was charging all night long, and then, this morning the battery died	−0.05	Neg	Correct
4	Powerful battery, needs charging once in 3–4 days	0.3	Pos	Correct

Table 4: Polarity Count for Various Aspects.

Aspect	Total	Pos	Neg	Neu
Battery	26,323	7890	5591	12,842
Body	23,840	14,446	4148	5246
Display	18,421	10,332	3451	4638
Network	17,884	1577	14,075	2232
Price	34,135	18,333	5816	9986
Service	37,323	7701	16,302	13,320
Sound	67,514	42,366	12,142	13,006

Authenticated | anoop.res15@iiitmk.ac.in author's copy
Download Date | 12/31/18 11:54 AM

Table 5: Result of Aspect-Oriented Sentiment Analysis.

No	Aspect	Sentiment
1	Battery	Neutral
2	Body	Positive
3	Display	Positive
4	Network	Negative
5	Price	Positive
6	Service	Negative
7	Sound	Positive

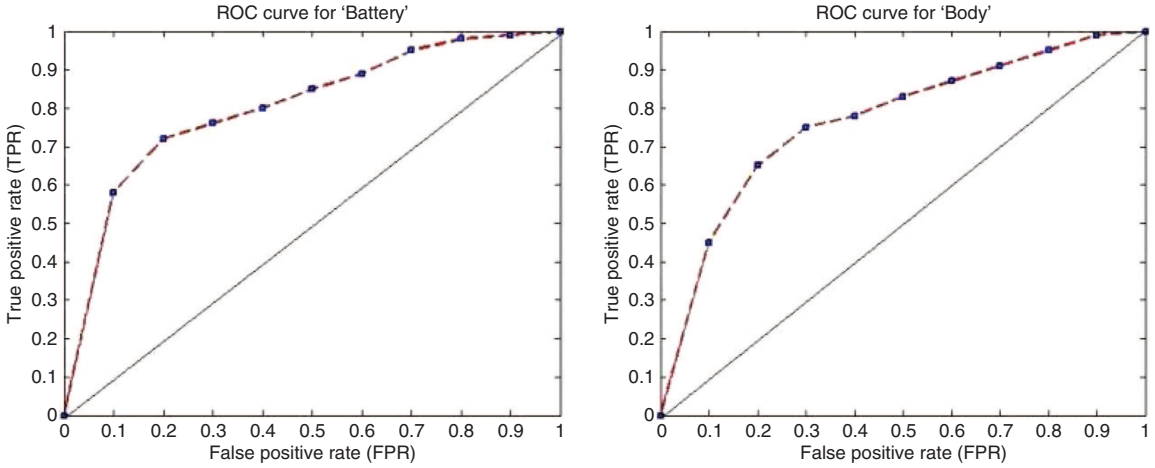


Figure 2: ROC Plots for "Battery" and "Body".

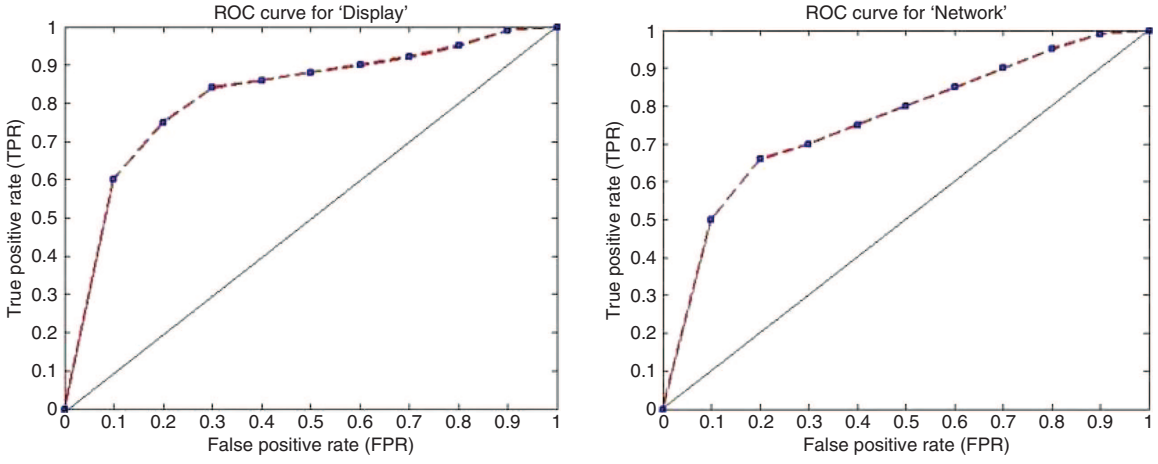


Figure 3: ROC Plots for "Display" and "Network".

