إقـــــــرار

أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان :

"Mapping Microblogs into A Network of Topics: A case study on Microblogs generated in learning Activities"

أقر بأن ما اشتملت عليه هذه الرسالة إنما هو نتاج جهدي الخاص ، باستثناء ماتمت الإشارة إليه حيثما ورد ،وإن هذه الرسالة ككل أو جزء منها لم يقدم من قبل لنيل درجة أو لقب علمي أو بحثي لدي أو مؤسسة تعليمية أو بحثية أخرى.

DECLARATION

The work presented in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted elsewhere for any other degree or qualification.

Student's name: اسم الطالب: عذير سعيد أبوعودة
Signature: التوقيع: عذير
Date: التاريخ: 4/5/2015

Islamic University of Gaza
Faculty of Information Technology

"Mapping Microblogs into A Network of Topics: A case study on Microblogs generated in learning Activities"

By

Ghadeer Abu-Oda

Master Thesis

A Master Thesis presented to the Faculty of Information Technology of Islamic University of Gaza in partial fulfillment of the requirements for the degree of Master of Science in Information Technology.

Advisor: *Dr. Rawia Awadallah*

Gaza, March 2015

بسم الله الرحمن الرحيم

الجامعة الإسلامية – غزة
The Islamic University - Gaza

مكتب نائب الرئيس للبحث العلمي والدراسات العليا هاتف داخلي: 1150

الرقم Ref ج س غ/35/

التاريخ Date 2015/03/04م

نتيجة الحكم على أطروحة ماجستير

بناءً على موافقة شئون البحث العلمي والدراسات العليا بالجامعة الإسلامية بغزة على تشكيل لجنة الحكم على أطروحة الباحثة/ **غدير سعيد مصطفى أبو عودة** لنيل درجة الماجستير في **كلية *تكنولوجيا المعلومات*** **برنامج تكنولوجيا المعلومات** وموضوعها:

تمثيل المدونات الصغيرة على هيئة شبكة من المواضيع

Mapping Microblogs into a Network of Topics: A case Study on Microblogs Generated in Learning Activities

وبعد المناقشة العلنية التي تمت اليوم الأربعاء 13جمادى الأولى 1436هـ، الموافق 2015/03/04م الساعة الواحدة والنصف ظهراً بمبنى اللحيدان، اجتمعت لجنة الحكم على الأطروحة والمكونة من:

د. رواية فوزي عوض الله	مشرفاً و رئيساً	.....
أ.د. علاء مصطفى الهليس	مناقشاً داخلياً	.....
د. يوسف نبيل أبو شعبان	مناقشاً خارجياً	31.03.2015 .....

وبعد المداولة أوصت اللجنة بمنح الباحثة درجة الماجستير في **كلية *تكنولوجيا المعلومات*/ برنامج تكنولوجيا المعلومات.**

واللجنة إذ تمنحها هذه الدرجة فإنها توصيها بتقوى الله ولزوم طاعته وأن تسخر علمها في خدمة دينها ووطنها.

والله ولي التوفيق،،،

مساعد نائب الرئيس للبحث العلمي والدراسات العليا

أ.د. فؤاد علي العاجز

P.O. Box 108, Rimal, Gaza, Palestine fax: +970 (8) 286 0800 فاكس Tel: +970 (8) 286 0700 هاتف ص.ب. 108 الرمال، غزة، فلسطين
public@iugaza.edu.ps www.iugaza.edu.ps

This thesis is dedicated to my kind father, who taught me that the more time a man spends working toward an objective, the more fruitful results he gains. It is also dedicated to my lovely mother who taught me that even the largest task can be accomplished if it is done one step at a time. . . .

Acknowledgements

I am grateful to Dr. *Rawia Awadallah*. She has been the ideal advisor I have ever had. Her sage advice, insightful criticisms, and patient encouragement were the most important factors to accomplish this thesis.
Dr.Rawia, Thanks so much for all you do . . .

I would like also to thank the following people who contribute in judgment tasks:
Doaa shamallakh, Web Developer - Information Technology Graduate.
Eman elshorafa, Web Developer - Information Technology Graduate.
Mariam AbuItiewi, Founder and CEO - wasselni - Information Technology Graduate.

Abstract

Microblogging– a kind of blogging where users publish snippets of information about their daily activities and thoughts over the Internet– has quickly become popular during the last few years. On Twitter as an example of microblogs, millions of people post short text based updates– tweets– about broad range of issues. Topics range from their personal life and work, to current events, news, and interesting observations and political thoughts.

In the midst of this general acceleration in using social media, there is an increase in using it in learning, in particular microblogging environments such as Twitter. As a result, there is an increase in the amount of data within these social media that reflects what people are learning and how well they are learning. For example, courses that adopt Twitter in the learning process allow students to discuss with each others and with their teacher different topics and to express their opinions on various aspects of these topics. The data generated out of these discussions constitutes a valuable resource on which teachers can rely for courses' or learning activities evaluation.

To understand this complicated landscape of topics that people are learning during the life time of the learning process, and how well these topics are being learnt, it would be helpful if they represented into a network . This network organizes topics being discussed in these microblogs according to their various subtopics and stored information about expressed opinions on these topics. This envisioned network would provide information that would help to track the learning process over time and to observe the strengths and the limitations in the learning process while it is happening.

In this research, we aim to study the possibility of mapping microblogs generated in learning activities into **a network of topics** by utilizing methods from Microblogs Analytic, Learning Analytic and Text Mining. This network would provide essential feedback about what topics are being learnt by people, the size of interest in particular topics and how well these topics are being learnt.

We conduct several experiments to develop and evaluate the framework that maps the microblogs generated into a network of topics. End to End evaluation of the complete system achieved 87% accuracy based on the used measurements.

Keywords: Learning Analytics, Microblog Analytics, Text Mining

Contents

List of Figures

List of Tables

List of Abbreviations

IR Information Retrieval

LDA Latent Dirichlet Allocation

MAMINT MApping MIcroblogs into A Network of Topics- thesis framework

MAP Mean Average Precision

TF Term Frequency

TF-IDF Term Frequency - Inverse Document Frequency

SPARQL SPARQL Protocol and RDF Query Language

VSM Vector Space Model

1 Introduction

During the recent years, microblogging – a kind of blogging where users publish snippets of information about their daily activities and thoughts– has become very popular. On Twitter as an example of microblogs, millions of people post short text based updates –tweets –about broad range of issues. Topics range from their personal life and work, to current events, news, and interesting observations and political thoughts. The tweets are published on the authors' personal Twitter page and sent to their followers, people who subscribe to other people's tweets. By following a group of people, users manage awareness of what is happening to their family, friends, and communities. In addition, they share their life online and express their personal feelings, opinions, and comments about anything they are concerned of. In the midst of this general acceleration in using social media, there is an increase in using it in education, in particular microblogging environments such as Twitter. As a result, there is an increase in the amount of data within these social media that reflects what people are learning and how well they are learning.

In this research we aim to identify and organize the various topics/subtopics– being discussed among people who are involved in learning activities using microblogging – by automatically analyzing the posted microblogs. This thesis is related to different research fields that we shall introduce in this section. First, we briefly introduce the use of microblogs in education. Then, we give an overview about Learning Analytics. Finally, Microblog Analytics is presented.

1.1 Miroblogs in Education

In recent years, auxiliaries that go in line with Learning Management Systems (LMS) [104](e.g. Blackboard and Moodle [64]) have begun to enter the field, with potential for

greater learner control and rich tools for media creation and sharing. Recent statistics [87] show that , 93.5 % of 18 year old and 95.4 % of 19 year old in the USA use social networking on a regular basis. Specifically the official 2013 Twitter statistic has shown that more than 85% of twitter active users around the world are in ages between 19 and 29 years old and 48% of users are in college. In addition 52.1% of academics in 2010 say that they have used Twitter and the ratio reached to 60% in 2013 [52]. Moreover, over 470 universities worldwide are using social networks such as Facebook and Twitter to communicate with students [14, 12]. The advantage of Twitter and microblogging as seen by the educators is in the possibility of giving immediate feedback and real time engagement [93, 72] , as well as offering the possibility to monitor learning process and the ability to track students' progress [95]. The limitation of 140 character of tweet forces the students to focus on the topic whereby they asking questions, giving opinions, changing ideas, sharing resources and reflection [40] . Twitter can enhance participate the conversation outside the classroom for a more sustained learning experience [48]. It is a powerful way to create a strong learning community that involves all students, especially those who are shy or less outspoken in a face-to-face in Mass class [39]. Twitter can provide students with a convenient way to express their feedback in free text [63].

1.2 Learning Analytics

In the field of *Formative Assessment* developing learning and teaching initiatives to improve retention and progression in education process is an important academic concern, which mainly depends on monitoring student performance and exploiting student feedback. Teachers and instructors need to uncover what and how well the student understands throughout the course of instruction. A teacher engaging in formative assessment uses information from a particular assessment analytically and diagnostically to measure the process of learning and then, in turn, inform himself/herself or the students of progress and guide further learning, and adjust instructional strategies in a way intended to further progress toward learning goals. Formative assessment makes both teachers and students aware of holes in knowledge or understanding, leading teachers to address specific content and provide additional learning strategies to fill in these holes, leading students to set goals and track their progress toward achieving them [21]. Learning Analytics **(LA)** is a research field that goes in line with the goals of formative assessment. It seeks to enhance the learning process through systematic measurements of learning related data, and informing learners and teachers of the results of these measurements, so as

to support the control of the learning process. The prime data source for most learning analytic applications is data generated by learner activities, such as learner participation in continuous, formative assessments. That information is frequently supplemented by background data retrieved from Learning Management Systems **(LMS)** [104]. However, the recent ubiquitous access to social networks like microblogs (e.g. Twitter) become a critical part of learners' online identity, and an expected part of learning platforms and analytic research as well. LA should as a result face the challenges of finding ways to capture and analyze learning data generated in social media networks. With the rapidly growing interest in and technical ability to leverage these data in educational settings, there is a sense that many recent educational technology and big data initiatives will provide education committee different point of view for what they know about learning and teaching. It will help educators better assess their pedagogical practices, and devise innovative educational methods, all with the goal of improving education.

1.3 Microblog Analytics

Microblog Analytics is a process of collecting data from microblogs and evaluating that data for making decisions. This process goes beyond the usual monitoring or a basic analysis of retweets or "likes" to develop an in-depth idea of the social users. From microblogs information, current affairs may be detected and public opinions may be extracted [68, 50] . For governments, a large amount of financial cost may be saved by online survey for public opinions; for news media, important events may be detected at early stage [9]; for commercial corporate, user comments may be collected efficiently so that advertisements can be put on more effectively. For twitter users, they can stay on top of news and events internationally or in specific regions, oftentimes before these topics even make it to the news. Twitter shows the reader what the most popular conversation topics on the microblogging site are right now by providing what so called Trending service [45]. For all these applications, understanding the content of microblogs text is essential to extract useful information and find value from social media conversations. Consequently, how to automatically understand, extract and summarize useful twitter content has become an important and emergent research topic [89] . Microblog Analytics faces many challenges. First it faces the challenge of dealing with informal text. A microblog is usually short and contains limited information; informal or oral language, full of slang, which makes the interpretation a very difficult task. Therefore, traditional text analysis techniques cannot work well as the case in the standard written language.

Alternative methodologies are being proposed and used for modern social media analysis in general and for Twitter in particular[59] . Second, analyzing microblogs content faces the challenge of dealing with big data. Based on recent statistics [26] , the number of Twitter users is 554, 750,000 and they sent more than 500 million tweets daily with the average of 58 million per day and 9,100 per second exploit several gigabytes of size per day. Apparently, it is an extremely large problem to tackle if a computer or agent has to access, process vast amounts of data, and then deduct context, common sense and domain specific knowledge presented at that massively unstructured content.

1.4 Motivation And Importance of Research

Formative assessment provides essential feedback about what students are learning and how well they are learning it. Teachers can quickly gather data to determine whether students are mastering the goals and standards or there are gaps in students' learning. Teachers can then use this information to change the content and products of instruction to customize learning based on a student's achievement of curricular goals. They can also establish priorities for future lessons. With the recent grow of using social media in education, in particular microblogging environments such as Twitter, there is a corresponding increase in the amount of data within these social media that reflect what students are learning and how well they are learning. For example, courses that adopt Twitter in their learning process allow students to discuss with each other and with their teacher different topics and express their opinions on various aspects of these topics. The data generated out of these discussions constitutes a valuable resource on which teachers can rely in order to conduct formative assessments. To understand this complicated landscape of topics that students are learning during the life time of the course, and how well these topics are being learnt, it would be helpful if there were a network that organized topics being discussed in these microblogs according to their various subtopics and stored information about expressed opinions on these topics. This envisioned network would provide essential feedbacks for the teachers that go in line with the goals of formative assessments. For example, it would provide a quick feedback about whether the majority of students class has mastered a specific topic and its related subtopics based on the volume and the polarity of the discussions on this topic. Having such feedback, the teacher may choose to use instruction time to delve into that topic further and explore an applied aspect of it, or may decide to reteach certain elements or assign more group exercises targeting a particular skill. These kind of networks evolve

over time in terms of number of topics, number of students' contributions, and polarity of opinions expressed on these topics. This means, such networks would provide the teacher with information that would help him/her track the learning process over time and to observe the strengths and the limitations in the learning process while it is happening. Consequently the teacher would be able to make decisions and to take steps in order to overcome limitations at early points of time. Figure 1.1 shows an example of the kind of network we envision. The network in the figure represents a course about "Operating System".

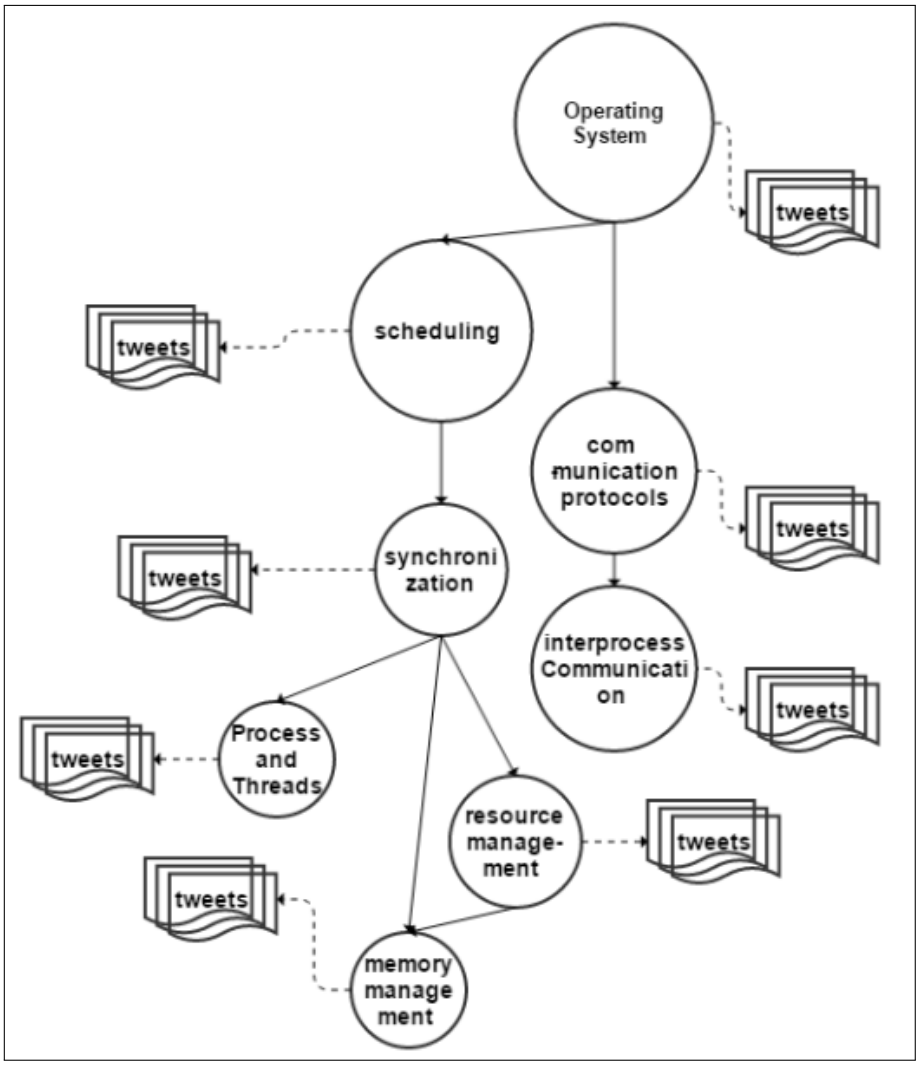

Figure 1.1 Assessment Network of Topics

The nodes in this network represent topics that students and their teacher are discussing over microblogging environment. These topics are organized according to their various subtopics. The size of each node reflects the number of discussions on the corre-

sponding topic. Each node is associated with all micro-blogs in which the corresponding topic was mentioned. The color of a node represents the overall polarity of the opinion expressed on the corresponding topic. A summary of opinions expressed on each topic is stored in its corresponding node. The challenge in building this network lies in the fact that the microblogs input is short and informal text, where it is difficult to spot phrases that denote topics and map them into a canonical representation. Moreover language diversity and ambiguity is another challenge. A topic facets that we aim to detect and organize according to their various subtopics can be mentioned several times with different wordings like ”binary search tree”, ”BST”, ”ordered BT”, ”sorted binary tree”, etc.. Thus, the entire problem can be seen as an interleaved task of topic extraction, and entity disambiguation.

To the best of our knowledge, there is no prior work within the fields of Learning Analytics and Microblogs Anayltics that addresses the problem of mapping microblogs data generated during learning activities into a networks of topics, in a similarly comprehensive manner.

1.5 Problem Statement

Given a set of microblogs posted by people who are involved in learning activities using microblogging, the problem we try to tackle in this research is how to identify and organize the topics being discussed among collaborators according to their various subtopics in a way that can reflect the extent to which a particular topic/subtopic is being discussed among them.

1.6 Objectives and Contributions

In this research, we map microblogs generated in learning activities into a **network of topics** which would provide essential feedback about what topics are being learnt by people in learning and how well they are being learnt. This main objective implies some specific objectives in the following points:

- Data Set: We aim to collect microblogs posted by people involved in learning by adopting microblogging.
- Implementation : We aim to use divers kind of techniques and methods rooted in Text mining, Machine learning and Microblogs Analytic to implement our model

for mapping microblogs generated during the learning activities into networks of topics. We denote this framework **MAMINT**.

- Demonstration: We aim to demonstrate the usage on in an implemented prototype of **MAMINT**.
- Evaluation: evaluate the accuracy of the generated networks of topics.

The main contributions of this research are:

- Using microblogs generated during learning activities as a source for identifying and organizing topics/suptopics being discussed among people.
- Building a scalable framework for collecting, monitoring and analyzing microblogs generated during learning activities. This framework **MAMINT** utilizes and combines methods from different disciplines such as Text Mining, Social Media Analytics, Topic Identification, Learning Analytics, and Big Data.
- Constructing a network of topic for a specialism that provides essential feedback about what topics are being learnt by collaborators and how well they are being learnt.

1.7 Scope And Limitation

Our research focuses on the problem of identifying topics in Twitter data sets and on organizing these topics according to their various facets in order to build the proposed network.

- In our study, we only consider tweets posted in English language.
- We only evaluate the accuracy of the generated networks of topics. Evaluating the usefulness of the generated networks in courses assessments is not possible for two main reasons, 1) in order to build the envisioned network as it has been described in the **Motivation** section of this thesis, we need to conduct further research on opinion mining and sentiment analysis which is out of the scope of this research, but it is the subject of future research. 2) in order to conduct a real formative assessment on teaching courses, the project need to be under test for a considerable period of time, which exceeds the time period of our research.
- Opinion mining and sentiment analysis are out of the scope of this research.

- Furthermore, we identify topics from static tweet data-sets text, real-time topic detection or *emerging* topic identification within a window time is not our focus.
- This research only focuses on analyzing microblogs posted using Twitter. Other posts generated in different social media tools such as Facebook, MySpace, will be considered as a future work.
- This research is highly recommended for Higher education systems.

1.8 Thesis Outlines

The thesis document is structured as follows: In Chapter 2 we present the Literature Review conducted in the thesis fields. We present our proposed approach in Chapter 3. The implementation of the applied part of the thesis is presented in chapter 4, while Chapter 5 is about the development and testing experiments and their evaluation. We conclude the thesis in the final Chapter 6.

2

Related work

In this chapter we introduce the *Literature Review* related to the scope of our research. We review prior works and cite examples in the fields of Topics Identification in Tweets, Topic Organization and Learning Analytics.

2.1 Topics Identification in Tweets

Extracting key phrases from text is an essential step towards content understanding. However, extracting these key phrases from microblogs faces the difficulty caused by the brevity and the noisy of the text in these microblogs. Different research works addressed this problem.

For example the study presented by Achananuparp et al.[71] is one of few works that deals with key phrase extraction from microblogging. They proposed a Page Rank (PR) [103] method and probabilistic scoring function for keywords and key phrase ranking. Candidate keywords are ranked and selected based on a topic specific PR algorithm that considers the topic associated with each tweet obtained using Latent Dirichlet Allocation (LDA) [23]. The top-k keywords of each topic are picked, and combinations of these keywords that occur as frequent phrases in the text collection are identified. A probability scoring function is used to ranking the final key phrases for each topic.

Li et al. [53] developed a keywords extraction methods for social snippets in the form of a classification task. They use a sample of Facebook posts in their experiments. They tokenize each post and generate unigrams and bigrams to be considered as keyword candidates. They calculate a set of features (TF-IDF,lin,pos,lenText,DF,capital) to each candidate. Then train a classifier based on the labeled keywords. Based on a threshold they select highest top words as a keywords of the sample posts.

Brendan et al. [69] presented a search application for Twitter TweetMotif. Unlike

traditional search information retrieval methods, TweetMotif responds to user queries with a set of tweet messages grouped by specific key-terms. They use the Twitter Search API to retrieve the matched messages then group and summarize these messages into topic phrases. Topic phrases are identified by a simple language modeling approach which are most distinctive for a tweet result set, scoring them by the likelihood ratio.

Due special characteristics of short text messages a.k.a Microposts :1) limited length (up to 140 characters in Tweets) restricting contextual information, 2)the noisy lexical nature of them where new terminology and jargon emerges as different events are discussed 3) and the large topical coverage of Micropost ,Some approaches try to address these challenges by proposing the use of external knowledge-sources. These sources provide additional textual data on wide growing number of topics, which can alleviate the sparsity of Microposts's content and incorporating additional contextual information necessary to effectively understand and disambiguate them. Disambiguating entity references by annotating them with unique ids from a catalog is a critical step in the enrichment of unstructured content that many research focus on. Wikipedia has been proposed and used as a comprehensive catalog for entity disambiguation in many research works.

Tweet terms are linked into Wikipedia pages as described by Oard et al. in the work in [67]. The Wikipedia link structure was utilized to find the similarity between different tweets as a clustering feature. Tweet terms (composed of multi token) that occur at article title or used at least once as anchor text are considered as featured terms in addition to all non-stop words single token of tweet terms. Using Wikipedia, they derived concepts to identify featured terms. The probability of an article to be used as destination for a term is the measurement used to retrieve a list of candidate Wikipedia articles. The decision of select the appropriate Wikipedia article to represent a term is taken by calculating overlap rate between term context (all tokens within the same post) and the content of Wikipedia article. The "similar" posts were clustered using cosine similarity matrix. Finally, each cluster was labeled by most frequent used concept and donated as group topic.

Michelson et.al [62] treated all capitalized, non-stop words as candidate named entities, and leveraged Wikipedia as a knowledge base for disambiguation based on the context (words) around the discovered named entities. Once disambiguated, the entities were mapped to the categories contained in "folksonomy", a Wikipedia user-defined category tree, which were finally treated as users' topics of interest. Their experimental results showed that the usage of external knowledge bases such as Wikipedia could significantly empower entity disambiguation and category matching to generate reasonable topic profiles for Twitter users. All non-stop words are considered key phrases

likely would match Wikipedia content.

An approach for tweet classification into topics is described in [66]. It maps tweets to a large context using Wikipedia pages. They calculate the semantic distance between tweets based on the distance of their closest Wikipedia pages. For each tweet, non-English words and stop words are eliminated while closest Wikipedia page is identified for each remaining word. The topics are discovered by traversing the tree structure of the Wikipedia taxonomy. A list of pages is retrieved for each word, the page with the highest word occurrences is associated with that tweet. Using Latent Semantic Analysis technique [51], the distance between two tweets is calculated to interpret the underlying relationship and the category membership.

To determine what a micro-blog is about, authors of research described in [60] proposed a method to automatically identify concepts it post. The concept is an item that has unique and disambiguated entry in a well-known large scale knowledge-source "Wikipedia". They map the tweets to Wikipedia articles in two steps i) obtain a raked list of candidate concepts from tweets ,and ii) a good precision method to determine which concept keep by exploring different lexical features. In order to determine which part of tweet is the source for a semantic link, they generate all possible n-gram from tweet. then train a classifier to determine if the words is vial concept or not by using a training set of tweets n-gram of tweets and associated concepts. The labeled concepts are Wikipedia pages titles. The researchers conducted a comprehensive comparative study for the usage of different lexical features to obtain high-recall concept ranking and high-precision concept selection.

Abel et al. [7] aim to contextualizing tweets in a very similar task. After adding context, authors used the tweets to profile Twitter users. Their approach depends on matching tweets to news articles, followed by semantic enrichment based on the news article's content. Finally, the semantically enriched tweets are used for user modeling. For the second step –semantic enrichment– the authors use OpenCalais [1].

Mendes et al. [61] proposed *Linked Open Social Signals*, a framework that includes annotating tweets with information from Linked Data. Their approach is rather straightforward and involves either looking up hashtag definitions or lexically matching strings to recognize (DBpedia) entities in tweets.

Within the Framework described in [18], authors offered an extension to exploit category meta-graph (a semantic graph) to drive a set of semantic features and comparatively evaluate their usage to enhance performance of a topic classifier. Afterward they

[1] An annotation web services, http://opencalais.com/

measured the conceptual similarity the KS documents and Microposts, considering the enriched representation of these documents. This work indicates that the accuracy of a topic classifier can be accurately predicted using the enhanced text representation.