with the chosen baselines for the aspects "Battery and Body", "Display and Network", "Price and Service". We compare our proposed algorithms with two state-of-the-art methods for establishing the usefulness and better accuracy of our proposed algorithms. The results in terms of precision, recall, and f-measure are given in Table 6, and the data clearly shows that our proposed method outperforms our chosen baselines for all the aspects we have considered. Figure 7 shows the screenshot of a web application we have developed to visualize the results of our aspect-oriented sentiment analysis framework implemented

Authenticated | anoop.res15@iiitmk.ac.in author's copy
Download Date | 12/31/18 11:54 AM

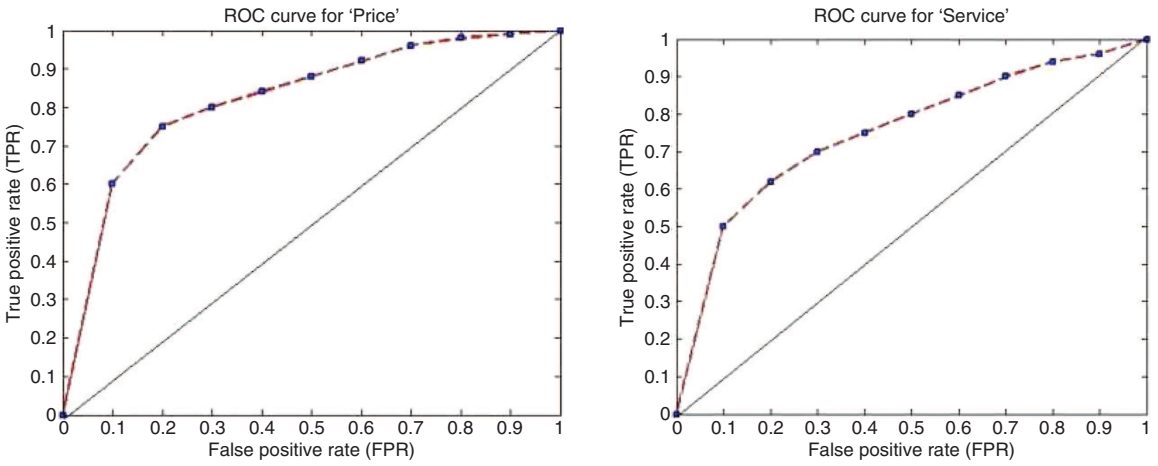


Figure 4: ROC Plots for "Price" and "Service".