Given the less textual information of micro-blogs and the implicit interest expression of microbloggers, authors of the work described in [28] employed the contextual enrichment idea by Wikipedia. They proposed a semantic spreading model (SSM) that leverage graph constructed by Wikipedia, they discover the semantically related interest tags which do not occur in micrposet.

In research presented in [38], Wikipedia is used for linking entities in micro-blogs. To overcome the short and noisy of micro-posts they propose Context-Expansion-based Microblog Entity Linking (CEMEL) approach to extend or enrich post context by similar posts. Given a collection of posts, they constructed a query from single post and search the collection. Most similar posts of search results are used during disambiguation for context expansion of original post as a kind of annotations. The researchers also proposed *Graph-based Microblog Entity Linking* (GMEL) method which depends on constructed neighbor nodes of similar posts to a post (node) instead of added directly to a context. The linking accuracy significantly improved by their proposed method by 8.3% and 7.5% respectively.

Authors of the research published in [36] employed Wikipedia to easily classify tweets into four categories: Locations, People, Organizations, and Miscellaneous. They normalized tweet before match its terms for Wikipedia pages's titles using Wikipedia native search APIs. The normalization tight in two ways: i) Extract noun phrases from tweets by considering tweet a regular sentence and use NLP tools for this purpose. ii) Extract n-gram (up to 4) from each tweet. Matched words are the candidate concepts/entities to be classified. The research utilize the Wikipedia graph structure to extract Wikipedia categories tagged in pages matched. They label container-categories with the entity label from the contest (Locations, People, Organizations, and Miscellaneous). For each tweet concept that matches a Wikipedia page title, they traverse up the page's category graph and count how many of the categories within 3 levels of the original page fall immediately under a labeled container-category. Then label the tweet with the container label that holds the maximum number of the categories from the page's category graph. This research is actually an extension to authors's pervious works that immerse the same idea (see their publication on [35] and [37]). Researchers of published work in [33] and [88] pursuing similar goal.

The researches of works described in [80] and [98], assumed the existence of trending

or classification topics in order to efficiently detect tweets that are related to them, despite not being explicitly marked as so. In our work we are interested to *Topic Detection* that uses a wide variety of techniques regarding text analysis to find the most common related words and hence detect.

The author of the work described in [19] extended the topic detection methods to not only detect the novel topics in static tweet text but also include *emerging* topics that will be hot and viral in near future for specific organization.They collect a sample of user generated tweets related to specific organization. Yet they use SVM classifier to filter out the irrelevant, noisy tweets using a training set of tweets generated from known organizational Twitter accounts. They use un-supervised method for topic detection to handle live and large volume of tweets without any prior knowledge of the number of topics . most probably emerging topics are constantly evolving and growing in size. Single-class clustering algorithm is employed for a time interval . The authors then proposed a hot emerging topic learner to infer the importance of topics for specific organization based on user/topic features.

Commonly the researches conducted on emerging topic detection mainly focused on keywords and textual content, whereas this one [79] find emerging topics with respect to an organization. The major difference is that for entities like organizations, in addition to textual content, user association to the organization and social relations among users of the organization used at the detection of emerging topics for the organization. Variants of PLSA and LDA have been proposed for online and dynamic topic modeling to detect emerging topics [13].
Because of tweet length restriction, researchers began to utilize Topic-Models (e.g. LDA) which are powerful tools to identify latent patterns in textual content. It is often used to discover topics in which Twitter users are interested for recommendation services or search agent preferences.

The authors of work described in [15] compared different LDA models based on three metrics: MSG, User, and Term, to study the performance of best one for twitter usage. The first model defines each tweet as a separated document, and LDA was trained to extract the topics of each tweet. Then, topics extracted from all tweets posted by the same user were aggregated to serve as the interest of that user. The second one aggregates Tweets posted by the same user into a document, and LDA was trained to extract the topics of that document, which were treated as the interest of that user. The last model aggregates Tweets containing a particular term into a document, and LDA was trained to extract the topics of that document. Then, topics extracted from all terms contained in

a user's tweet vocabulary were aggregated to serve as the interest of that user. In their experiments, the User metric was revealed to be best performing.

A tweet topic-based recommendation algorithm is described in [27]. The algorithm decides whether an incoming tweet will be interested for a user depending on whether the topic of this tweet is relevant to the topic model established for the user according to his/her posting history.

Similarly, TwitterRank [78] is another system that relies on tweet topic modeling. The goal of TwitterRank was to identify influential microbloggers. First LDA was to build "topic model" for each author based on all tweets posted by that author. Then they compared queries with each author topic model to find the most relevant author. Author tweets captured in a single document then leverage LDA to discover the topics over that document. Such approach has a problem of LDA result, since the twitter data is sparse, and the generated topics are based on user mode rather than actual concepts.

As a step towards improving methods for following new users and topics, and for filtering feeds, a study was conducted and later described by authors in [55]. They statistically studied the user following behavior and come with four dimensions (substance, social, status, style)for any tweet post. By using Labeled LDA, they map a set of posts into these dimensions. Furthermore, features extracted based on these general dimensions in aim of characterizing every user by the topics they commonly use in their posts. By doing so, they could personalize feed re-ranking and user suggestions.

Shin et al. [86] proposed a graph-based approach for detecting persistent topics (PT) from Microposts, which correspond to topics of long-term, steady interest to a user. For their graph based approach they introduced two novel scoring functions that measure the properties inherent to PT terms: regularity and topicality. They allow to distinguish between terms that represent persistent topics, and terms which appear in static documents. Experimental results showed that this approach outperformed other existing alternatives (including LDA and keyword extraction models).

2.2 Topic Organization

Organization of information is obviously an important issue that many researchers tackle in the field of data and document management or classification. One of the earliest methods of organizing data is creating a hierarchy or taxonomy from the data to reflect topical relations. One of the most popular techniques for hierarchy construction is text clustering, which groups similar key-words or documents together in a hierarchical

fashion. In [20] authors applied a hierarchical clustering algorithm on web pages resulted from search queries to produce a taxonomy instead of a linear list of ranked pages that commonly produced by search engines which may be insufficient for many applications. Then, they applied top-down partitioning to generate a multi-way-tree taxonomy from the binary tree.

Divisive form of Hierarchical clustering is used by Wang et al. [101] for the same purpose. They proposed a term co-occurrence network to group terms that highly co-occur with each other to the same topic divisively.

Other methods used word sense disambiguation to discover the relations between topics yet create paths between them. In the work cited in [96], authors use WordNet [97] for this purpose. They used LDA to find the candidate topical concepts, then lookup these words in the WordNet to find their context. They benefited from "hypernym" relation to find the super ancestor between each two topical words. The discovered relations between words used to enhance multi-way hierarchical agglomerative clustering algorithm since it may not show these relations between words appropriately. They evaluated their method by comparing the similarity between a target hierarchy and the constructed one. Evaluation shows that the combination using multi-way clustering and WordNet get the highest similarity score between the two hierarchies.

Since most of the these techniques rely on a single type of data – usually text, the problem with text-only hierarchy induction is that words often have multiple meanings. Therefore without proper context proper taxonomy induction can be difficult. Most document repositories contain linkages between the documents creating a document-graph. These links provide proper context to the terms in each document. Document-graphs are especially common in nonfiction and scientific literature, where citations are viewed as inter-document links. This ideas is utilized by authors of the work described in [102].

2.3 Micro-blog Analytics for Research And Education

”The simplicity of publishing such short updates in various situations . . . makes microblogging an innovative communication method that can be seen as a hybrid of blogging, instant messaging, social networking and status notifications” [81].

There are different kinds of analytics in online learning platforms. For example the analytics dashboards focus on rendering data logs via a range of graphs, tables and other visualizations, and custom reports designed for consumption by learners, educators,

administrators and data analysts [56, 44, 41, 42, 46, 57, 43, 22].

Discussions on microblogging platforms provide vast amount of data on various topics of social importance, which has helped to establish microblogging as a rich resource for academic research. A recent meta-analysis of 575 peerreviewed publications that used Twitter microblogging data identified the wide range of science domains to which these studies belong: geography, marketing, natural disaster management, linguistics, politics, and many others [109].

There are numerous free tools for interactive visualization and analysis of networks [70]. One tool specifically designed for learning networks is SNAPP [10] which renders discussion forum postings as a network diagram to help trace the growth of a cohort, identify disconnected students, or visualize how teacher support is employed within the network. Another is NAT, designed to help teachers see their offline social networks, which annotates social ties with the relevant topics [83].
Predictive analytics is another kind of analytics. These analytics seek for example to classify the trajectory that the students are on (e.g. ”at risk”; ”high achiever”; ”social learner”) from the pattern of students' static data (e.g. demographics; past attainment) and dynamic data (e.g. pattern of online logins; quantity of discussion posts), and hence make more timely interventions (e.g. offer extra social and academic support; present more challenging tasks). The design of more complex data-driven predictive models must clearly improve on the prediction of final exam results, but requires statistical analysis to identify those variables in the data that can be historically validated as being the strongest predictors of ‘success’ [11, 91]. Discourse Analytics and Text Mining go beyond simple quantitative logs, and provide feedback to educators and learners on the quality of the contributions, seek to classify these contributions into topics, and to analyze their subjectivity level and their polarities (positive/negative). Researchers are beginning to draw on extensive prior work on how tutors mark essays and discussion posts, how spoken and written dialogue shape learning, and how computers can recognize good argumentation. Also researchers build on prior work on topic models [23], named entity disambiguation [24], opinion and sentiment analysis [73] in order to design analytics that can assess the quality of text and its type, the ultimate goal of improving students learning performance. Discourse Analytics and Text Mining specifically tuned for learning [91, 100] or sensemaking in contested domains [99, 54] are at the stage of research prototypes.

2.4 Summery

None of the previous works focused on mapping the micro-blogs into a network of topics in comprehensive manner as we propose in this research. Our approach for topics identification adopts and combines ideas from pervious works in the topic and updates these ideas and develops them to match the requirements of this research.

3

Proposed Approach

Our approach for mapping micro-blogs on specific hash-tags into network of topics consist of a number of phases as illustrated in Figure 3.1. In the next few sections we describe each phase in more details.

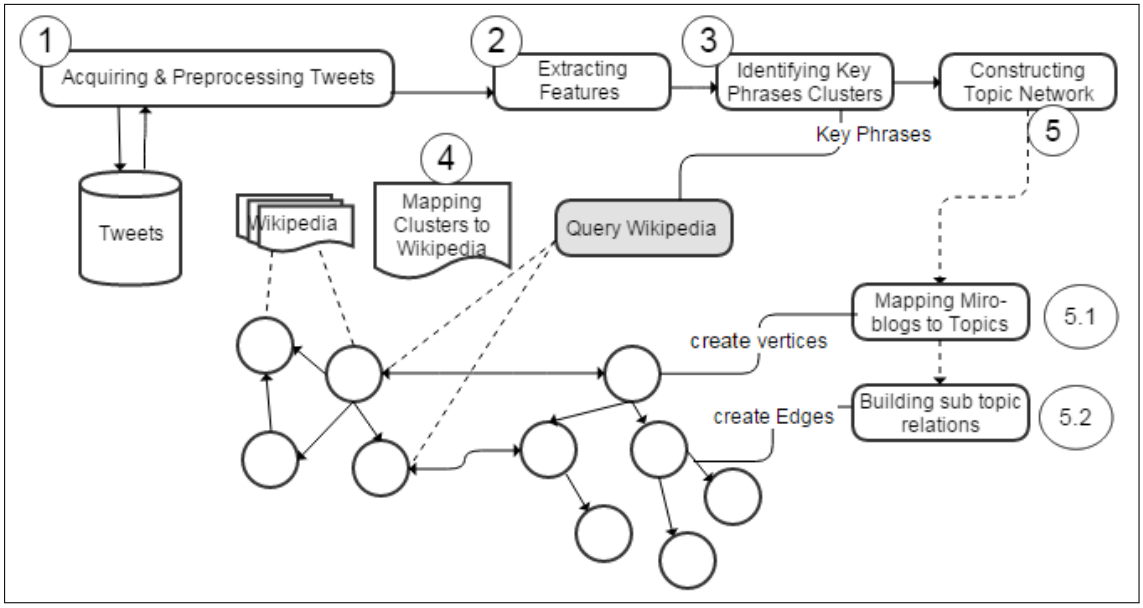

Figure 3.1 MAMINT Framework

3.1 Acquiring And Preprocessing Tweets

In this phase our prototype first tracks and aggregates tweets from twitter stream based on different "hash tags". A hash tag is a line of text that Twitter can use to track any tweet with that specific hashtag. We assume these hash-tags are provided by the course instructor during course preparation and setup.