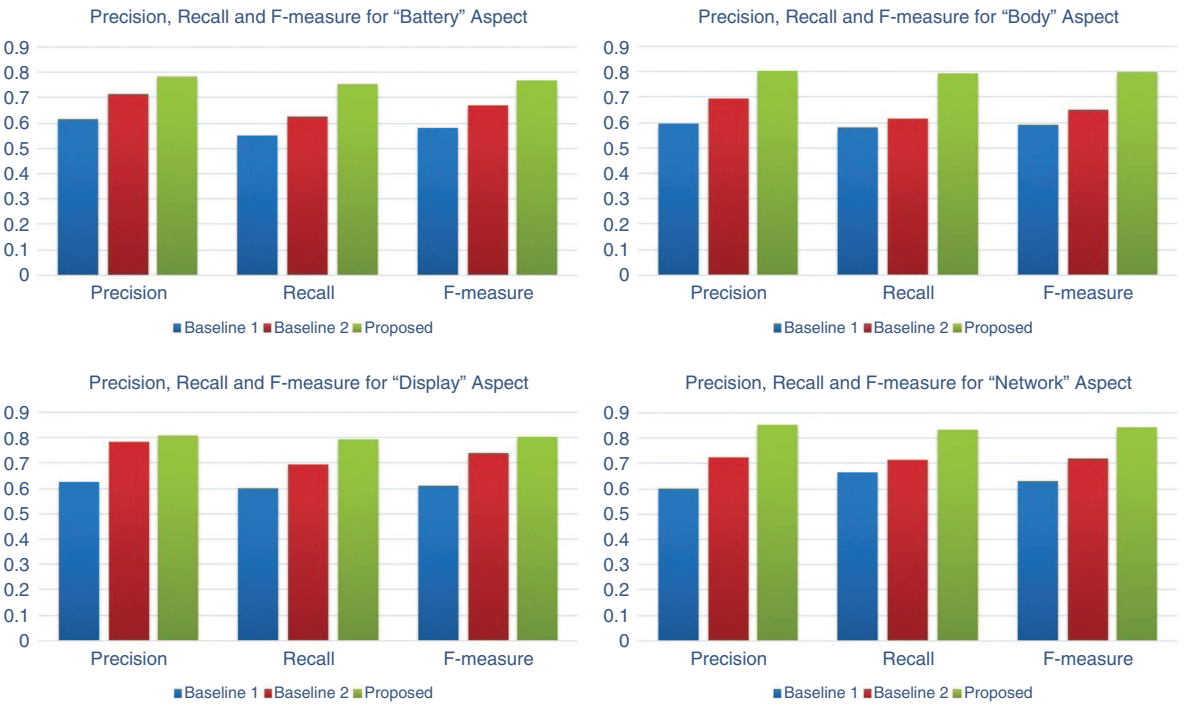


Figure 5: Graph Showing Precision, Recall, and f-Measure for Various Aspects.

on real-time data. The current screenshot shows the result of aspect-based sentiment analysis for the mobile Xiaomi Redmi Note 3, and the corresponding user reviews are collected from the famous customer review platform called MouthShut. Our proposed algorithms could clearly show the user sentiments on different aspects of the mobile, and the algorithms were found to be highly scalable. This system has already been implemented for some e-commerce vendors, and they are using it as a pilot version.

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{5}$$

$$Precision = TP/(TP + FP) \tag{6}$$

$$Recall = TP/(TP + FN) \tag{7}$$

Authenticated | anoop.res15@iiitmk.ac.in author's copy
Download Date | 12/31/18 11:54 AM

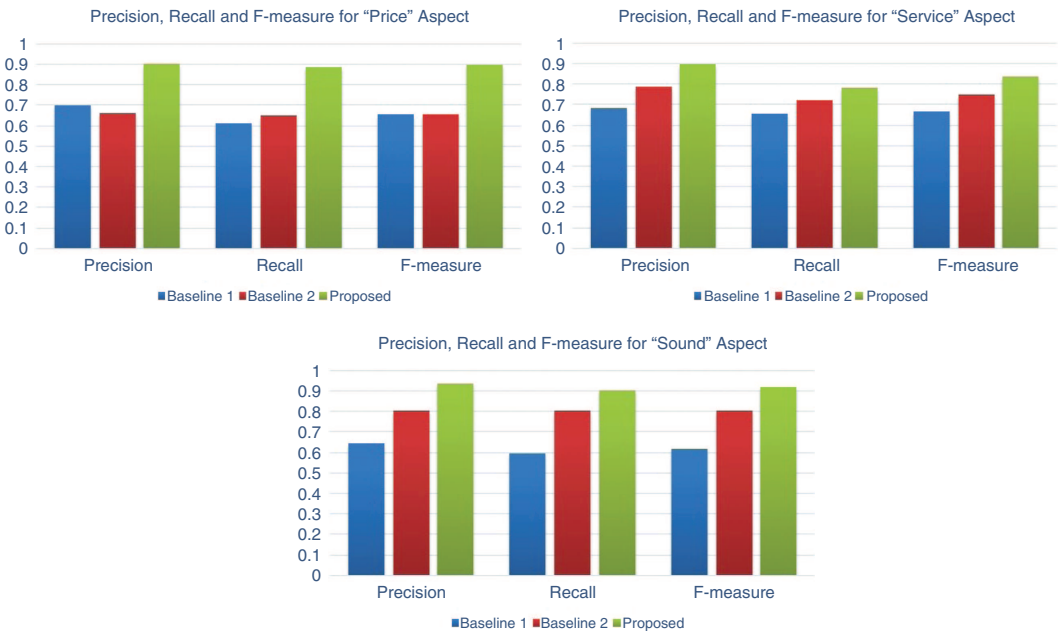


Figure 6: Graph Showing Precision, Recall, and f-Measure for Various Aspects.

Table 6: Performance of our Proposed Aspect-Oriented Sentiment Analysis When Compared with Chosen Baselines [25, 19].