Pre-processing techniques are necessary, to acquire a more clean data-set [49]. These techniques often involve transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Pre-processing techniques are usually used to overcome these issues. The Preprocessing strategy we follow includes the following steps (see

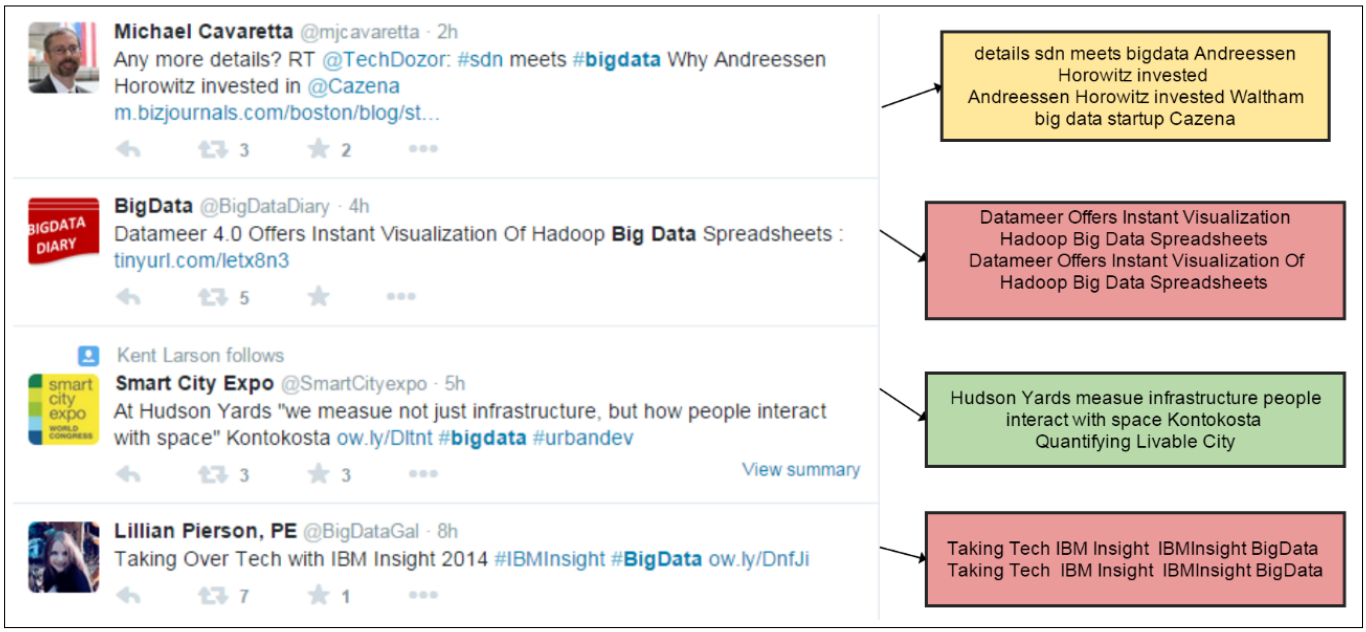

Figure 3.2 Preprocessing Tweets

Figure 3.2 for some examples):

- We discard a tweet if it contains less than four distinct words or if it has in common 60 characters or more with another tweet – most probably one of them is a re-tweeting of another.
- Tweets with non-English characters will be eliminated. If a tweet contains no English words or become less than four words after removing other characters, we discard the whole tweet from any further analysis.
- Emoticon – punctuation mark, which usually expresses a person's feelings or mood, will be removed.
- Stop words are by definition common words, which need to be removed from text in natural language processing. In our approach we also remove stop words, since they do not have much influence on the context. Meanwhile, there is no definite list of stop words. All the dictionaries have therefore to be selected manually from

different resources [1]. In addition to English standard stop word list , we consider some common words that are used in social networks and give no meaning such as: Re-tweet, follow, share, RT and other similar words.

- More Twitter specific pre-processing techniques based on Twitter orthography can be used to filter out the special features (hash-tags, user mentions and retweets). For example, usernames are removed since they have a large impact on the dictionary word reduction. The diversity of usernames causes a lot of new words, therefore leaving out the usernames results in a great reduction of the data and the featured vectors as well. In our approach, user name and RT user name have been discarded.
- Hash-Tags are considered as features and appended to the corresponding tweet after removing # character (i.e. #bigdata become bigdata)
- The title of any reference page is extracted from its URL if it appears in a tweet.

3.2 Extracting Features From Tweets Collection

In this phase, different kinds of features (e.g. unigrams, n-grams, hash tags) are extracted from tweets and vectorized in order to fed them to the next phase for further processing. These features are used to identify the key phrases in a given set of tweets. In other words it is the task of preparing the dictionary space that will be used by topic detection algorithm.
The set of all words that encountered in the given series of tweets are vectorized. The dictionary space is the set of all words that appear at least once in any of the tweets collection. Each word being assigned a number, which is the dimension it occupied in tweets vectors. A tweet's vectorized form consists of the number of times each word occurs in it.The value of the vector dimension for a word is usually the number of occurrences of the word in the document (tweet). This is known as term frequency (TF) weighting [82]. **The term frequency** (tf(w, d)) is the number of times a word w occurs in a document d, stated in Equation (3.1)

$$tf(w,d) = |w|, w \in d \tag{3.1}$$

This measure is used in most text applications for full documents and articles retrieval and classification; however, short and noisy tweets are different. Tweet has maximum 140

[1] https://github.com/arc12/Text-Mining-Weak-Signals/wiki/Standard-set-of-english-stopwords

characters and on average the number of words in a tweet are around 15 [76]. Typically, in twitter domain frequent words are not necessarily good features; since most term frequencies will usually be 1. More precisely, the number of unique words that appear in one tweet is typically small compared to the number of unique words that appear in any tweet in our processed collection. As a result, these tweets vectors are quite sparse. **Term Frequency–Inverse Document Frequency (TF-IDF)** [82] weighting is a widely used improvement of Term-Frequency weighting; instead of simply using term frequency as the value in the vector, this value is multiplied by the inverse of the term's document frequency. where the **document frequency** (DF) is df(w, D) of word w across all documents D is defined as follow:

$$df(w,D) = |d|, d \in D : w \in d \tag{3.2}$$

In Twitter domain the count of a word is almost the same as the count of messages it occurs in (DF and TF are the same). The **inverse document frequency** (IDF) is used to measure the rareness of a word across all the documents. The higher the value of the inverse document frequency , the more rare the word across the set of documents is. The inverse document frequency of a word w across all documents D is shown in Equation 3.3

$$idf(w,D) = \frac{\log |D|}{df(w,D)} \tag{3.3}$$

The inverse document frequency is combined together with the term frequency to in one Equation measure shown in Equation 3.4.

$$tf - idf(w,d,D) = tf(w,d)idf(w,D) \tag{3.4}$$

Usually the value computed by TF-IDF measure is higher for words that are less frequently used across all documents in the data-set. IDF treats features independently and computes the likelihood that a document would contain such features. As a result, the important words, or the topic words, usually have high TF and IDF values. However In twitter domain, for TF-IDF weighting the tweets are the documents , which means that words with a high (TF) frequency within the tweet and a low frequency over all tweets have a high TF-IDF value. These words do not seem to be the best words to use for generalization, since they do not cover many tweets. Therefore the TF-IDF measure is not efficient as feature selection for topic detection. Moreover, TF-IDF weighting assumes that words occur independently of other words, but vectors created using this method

usually lack the ability to identify key features of documents, which may be dependent. For example, the word *"big"* will most probably co-occur along with the word *"data"* in tweets collected for the topic ""cloud computing"" . Moreover *"cloud"* and *"computing"* words are not completely independent. So, we have to identify groups of words that have unusual high probability of co-occurring together. Therefore we extend the feature space to include word **N-gram** features. Word N-gram model [16] has the role of identify a contiguous sequence of n words from a textual or spoken source. In case of unigrams (n = 1), each text (tweet) is a document and is split up into words. Counting the frequency of all the words in a all documents results in a word frequency table. Extending the feature space to higher order word n-grams, will split the texts into N length words. In case of n = 2 (bigrams), the feature consists of one or two consecutive words in the original text. The same holds for n = 3 (trigrams), and higher n values. Intuitively, high frequent words have a higher probability to appear in texts, therefore those words describe the data-set better. In our work, the vectors dimensions are mapped to **bi-grams**.

linguistic features are commonly considered in Text Mining. In most cases, nouns have higher probability to be keywords [85]. Such a feature make no sense with the “short” and “informal” text of tweets.

Capitalization is another insufficient feature for tweets or micro-blog [84]. It is unapplicable to identify upper case words as featured terms within the informal tweet nature.

On the other hand, Position Weight (PW) score considers the position of words in the article [34]. It emphasizes on the position feature; therefore, it suits the structural document that has clear title, abstract, subtitle, and conclusion and so on, and therefore it is not be appropriate for unstructured documents like blogs and microblogs.

The output of this phase is a dictionary vector space representing the tweets collection (see Table 3.1 for an example).

Term	TF	DF	Tweet1	Tweet2	Tweet3
secure	2	1	1	0	0
enterprise	1	1	0	0	1
database	1	1	1	0	0
python	1	1	0	1	0
system	1	1	0	1	0
sport zdnet	2	2	1	1	0

Table 3.1 Extracting Features From Tweets

3.3 Identifying Key-Phrases Clusters

In this phase, we identify a set of key-phrases clusters that would likely express a topic or a subject. A topic describes a high-level human concept, and can be represented by a set of key phrases. For example (students, education, cr, learning, student, free, courses, class, university, higher) are key phrases for a topic that a human can interpret as "Higher Education" or simply "Education". The key-phrases clusters that will be identified in this phase, will be later used to identify the topics/subtopics for the tweets collection under consideration.

Most used key phrases extraction algorithms are based on supervised learning [90]. However, these algorithms requires manually labeling a large data-set which is time consuming and expensive. On the other hand, clustering is an unsupervised method which has also been used for key phrases extraction in large corpus of unseen documents [8]. It identifies clusters of documents (tweets in our case) on the same topic; however, these clusters are unlabeled. The clustering is done by the aid of featured terms without any prior knowledge about topics or possible categories that clusters will belong to. The cluster is identified by a list of keywords describing the topic. These are the key phrases we aim to identify.

Topic modeling techniques are generative models like Dirichlet clustering. It considers topic as a mixture of words with a probability [23]. The list of values represents an individual topic and different topics will have different probability values associated with each word. This way of representation makes this technique a robust one for key phrase extraction and clustering. Furthermore, it considers the document as mixture of words or "bag of words". In our approach we model the tweets using topic modeling as these tweets lack formal grammatical structure. It is a robust method for generating a document's topic mixture. This feature allows a model trained on an existing corpus to identify the topic mixture of new documents without re-training the entire corpus. Latent Topic Model (LDA) is a powerful Topic modeling because it can both cluster words into topics and documents into mixtures of topics.
For each document in the data-set, it generates words in two stage process:

1. Randomly choose a distribution over topics.

2. For each word in the document

   (a) Randomly choose a topic from the distribution over topics in step #1 .

(b) Randomly choose a word from the corresponding distribution over the vocabulary [2].

This process is run repeatedly until the model starts explaining the documents better – more precisely, when the sum of the probabilities stops changing. The list of words of a topic with high probability values form an interesting cluster of words to examine because they co-occur in the topic distribution (see Figure 3.3).

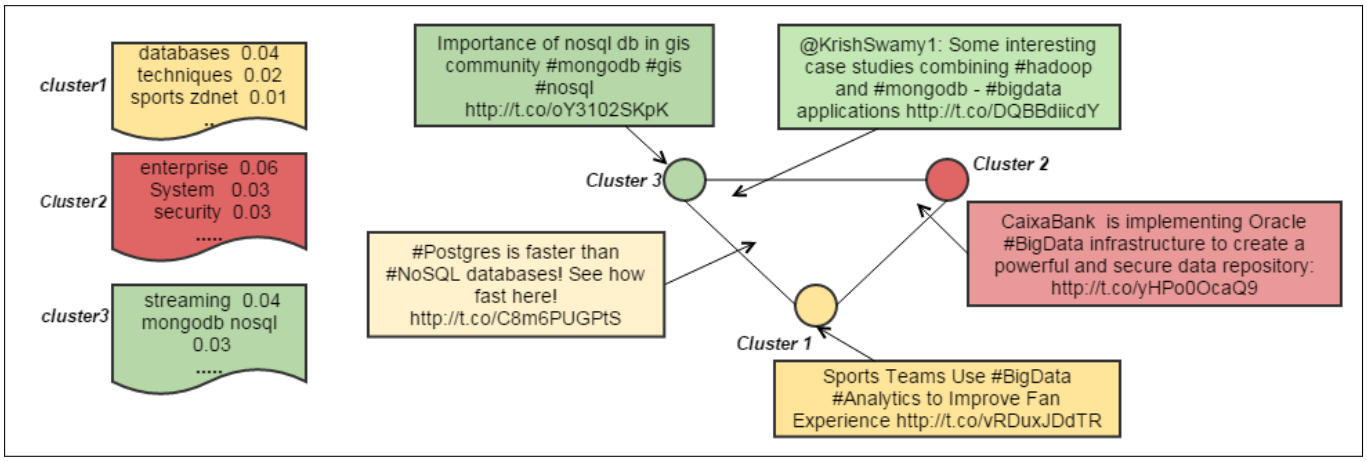

Figure 3.3 Topical key-phrases & Tweets clustering

There are several parameters that we must specify before estimating a topic model, the most significant one is the number of topics. The number of topics is a subjective selection that depends on the size and the shape of the corpus. In our case (tweets corpus), the lower value results in a large number of topics and higher value may result in limited number of topics. Optimal value for this parameter can vary depending on the given use case. In our approach, to estimate the number of clusters, we use a heuristic based on the number of categories and sub-categories of the corresponding hashtags used to collect the tweet data set. For example, if we used three hashtags (data science, cloud computing and, big data) to collect a data-set of tweets, we can find that corresponding Wikipedia categories with similar titles in Wikipedia are: "Information science" ,"cloud computing" ,"big data". Furthermore, we can also find that the distinct total number of sub-categories of these categories does not exceed 20. As Wikipedia reflects the wisdom of the crowd we assume that the total number of sub-categories in Wikipedia is a good estimation of the number of clusters.

[2] vocabulary: is the collection featured words which unigrams, bigrams or n-grams that have been tokonizd, filtered, scored and counted

3.4 Mapping Clusters to Wikipedia

In order to ensure uniformity in referring to semantically equivalent key phrase (e.g., “binary search key” and “sorted binary tree” are equivalent), we need to automatically derive topics. Most approaches are focused on how to best represent content using only evidence that can be found directly in that content or generalized automatically from that content. However, the short text we have and the common use of informal language tends to increase data sparsity and the possibility of finding lexical evidence that could be generalized as concept representing the content becomes relatively low [89].

In our research, we propose a method that makes use of existing knowledge bases in order to identify topics. In particular, we propose a mapping technique, based on a set of topics represented by the set of articles from Wikipedia.
Wikipedia contains over 3.7 million English articles that are neatly structured through inter-wiki links [107] and can be used to characterize the topical relationships in a manner that complement lexical existence. Moreover, the wiki pages and articles are mapped to categories. Briefly, given the size and diversity of English Wikipedia, we posit that the vast majority of (coherent) topics or concepts are encapsulated in a Wikipedia articles. By making this assumption, the difficult task of mapping Key-Phrases to potential topics is transposed to finding relevant set of Wikipedia articles, and using its direct categories and their sub-categories as candidate labels for the key-phrases clusters generated in the previous phase.

The *relevant* document retrieved by any word-based information retrieval method for specific query is a document with a high likelihood that the query key-words has matched the *bag of words* of that document or there is some kind of similarity between both. Considering document’s content as bag of words is to create a dictionary of terms present in document disregarding grammar and even word order. This dictionary forms the feature space of document where text is represented as a vector of numbers instead of its original string textual representation. The vector represents the importance of a term or even the absence or presence (Bag of Words) of it in a document. Vector Space Model (VSM) [82] is the algebraic model that represents the term extracted from the document into these kind of vectors in what so-called *"indexing process"*. The emphasis here is on the words space of document to use and the weighting of the indexed terms. It is obvious that many of the words in a document do not describe the content, for such, words like stop words or function words. In document indexing those non significant words are removed from the document vector, so the document will only be represented by content

bearing words. To find similarity between document and query in VSM model requires two important pre-steps 1) Convert the document and query string into vector of terms. 2) Find a proper calculation method to determined the similarity between query vector and document vector using associative coefficients based on the inner product of both vectors – based on overlap between two vectors –. For example, Cosine similarity [65] measures the cosine coefficient, which is the angle between two vectors. As a consequence, cosine similarity between a query vector q and a document d vector can be used for scoring the relevance of the document for that query (see Equation (3.5)).

$$SimilarityScore(d,q) = \frac{\vec{V}(q) \cdot \vec{V}(d)}{|\vec{V}(q)|.|\vec{V}(d)|} \tag{3.5}$$

The resulting scores are then used to select the top-k relevent documents for a query. By calculating similarity, each document scored based on query vector. Documents with highest scores supposed to be the most relevant documents to the associated query.
In our research we apply the VSM model. A collection of Wikipedia articles represent the document and is used to create the index repository; Figure 3.4 shows roughly the process for creating an index from Wikipedia articles text and their titles.

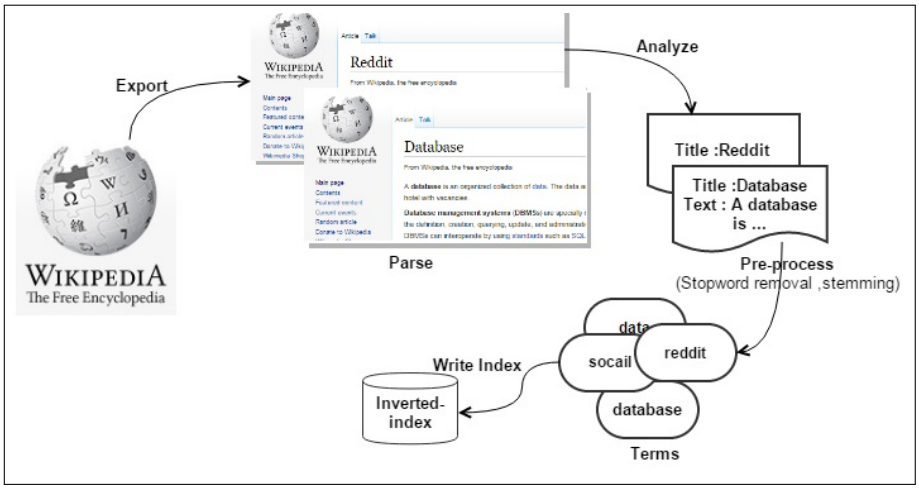

Figure 3.4 Wikipedia Indexing Process

In our research, we need to investigate two core issues: 1) The best method to structure the queries from the key-phrases clusters we generate in previous phase. Most, if not all, of the linking methods assume that the input text is relatively clean and grammatically correct and that it provides sufficient context for the purposes of identifying concepts and matching documents which is not the case in our context. 2) The best method to score any matching between the query phrases and indexed repository to find the most relevant

document. This is influenced by a number of factors: the vector words in the index, the method used to tokenize and store the text in the index, and the measurement used to score the relevance of a document.

3.4.1 Structuring the Query

In order to structure the query, we select the top-10 terms from each key-phrase cluster (based on the probability values obtained from the original LDA model). However, the number of relevant Wikipedia documents retrieved, using the terms as they are without any further processing may be limited or zero due to the problems of brevity in Twitter domain. For instance, "cloud computing" usually written "cloudcomputing" in tweets which will not have any match to the formally written Wikipedia articles. To relatively solve this problem we propose decomposition method; for each term that has more than five characters, we assume it is a compound term and should be decomposed to its original set of terms. Each term should not be less than two characters after decomposition. We use a dictionary-based filter for this task. The dictionary is a collection of indexed terms of Wikipedia articles. So, "cloudcomputing" would be decomposed into "cloud" and "computing" only if there are "cloud" and "computing" terms at the dictionary. We do not provide any new characters, so "compute" and "computer" will not be generated. A cluster of key-phrases and its expanded terms from the decomposition process form a basic query.

A query Q is seen as a set of n words w : $Q = w_1, w_2, w_3, \ldots, w_n$. A basic query will generate a set of alternative queries to be matched with the index generated from Wikipedia articles. These alternative queries are structured so that they respect the structure of the generated index and at the same time utilizes the default query structure supported by the indexing tool and overcome its limitations. They are described in what comes next:

1. A query is constructed from the individual terms in each index fields (default); search for each single word in **page title**:

   $Q_1 = w_1 \quad OR \quad w_2 \quad OR \quad w_3 \quad OR \ldots \quad W_n$

   This type of query will match any documents in which at least one of the query terms appears.

2. The default query structure as described in pervious point is not suitable to be applied when matching the query with the page text. The number of irrelevant

documents returned for such a query will be high. Therefore, we use an alternative structure based on AND operation. The query structure is as follows:

$Q_2 = w_1 \quad AND \quad w_2 \quad AND \quad w_3 \quad AND \ldots \quad W_n$

3.4.2 Scoring Wikipedia Articles

Structuring the query as explained in Section 3.4.1 allows us to score each document by sum-up the matching scores for each single query generated from each key-phrase cluster. The scoring strategy we follow is based on the idea that more weight is given for documents in which different terms of the query appear than the weight given to documents in which few terms of the query appear many times. The approach we follow also gives more weight for documents in which the query terms appears in the titles of pages over the weight given for documents in which the query terms do not appear in their titles. This is because matching page title means it is most probably a concept and the matched article is highly relevant.

Formally speaking, we apply the following scoring strategy:

- If combined words are matched as in case Q_2, the document is scored according to the number of matching terms.
- If the word is matched in title, the document is assigned a higher score than if it is matched in the page text alone.

The similarity between each basic query (and its alternative queries Q_1 Q_2 see 3.4.1) is calculated based on Dirichlet similarity proposed for information retrieval document scoring taking into account the language model of the whole search repository or index [110]. The Equation given below scores each document as follows :

$$SimilarityScore(d,q) = Boost(w).\frac{\log(1+tf(w,d))}{M.tf(w,D)} + \log(\frac{M}{M+|d|}) \qquad (3.6)$$

Referring to Equation in (3.6), the frequency (tf(w,d)) of query terms is the number occurrences of a query term in the document d. The tf(w,D) is the number of query term w occurrences in the whole index (which is the set of all words in the repository of documents D). M is a smoothing parameter, where boost is a weight for the word in a query. In our approach the boost is a number associated with each query. It is the sum of boost associated with each single key-phrase of a query. A boost associated with each key-phrase is the probability of key-phrase in the whole corpus of generated key-phrase

in all clusters. This number is actually how frequent the term is in the collection of the cluster key-phrases generated. The idea behind this calculation is the intuition that the key-phrase with large corpus frequency is less relevant to one specific cluster and hence should have a less boost value.
We compute the boost associated with each key phrase (w) in cluster (c) in terms of its probability in all clusters (C) as follows:

$$P(w) = \frac{tf(w,c)}{tf(w,C)} \tag{3.7}$$

$$Boost(w) = 1 - P(w) \tag{3.8}$$

The relevant articles are identified as follows:

1. We retrieve up to top 10 scored articles (or less if the number of matched documents are less than 10) then
2. We calculate the average of the scores computed for these articles, i.e. (see Equation 3.9)
3. We select all articles among the top 10 with scores above or equal the computed average score.

$$AverageScore = \frac{\sum(Score(d,q))}{10} \tag{3.9}$$

For example, assume we have the following cluster : *bigdata - business - analysis - intelligence - systems - para - facebook - businessintelligence - bigdata business - mobile*. We start by calculating the boost of each word in this topic, which is in this case the frequency of each word given all key phrases for all clusters extracted (see Table 3.2).

Table 3.2 Boost of key-phrases

Word	Term Probability	Boost
analysis	0.00334	1- (0.00334)
mobile	0.006688	1- (0.006688)
data	0.020066	1 - (0.020066)
..	..	..

Based on these calculations, *analysis* phrase would have higher boost value than the boost associated with the key-phrase *data*. Therefore, for each formulated query,

a Wikipedia article is scored based on key-phrases occurrences, location where these key-phrases appear (text or title of article) and associated boost. For example Wikipedia article *Mobile business intelligence* will have a higher score than the score of the article *Big data* article.

3.5 Constructing the Topic Network

The final phase of our framework is to construct the topic network. The set of Wikipedia articles representing the tweets collection which has been obtained in previous phase of the framework is used to construct this network. The topic are the nodes of this network. The edges between different nodes represent the sub-topic relations between the different topics. In this section we describe our method for constructing this network.

3.5.1 Identifying Nodes: Mapping Miro-blogs to Topics

Wikipedia categories considered as a hierarchical Knowledge Organization System. They provide navigational links and conjunction between all Wikipedia pages in a hierarchy of categories that characterizing knowledge of a domain. So readers can browse and quickly navigate sets of pages on topics that are belong to similar characteristics.

We parse Wikipedia articles –representing the tweets– to extract the direct categories and their sub-categories of each article. The collected, indexed Wikipedia content is only related to Wikipedia articles and their associated information. We do not access or collect any information related to Wikipedia structure or its category system. Since we do not need to parse around 150,000 categories [108] to get sub-categories of small category list, we exploit *DBpedia* to retrieve the direct sub-categories of each category in our list. DBpedia [47] is a World Wide Web(WWW) service that extracts structured content from Wikipedia. The structured content can be used to obtain relationships and properties associated with Wikipedia resources including links to other related data-sets, resource information including the categories of articles, and sub-categories of specific categories. For instance, Figure 3.5 below, shows an entity of DBpedia that represents the social network *Reddit* Wikipedia article at "http://en.wikipedia.org/wiki/Reddit".