	Baseline 1 [19]			Baseline 2 [25]			Proposed method		
Aspect	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Battery	0.6158	0.5485	0.5802	0.7155	0.6252	0.6673	0.7845	0.7544	0.7691
Body	0.5952	0.5812	0.5881	0.6925	0.6152	0.6515	0.8012	0.7955	0.7983
Display	0.6266	0.5984	0.6121	0.7852	0.6951	0.7374	0.8101	0.7920	0.8009
Network	0.6011	0.6652	0.6315	0.7241	0.7129	0.7184	0.8522	0.8311	0.8415
Price	0.7010	0.6123	0.6536	0.6612	0.6522	0.6566	0.9054	0.8856	0.8953
Service	0.6840	0.6551	0.6692	0.7852	0.7210	0.7517	0.8995	0.7845	0.8380
Sound	0.6450	0.5963	0.6196	0.8021	0.8045	0.8032	0.9388	0.9012	0.9196

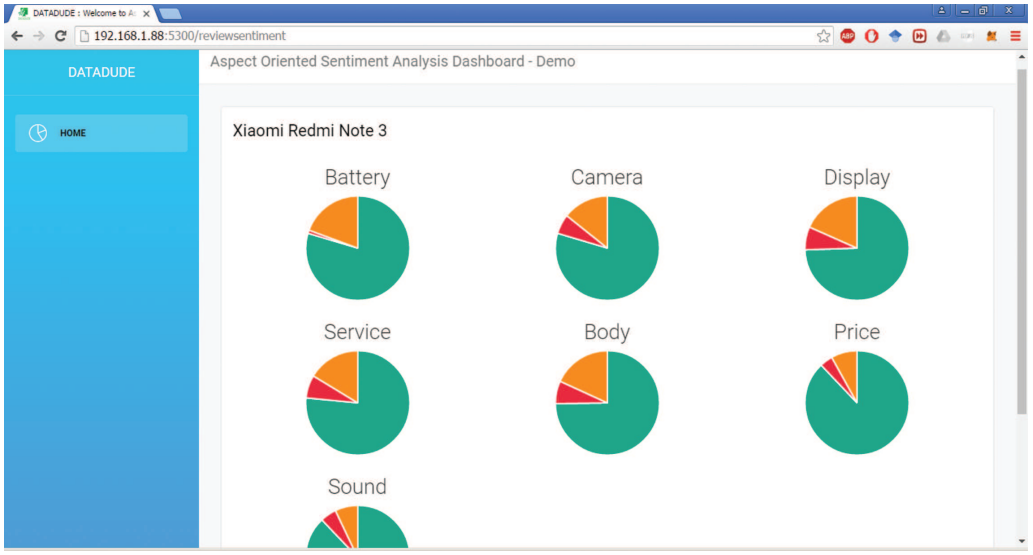


Figure 7: Screenshot of Web Application Developed for Finding Aspect-Oriented Sentiment of Customers for Mobile Phones and Accessories Using Proposed Algorithms.
This system was extended to production level, and a number of vendors are already using a pilot version of the same.

Authenticated | anoop.res15@iiitmk.ac.in author's copy
Download Date | 12/31/18 11:54 AM

7 Conclusions and Future Work

This paper proposed a novel approach for aspect-oriented sentiment analysis of e-commerce product reviews, which are publicly available in the form of unstructured text. This approach make use of LDA algorithm to perform topic modeling on top of the dataset to bring out hidden themes as "topics", and then these topics are mapped to various aspects of the product. Then, an aspect specific sentiment analysis is implemented using machine learning techniques. The proposed method shows accuracy ranges from 74 % to 81 % for each aspect and also outperforms some of the state-of-the-art methods on sentiment analysis.

In this work, the aspects from the vendor websites are collected and then topics are assigned to certain aspects, which requires manual effort. As the end results are promising, the authors would like to work on the dimensions of extracting aspects automatically, and algorithmic techniques can be developed to map the topics into extracted aspects. That may certainly improve the efficiency of the proposed algorithms, and the authors would like to work on these dimensions in the future. Studying and improving the scalability aspects of the algorithms are another potential future work.

Acknowledgment: The authors would like to thank staffs and researchers at Data Engineering Lab, Indian Institute of Information Technology and Management – Kerala (IIITM-K) for their valuable suggestions and comments that significantly improved the quality of this paper. The authors also thank Dr. Julian McAuley (Assistant Professor, UC San Diego) for providing the complete datasets for this experiment. On behalf of all the authors, the corresponding author states that there is no conflict of interest.

Bibliography

[1] M. S. Akhtar, D. Gupta, A. Ekbal, and P. Bhattacharyya, Feature selection and ensemble construction: a two-step method for aspect based sentiment analysis, *Knowl. Based Syst.* **125** (2017), 116–135.

[2] M. S. Akhtar, S. Kohail, A. Kumar, A. Ekbal and C. Biemann, Feature selection using multi-objective optimization for aspect based sentiment analysis, in: *International Conference on Applications of Natural Language to Information Systems*, pp. 15–27, Springer, Cham., Liege, Belgium, 2017.

[3] D. Bespalov, B. Bai, Y. Qi and A. Shokoufandeh, Sentiment classification based on supervised latent n-gram analysis, in: *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pp. 375–382, ACM, Glasgow, Scotland, UK, 2011.

[4] S. Bird and E. Loper, NLTK: the natural language toolkit, in: *Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions*, p. 31, Association for Computational Linguistics, July 2004. Available at http://www.aclweb.org/anthology/P04-3031

[5] D. M. Blei, A. Y. Ng and M. I. Jordan, Latent Dirichlet allocation, *J. Mach. Learn. Res.* **3** (2003), 993–1022.

[6] Colorado Reed, Latent Dirichlet allocation: towards a deeper understanding, January 2012.

[7] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas and R. A. Harshman, Indexing by latent semantic analysis. *JAsIs* **41** (1990), 391–407.

[8] W. Ding, X. Song, Y. Shang, J. Zhu, L. Guo, X. Hu, Similarity dependency dirichlet process for aspect-based sentiment analysis.

[9] D. Hazarika, S. Poria, P. Vij, G. Krishnamurthy, E. Cambria and R. Zimmermann, Modeling inter-aspect dependencies for aspect-based sentiment analysis, in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, New Orleans, LA, Volume 2, Short Papers, pp. 266–270, 2018.

[10] T. Hofmann, Probabilistic latent semantic analysis, in: *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pp. 289–296, Morgan Kaufmann Publishers Inc., Stockholm, Sweden, 1999.

[11] J. Huang, J. Lu and C. X. Ling, Comparing naive Bayes, decision trees, and SVM with AUC and accuracy, in: *Data Mining*, 2003. *ICDM* 2003, *Third IEEE International Conference on*, *IEEE*, Melbourne, FL, USA, 2003.

[12] S. Kim, J. Zhang, Z. Chen, A. H. Oh and S. Liu, A hierarchical aspect-sentiment model for online reviews, in: *AAAI*, Bellevue, WA, USA, 2013.

[13] H. Li, H. Y. Chen, H. Ji, S. Muresan and D. Zheng, Combining social cognitive theories with linguistic features for multi-genre sentiment analysis, in: *Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation*, Bali, Indonesia, pp. 127–136, 2012.

[14] J. J. Li, H. Yang and H. Tang, Feature mining and sentiment orientation analysis on product, in: *Management Information and Optoelectronic Engineering: Proceedings of the 2016 International Conference on Management, Information and*

Authenticated | anoop.res15@iiitmk.ac.in author's copy
Download Date | 12/31/18 11:54 AM

Communication (ICMIC2016) and the 2016 International Conference on Optics and Electronics Engineering (ICOEE2016), p. 79, World Scientific, Guilin, China, 2017.

[15] T. Li, Y. Zhang and V. Sindhwani, A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge, in: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, Volume 1, Association for Computational Linguistics, Suntec, Singapore, 2009.

[16] B. Liu, Sentiment analysis and opinion mining, in: *Synthesis Lectures on Human Language Technologies*, Morgan & Claypool Publishers, San Rafael, CA, USA, 2012.

[17] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng and C. Potts, Learning word vectors for sentiment analysis, in: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Volume 1, pp. 142–150, Association for Computational Linguistics, Portland, OR, USA, 2011.

[18] R. Malouf and T. Mullen, Taking sides: user classification for informal online political discourse, *Internet Res.* **18** (2008), 177–190.

[19] A. Marstawi, N. M. Sharef, T. N. M. Aris and A. Mustapha, Ontology-based aspect extraction for an improved sentiment analysis in summarization of product reviews, in: *Proceedings of the 8th International Conference on Computer Modeling and Simulation*, pp. 100–104, ACM, Canberra, ACT, Australia, 2017.

[20] J. McAuley and J. Leskovec, Hidden factors and hidden topics: understanding rating dimensions with review text, in: *Proceedings of the 7th ACM Conference on Recommender systems*, ACM, Hong Kong, China, 2013.