The objective of collecting the categories and their sub-categories is to find the best topic label for each key-phrases cluster. Each key-phrase cluster as we have mentioned in Section 3.3, represents a set of tweets that are likely related to one coherent concept. A good topic label should be strongly associated with the key phrases cluster. To find a candidate label for each cluster, we use all tweets associated with one cluster to represent

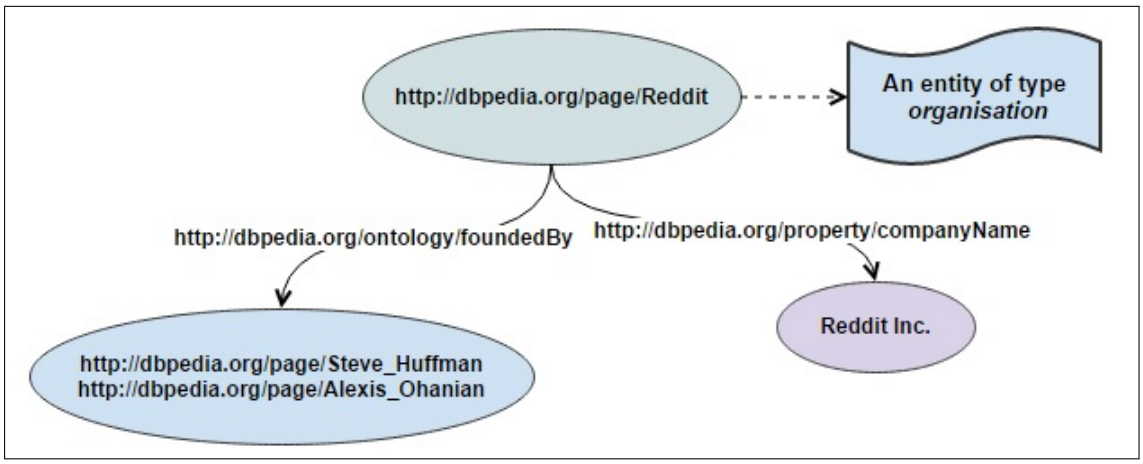

Figure 3.5 Reddit DBpedia Entity

this cluster as a vector of terms (unigram). We denote this vector as (**tweet vector** V_t). We also represent each Wikipedia category and its direct article's titles as another vector (denoted as **category vector** V_c). Each vector represents terms(unigram) and their TF/IDF values (see Equation 3.4). Using Cosine similarity measurement, we calculate the similarity between each tweet vector and category vector of terms (see Equation 3.10).

$$SimilarityScore(V_t, V_c) = \frac{\vec{V}(V_t) \cdot \vec{V}(V_c)}{|\vec{V}(V_t)|.|\vec{V}(V_c)|} \tag{3.10}$$

Finally the proximity between of two vectors is computed as follows:

$$Distance(V_t, V_c) = 1 - SimilarityScore(V_t, V_c) \tag{3.11}$$

Category's title –representing one category vector– with minimum distance to tweet vector is selected as a topic label for the corresponding key-phrase cluster. For instance, if the distance computed for the Wikipedia category *"Business intelligence"* and the tweet vector associated with key-phrases cluster is 41.25, while the distance for another category *"Data analysis"* for the same tweet vector is 45.4, the former category *"Business intelligence"* is used as a topic label for the key-phrases cluster.

3.5.2 Identifying Edges: Building Sub-Topic Relations

Once all key phrases are mapped and labeled by topic concepts and are associated with the corresponding Wikipedia articles and categories the next task is to organize these topics into a network or graph of topics.

A graph is a set of nodes. A node (or vertex) is a topic label in the graph. An edge (or connection) is a link between two verticesNormally our graph is relatively small, with a small number of nodes and edges. We extend the set of categories by including their direct super and sub-categories as we should describe later in this section.

We utilize the Wikipedia category system to construct the adjacency list [3] for the collected Wikipedia categories. To construct the graph we add edges one by one from adjacency list as given by algorithm 1. The intuition behind this algorithm is to guarantee constructing acyclic graph. Once we detect a cycle in the graph we remove the edge that creates the cycle.

Algorithm 1 Graph Construction

Output is A graph represents each group of topics labels(Wikipedia categories)
$L \leftarrow$ List of Topic Labels
$DL \leftarrow$ Adjacency List of Categories
$Vs \leftarrow$ List of current Graph Vertices
for each *cat_id* $\in$L **do**
  $v1 \leftarrow$ CreateVertex(*cat_id*)
  parent_cat_id $\leftarrow$ *DL*(cat_id)
  if there is $v2 \in Vs$ where *parent_cat_id* =v2 **then**
    CreateEdge(v1,v2)
  else
    $v2 \leftarrow$ CreateVertex(*parent_cat_id*)
    CreateEdge(v1,v2)
  End *loop*

Graph Enrichment

One of pitfalls of graph created is that not all topic labels have a parent or sub-category in our adjacency list (as we described above, usually it is a category extracted directly from the scored Wikipedia article for key-phrase cluster and has no sub-category but it has been selected as a topic label). This would generate a sparse graph with few edge. To deal with this problem, we enrich the graph with new nodes and edges that are extracted directly from Wikipedia category system in two different directions (see Algorithm 5):

- Upward: finding a common super category between graph nodes (see Algorithm 2)

[3] In graph theory, an adjacency list data structure is a collection of unordered lists to represent a graph, one for each vertex in the graph. Each list describes the set of neighbors of its vertex.

- Downward: finding a common sub-category between graph nodes (see Algorithm 3) and DBpedia is used for this task.

Algorithm 2 Function GetSuperCategory

Input: Two Vertices Labels ; they are Wikipedia categories.
Output : A List of 3 common super Wikipedia Category between Wikipedia Input Categories with zero or one path distance.
$Cat\_name1 \leftarrow$ first category input
$Cat\_name2 \leftarrow$ first category input
$query \leftarrow$ SPARQL Query
$QueryExecutor \leftarrow$ sparqlService(http://dbpedia.org/sparql) // http://dbpedia.org/sparql is DBpedia EndPoint
$ResultSet \leftarrow$ QueryExecutor($query$).execSelect() // execute SPARQL query.
Return ResultSet

Algorithm 3 Function GetSubCategory

Input: Two Vertices Labels ; they are Wikipedia categories.
Output : A List of 3 common sub-categories Category between Wikipedia Input Categories with zero or one path distance.
$Cat\_name1 \leftarrow$ first category input
$Cat\_name2 \leftarrow$ first category input
$query \leftarrow$ SPARQL Query
$QueryExecutor \leftarrow$ sparqlService(http://dbpedia.org/sparql) // http://dbpedia.org/sparql is DBpedia EndPoint
$ResultSet \leftarrow$ QueryExecutor($query$).execSelect() // execute SPARQL query.
Return ResultSet

Algorithm 4 Function ComputeDistance

Input: Two sequence of characters.
Output : A value of distance between two inputs.
$str1 \leftarrow$ first String input
$str2 \leftarrow$ second String input
$str1\_vector1 \leftarrow$ createVector($str1$) // see Equation (3.4) such that docs is the collection topical terms from and $t \in$str1 terms
$str2\_vector2 \leftarrow$ createVector($str2$) // see Equation (3.4) such that docs is the collection topical terms from and $t \in$str2 terms
$distance \leftarrow$ COSINE($str1\_vector1$,$str2\_vector2$)

In our approach, we execute the proper *SPARQL* [77] queries that check for each node's relation in DBpedia data-sets. The found proper category is then inserted in the

graph as a new node. More precisely, we start by exploring the super and sub-category relations between graph nodes by investigating the relation between nodes that belong to different clusters which have minimum distance between their key-phrases.We use a distance measurement (see Equation (3.5)) to decide which key-phrases cluster we can choose. We choose the cluster that its key-phrases have a minimum distance with unconnected vertex's key-phrases cluster (see distance computing in Algorithm 4). Once we identify the key-phrases clusters to which we need to connect the node to, we fire the required SPARQL queries.

Algorithm 5 Graph Enrichment

Input: A Graph of topics labels (Wikipedia categories) that has a number of unconnected vertices.
Output : A connected Graph of topics labels (Wikipedia categories)
$Vs \leftarrow$ List of current Graph Vertices
$v1 \in$Vs and $v1$ has no edge
for each $topic_id \in$LDATopics **do**
  $distance \leftarrow$ ComputeDistance(TopicTerms($topic\_id$),TopicTerms($v1$))
  $Array$(topic_id) =distance // Array of distance computed index= topic id , value = distance computed
  End *loop*
$L \leftarrow$ GetTopicLabels($topic\_id$) such that $distance \leftarrow Array$(topic_id) is minimum
for each $cat_id \in$L **do**
  $Super\_List \leftarrow$ GetSuperCategory(cat_id,$v1$) //or getSubCategory
  if there is $v2 \in Vs$ where $V2 \in$Super_List **then**
    CreateEdge(v1,v2)
    Exit *loop*.
  else
    for each $id \in$Super_List **do**
      $distance \leftarrow$ ComputeDistance(GetName($parent\_cat\_id$),$GetName$(v1))
      $Array$(id) =distance // Array of distance computed index= category id , value = distance computed
      End *loop*
    $parent\_cat\_id \leftarrow id$ such that $distance \leftarrow Array$(id) is minimum and $id \in$Super_List
    $v2 \leftarrow$ CreateVertex($parent\_cat\_id$)
    CreateEdge(v1,v2)
  Exit *loop*.

If the graph still has unconnected vertices, we simply discard them from the graph. The intuition behind this is that we assume such nodes are outliers categories as Wikipedia often exhibits unsystematic diversity and does not enforce terminological standards.

3.5.3 Summery

In this chapter, we present our model of mapping microblogs into network of topics. We describe the framework components, their input, methods and expected outcome.

4 Implementation

In this chapter we describe the applied part of the thesis. We introduce the development environment used and a detail of implementations done for each components in addition to the APIs, tools and techniques.

4.1 Development Environment

We use a server that has two Intel Xeon E5-2650 2.294 GHz processors ,8 core, 64 GB of physical memory, 4 TB of Hard Desk storage and HP Ethernet 1 Gbps cards. vCenter server software [1] is installed on the server that runs ESXi host. By using vCenter facility, we create three separated virtual machine instances connected to each other through a 10 Mbps Ethernet link. Each machine contains a hyper-threaded Intel(R) Xeon(TM) i7 3770 3.4 GHz processor, 8 GB of RAM, 1 TB of Hard Desk space and Intel(R) 1 Gbps ethernet card. They are all running version 12.04.4 LTS of the Linux 64-bit Ubuntu Server. To access the server remotely , we use VSphere client [2] setup in Windows 7 version 5.5. See Figure 4.1.

Hadoop system [30] is the used processing infrastructure in our framework . Hadoop provides a distributed filesystem and a framework for the analysis and transformation of very large data sets using the MapReduce [25] paradigm. Through Hadoop we can build a scalable, extendable, fast, and cheap framework. With Hadoop a machine has one or more roles that categorize it into Master nodes or Slave nodes. The Master nodes oversee the two key functional pieces that make up Hadoop: storing lots of data (HDFS), and running parallel computations on all that data (Map Reduce). They are: The Name Node

[1] vCenter: Recent VMware cloud technology tools for managing the creation and delivery of virtual machines on vsphere server

[2] VSphere: is a Windows program that used to configure the host and to operate its virtual machines

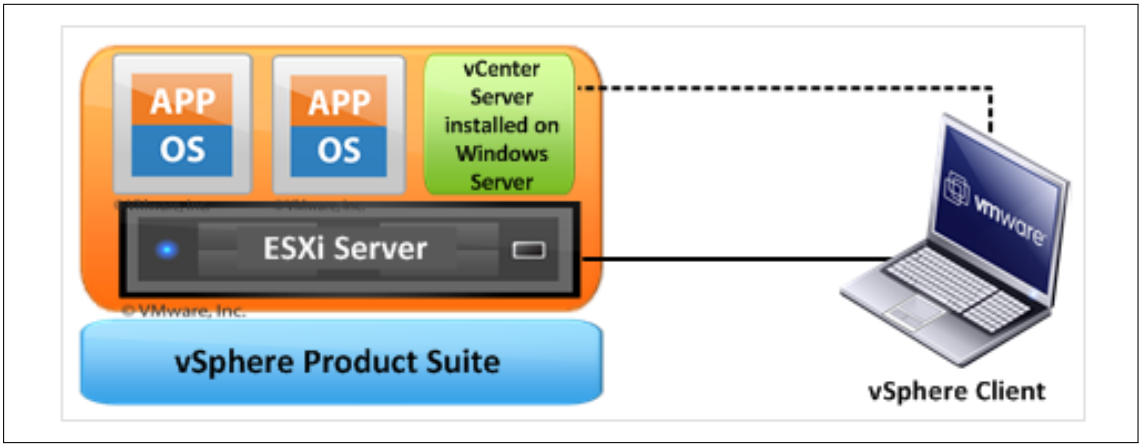

Figure 4.1 Server Machine [6]

oversees and coordinates the data storage function (HDFS), and the Job Tracker oversees and coordinates the parallel processing of data using Map Reduce. Slave Nodes make up the vast majority of machines and do all the dirty work of storing the data and running the computations. Each slave runs both a Data Node and Task Tracker daemon that communicate with and receive instructions from their master nodes. The Task Tracker daemon is a slave to the Job Tracker, the Data Node daemon a slave to the Name Node. We configure Hadoop in our system's nodes where one node act as Name Node and Job Tracker and two others act as Task Tracker. The Three nodes act as Data Node of the hadoop replication data.

4.2 MAMINT Framework

4.2.1 Acquiring And Preprocessing Tweets

Apache Flume project [4] used for the purpose of collecting and aggregating data into Hadoop file system. It starts collecting data based on events from any required data source. We configure Flume to be a channel between Twitter Streaming API and Hadoop file system (HDFS). Twitter streaming API is one of released set of APIs provided by Twitter for developers to give them low latency access to Twitter's global stream of Tweet data. The issuing of tweets by the Streaming API is the event that triggered Flume to handel and deliver the tweets into HDFS. We develop a java program using Twitter4J [1], a Java library for Twitter Streaming API, by which the Flume can select what kind of data he should deliver once a new tweet issued. At this point the tweet collector is ready to start its task. The different filters mentioned at Section 3.1 are applied sequentially on our

tweets collection. The most common technique to replace and filter strings is a regular expression [94]. It provides a way to filter out errors and overuse, and more generally it provides a decent way to recognize substrings in strings. Regular expressions are a very powerful string matching technique which is used in many "search and replace" functions of applications. They are written in a formal language, which is compact but have the power to match with almost all string patterns. All our pre-processing techniques relay on regular expressions, which is a powerful string replacement algorithm.

For stop words we calculate Term Frequency for each single term in the collection and discard all terms that has irregular frequency (more than 90% of corpus words). For each cleaned tweet, a new sequence file is generated. The sequence file is a format from the Hadoop library that encodes a series of key-value pairs. In our case, the sequence file is given a key extracted from its JSON object; the value is the tweet's status preprocessed content.

4.2.2 Acquiring And Processing Wikipedia

The hashtags – data science, cloud computing, big data – used for the tweet collecting are the same source for Wikipedia collecting but as categories [3]. *big data* hashtag has a direct category in Wikipedia, the same thing for *cloud computing* category, but data science hashtag has a Wikipedia article with same name that tagged *Information science* Wikipedia category. So we use *big_data*, cloud_computing and Information_science categories to export all of its pages, sub-categories and their pages, parent categories and their pages. We input this list of chosen Wikipedia categories for Wikipedia export API, and export the XML dump. We use wikixmlj project [105] to parse the exported dump content to extract each article in addition to extracting the the Wik Text and Page title from each article separately.

In our research we use Lucene project to tokonize, parse, and interpret Wikipedia pages into vectors using support vector model [31]. The indexing process is carried on all article pages of several categories under *cloud_computing* and *big_data* and *Information_science* categories, which collected by the aid of Wikipedia's native export API [106]. We exclude all non-article pages from the indexing process. Pages of type disambiguation, redirect and special pages are discarded from this process. Commonly, these kind of pages convey no meaning or concept description. Title and text of chosen pages are extracted and formalize the word vector of each page. This transformation result

[3] We pick these hashtags from our own choice and we assume they are be included is tweets that discussed topics related to cloud computing and data science and big data

in indexed corpora likely composes the concepts knowledge of the extracted key-phrases.

4.2.3 Extracting Features From Tweets Collection

The filtered data-set after pre-processing phase is used to extract features which is subsequently used for generating topic model data-set. The feature extraction and topic modeling is implemented using Mahout machine learning framework [32].
Several experiments are done to compare combination of different features affection on topic modeling accuracy, so different feature vectors with different dictionary word size have been fed for topic model algorithm at each turn (see Section 5.1.1 for the experiments).
We use Mahout API to convert the tweets in Sequence File format to vectors using both TF and TF-IDF weighting for both unigram and bigram combinations. Starting from a directory of sequence files, with each file containing a cleaned tweet, we use API specific classes to convert the content in each sequence file to TF-TFIDF vectors .

4.2.4 Identifying Key-Phrases Clusters

Topic models are algorithms for discovering themes that pervade large and otherwise unstructured collection of document. To identify these topic model (key-phrase clusters), we used Collapsed Variational Bayes [92] which is implemented in Mahout framework [32]. It depends in two assumptions :1) topics tend to place high probability on words that represent concepts or subjects and 2) documents are represented as expressions of those concepts. We used a Map-Reduce instance over our collection using the three processing nodes. The method starts with an empty topic model, reads all the tweets in a mapper phase in parallel, and calculates the probability of each topic for each word in the collection. Once this is done, the counts of these probabilities are sent to the reducer where they're summed, and the whole model is normalized. This process is run repeatedly until the model starts explaining the documents better (when the sum of the (log) probabilities stops changing).

4.2.5 Mapping Clusters to Wikipedia

Lucene project [31] employs Vector Space Model (VSM) [74] and Boolean model for searching and scoring indexed corpora. Briefly, the idea behind the VSM is the more times a query term appears in a document relative to the number of times the term appears in all the documents in the collection, the more relevant that document is to the query.

Based on query specification we introduced in Section 3.4.1, we use Boolean model to narrow down the documents that need to be scored based on document field matching (Text or Title), matching a contiguous phrase, and/or matching several different terms. An example for lucnce specific query structure is given in Figure 4.2.

Q_1 = boolean clause (text:bigdata text:big text:data text:business text:analysis text:intelligence text:systems text:para text:facebook text:businessintelligence text:panel text:mobile text:measure text:negocio)~2^6.0

Q_2 = boolean clause (text:bigdata text:big text:data title:data text:business title:business text:bus title:bus text:analysis title:analysis text:intelligence title:intelligence text:systems title:systems text:system title:system text:future title:future text:para title:para text:facebook title:facebook text:businessintelligence title:businessintelligence text:panel title:panel text:mobile title:mobile text:measure title:measure text:negocio title:negocio)~3^9.0

Figure 4.2 Lucene-based Query Example for MAMINT

4.2.6 Constructing the Topic Network

In this section we describe the implementation of constructing the topical network component.

Identifying Nodes: Mapping Miro-blogs to Topics

For each high scored Wikipedia article we use the categories tagged in and the sub-categories linked to by issuing the corresponding SPARQL query. SPARQL [77](SQL-like query language to query RDF databases) queries are used to access the structured information of DBpedia via WWW end-points. For example, Suppose that we would like to retrieve 10 number of entities of type *company* that founded by *Alexis Ohanian* entrepreneur. The SPARQL query would be as shown in Figure 4.3.

We extract the Wikipedia categories tagged in the Wikipedia articles to which the key phrases clusters were mapped The categories are expanded by their sub-categories using DBpedia structured information.

To select the category that best represents a key phrases cluster, we compute the similarity distance (see Equation (3.10)) between each category (parent and its associated sub-categories) and the corresponding cluster key-phrases.

```
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>

PREFIX dbpprop: <http://dbpedia.org/property/>

PREFIX dbres: <http://dbpedia.org/resource/>

select * where {

 ?company dbpedia-owl:foundedBy ?Alexis_Ohanian.

}

limit 10
```

Figure 4.3 SPARQL Query Example

Using Jena API [5], we adopt a set of SPARQL queries to fetch a list of sub-categories to each category extracted from Wikipedia article to constitute the final topic label candidates. We store the categories and their sub-categories in a kind of adjacency list to devise for graph/network construction. We keep a master list of all categories and then each category in the list mapped to its parent category that it is directly connected to. We use TF-IDF distance measurement implemented in LingPipe API [3] to find the closest label to each topic terms. Each Wikipedia category of our list is measured against key-phrases of each topic. Category with minimum distance chosen as a label for the topic and subsequently to the associated tweets.

Identifying Edges: Building Sub-Topic Relations

We use jgrapht API [2] for nodes creation from adjutancy list of categories and their parent categories. We develop undirected graph that depends on algorithms described earlier at Section 3.5. Based on our algorithm explained (Algorithm 1) in Section 3.5.2, each category selected to represent a key phrase cluster represents a node in the graph. Each of these node is connected with an edge to its direct parent category. See Figure 4.4 for an example. In Figure Figure 4.4 some Wikipedia categories inserted into a graph for specific key phrases clusters. Some of these categories are not connected to any category

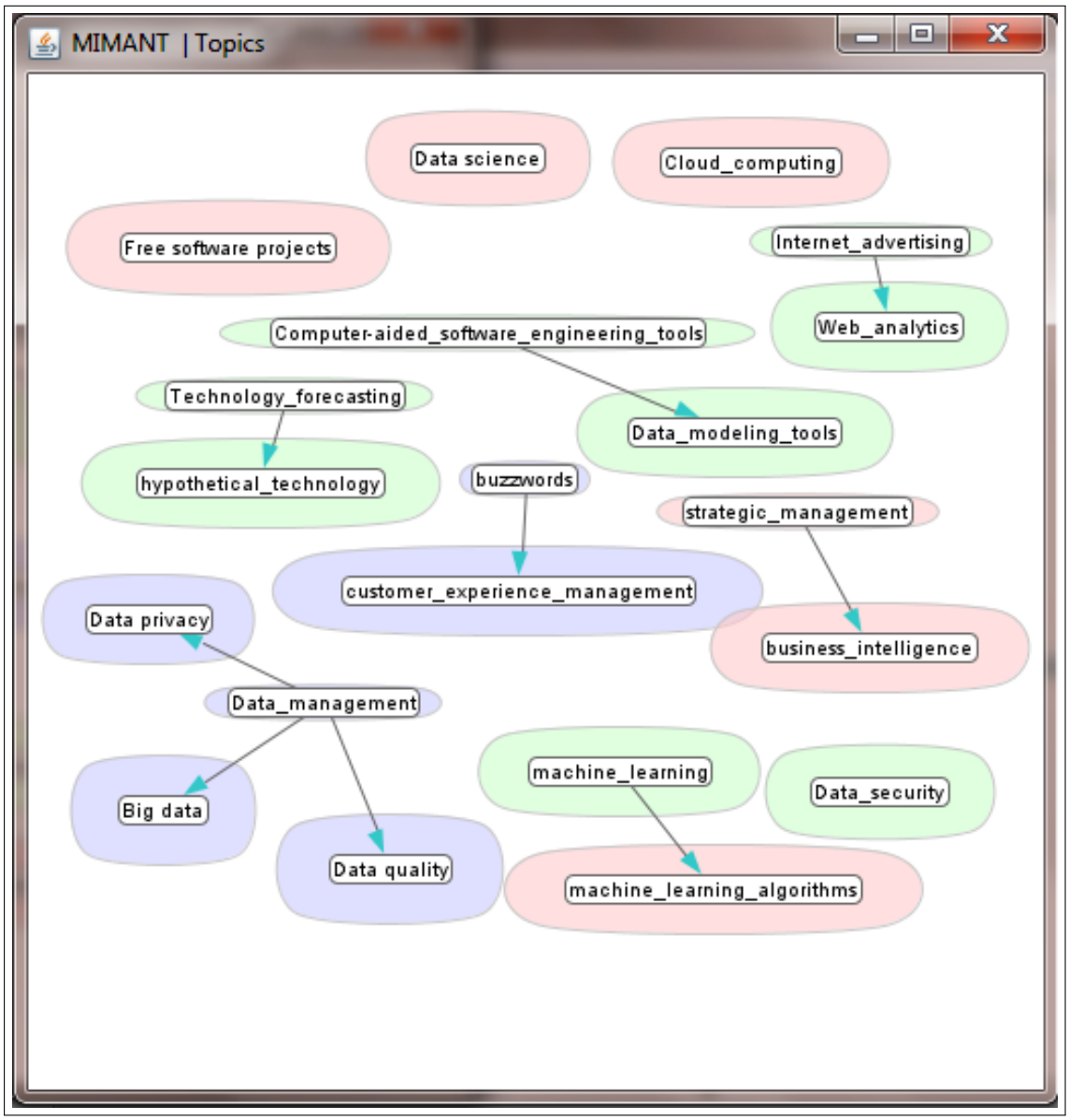

Figure 4.4 Graph of Topics' Labels - Wikipedia Categories

in the graph like: *Cloud computing*, *Data security*, *Free software projects*. Our approach to form a fully connected graph described in Section 3.5.2 (see Algorithm 5). An example for such a graph is given in Figure 4.5.