[21] A. K. McCallum, MALLET: a machine learning for language toolkit, 2002. Available at: http://mallet.cs.umass.edu.

[22] A. Neviarouskaya, H. Prendinger and M. Ishizuka, Affect analysis model: novel rule-based approach to affect sensing from text. *Nat. Lang. Eng.* **17** (2011), 95–135.

[23] G. Paltoglou and M. Thelwall, A study of information retrieval weighting schemes for sentiment analysis, in: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 1386–1395, Association for Computational Linguistics, Uppsala, Sweden, 2010.

[24] B. Pang, L. Lee and S. Vaithyanathan, Thumbs up?: sentiment classification using machine learning techniques, in: *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing*, Volume 10, pp. 79–86, Association for Computational Linguistics, Stroudsburg, PA, USA, 2002.

[25] R. Piryani, V. Gupta, V. K. Singh and U. Ghose, A linguistic rule-based approach for aspect-level sentiment analysis of movie reviews, in: *Advances in Computer and Computational Sciences*, pp. 201–209, Springer, Singapore, 2017.

[26] S. Poria, A. Gelbukh, B. Agarwal, E. Cambria and N. Howard, Sentic demo: a hybrid concept-level aspect-based sentiment analysis toolkit, in: *ESWC 2014*, 2014.

[27] A. Qazi, R. G. Raj, G. Hardaker and C. Standing, A systematic literature review on opinion types and sentiment analysis techniques: tasks and challenges, *Internet Res.* **27** (2017), 608–630.

[28] M. D. P. Salas-Zárate, J. Medina-Moreira, K. Lagos-Ortiz, H. Luna-Aveiga, M. A. Rodríguez-García and R. Valencia-García, Sentiment analysis on tweets about diabetes: an aspect-level approach, *Comput. Math. Methods Med.* **2017** (2017). Article ID: 5140631.

[29] K. Schouten, O. van der Weijde, F. Frasincar and R. Dekker, Supervised and unsupervised aspect category detection for sentiment analysis with co-occurrence data, *IEEE Trans. Cybern.* **48** (2018), 1263–1275.

[30] N. Shelke, S. Deshpande, and V. Thakare, Domain independent approach for aspect oriented sentiment analysis for product reviews, in: *Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications*, pp. 651–659, Springer, Bhubaneswar, India, 2017.

[31] A. Shen, Aspect-specific ranking of product reviews using topic modeling. Available online at: https://cs.brown.edu/research/pubs/theses/masters/2012/shen.pdf

[32] Y. Wang, Y. Huang and M. Wang, Aspect-based rating prediction on reviews using sentiment strength analysis, in: *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pp. 439–447, Springer, Cham. Arras, France, 2017.

[33] S. Wang, M. Zhou, S. Mazumder, B. Liu and Y. Chang, Disentangling aspect and opinion words in target-based sentiment analysis using lifelong learning, 2018, arXiv preprint arXiv:1802.05818.

[34] A. Yadollahi, A. G. Shahraki and O. R. Zaiane, Current state of text sentiment analysis from opinion to emotion mining, *ACM Comput. Surv. (CSUR)* **50** (2017), 25.

[35] H. Yu and V. Hatzivassiloglou, Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences, in: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 129–136, Association for Computational Linguistics, Honolulu, HI, USA, 2003.

Authenticated | anoop.res15@iiitmk.ac.in author's copy
Download Date | 12/31/18 11:54 AM

BIO-DATA

1. **Name** : Anoop V.S.
2. **Age & Date of Birth** : 31, 13 – 10 – 1987
3. **Address for correspondence** : 'Pratheeksha', Pullupana (P,O)
   Kadakkal, Kollam, Kerala
   Pincode: 691536
   Phone: +91 9747091417
4. **E-mail address** : anoop.res15@iiitmk.ac.in / anoop.academia@gmail.com

5. **Educational Qualifications**

Sl. No	Examination Passed	University / Institution	Year of Passing	Marks / Grade
1	Master of Philosophy (M.Phil)	CUSAT	2014	9 (CGPA)
2	Master of Computer Applications (MCA)	IGNOU	2011	71%

6. **Experience**

Sl. No.	Particulars	Period	Company / Institution
1.	Software Engineer	May 2008 – July 2011	M Squared Software & Services, Technopark, Thiruvananthapuram

7. **Declaration**

I hereby declare that all the information furnished above are correct and best to my knowledge and belief.

ANOOP V.S.