Data security category has been successfully connected to *Big data* category after finding *Information retrieval* category which is a parent category for both categories in no more than one path a way of them. The same thing happened to *Cloud computing* category; it is connected to *Customer experience management* after finding *Service industries* as a parent category for both. Notice this condition did not satisfied for *Free*

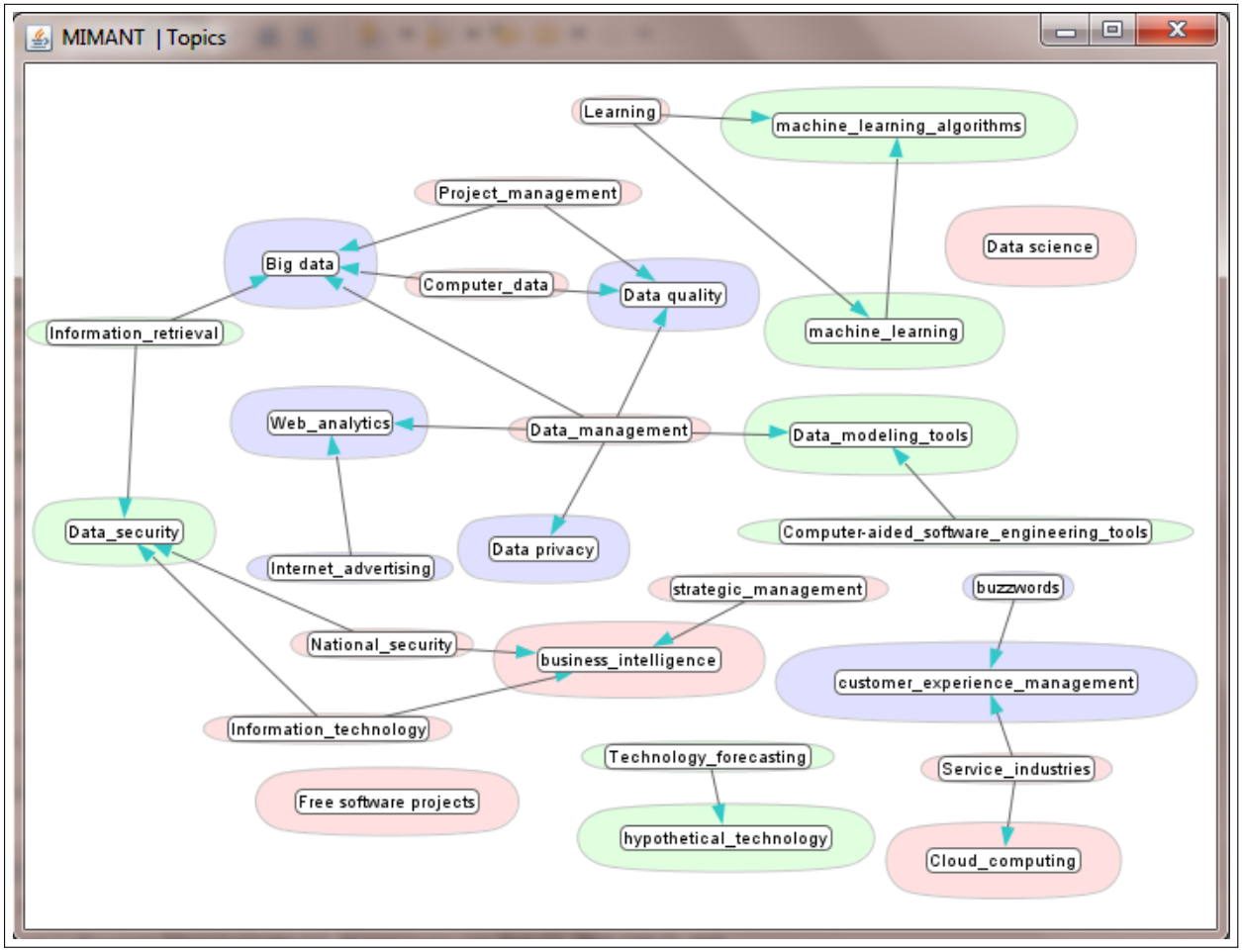

Figure 4.5 Graph of Wikipedia Categories - Solve Unconnected Nodes

software projects so it remains unconnected (see Figure 4.5).

4.3 Demonstration

We use Prefuse [75], the Java-based visualization toolkit to visualize the graph generated by our system. A snapshot of how MAMINT prototype looks like is shown in Figure 4.6.

4.3.1 MAMINT Basic Usage

- User enters hash-tags for topics he/she is interested in (see Figure 4.7).
- MAMINT collects the set of tweets associated with the given hash-tags.
- MAMINT prototype shows a graph/network of topics distilled from the tweets (see

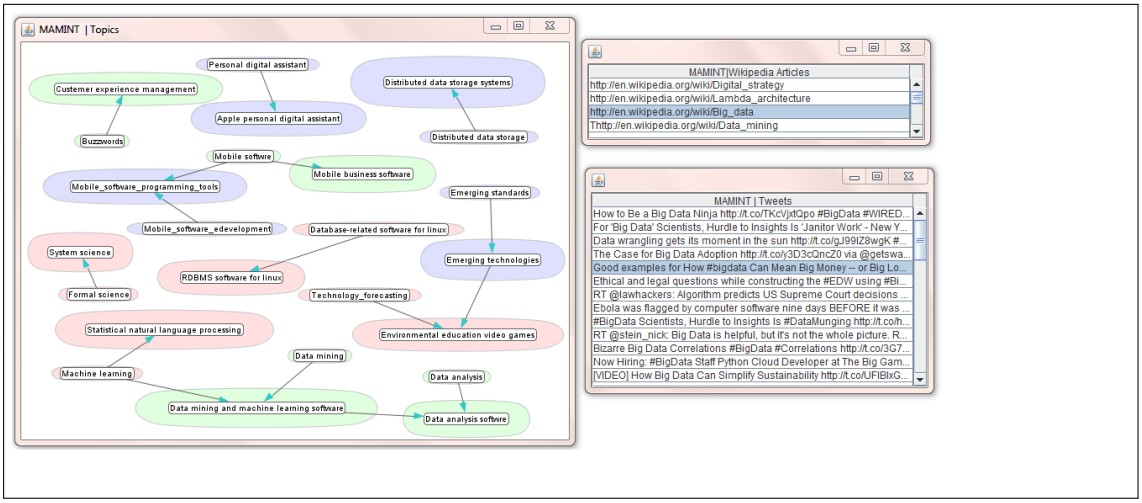

Figure 4.6 A snapshot of MAMINT system

Figure 4.7 MAMINT Hash-tag Insertion Window

Figure 4.8).

- User then can click on any node of the network node. Each node represents topic/-subtopic. A “Wikipedia Articles” window appears and shows a set of Wikipedia articles related to the node’s topic (see Figure 4.9). Also the "Tweets" window appears and shows a list of tweet collection related to the clicked node (see Figure 4.10).

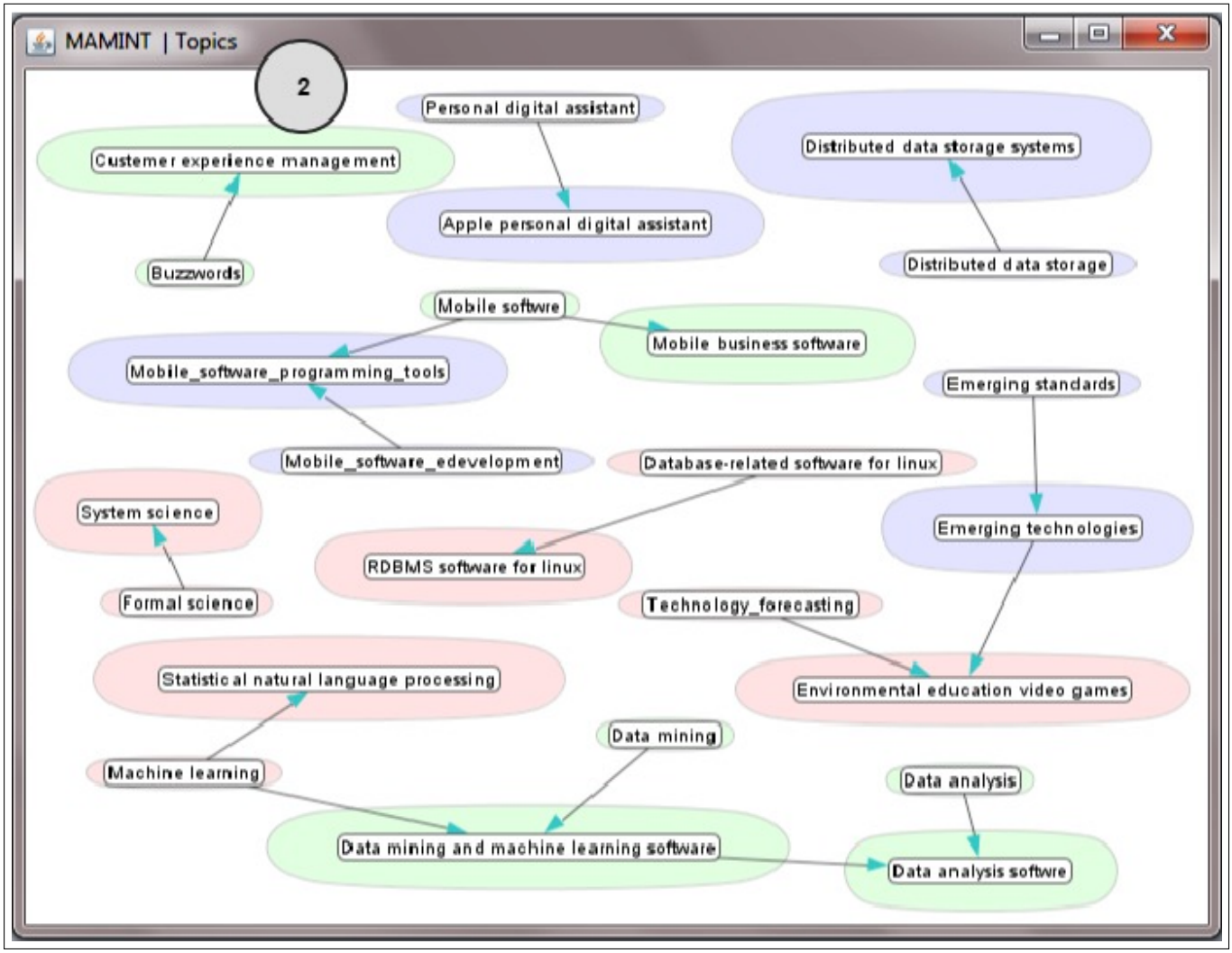

Figure 4.8 MAMINT Topic Network Window

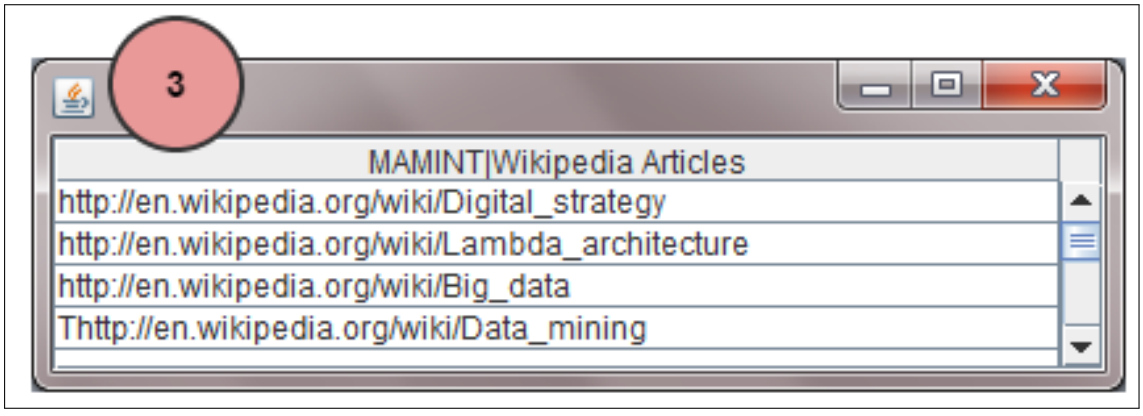

Figure 4.9 MAMINT Wikipedia Articles Window

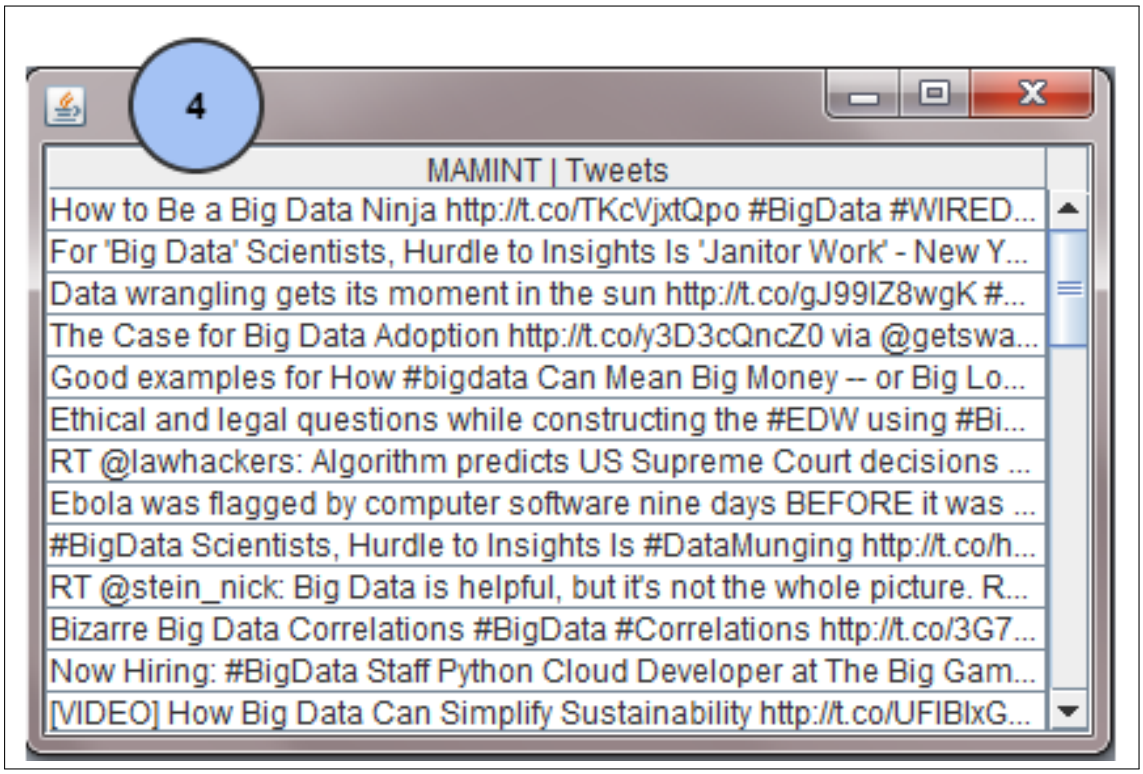

Figure 4.10 MAMINT Tweets Window

5 Evaluation

We want to evaluate our implemented approach in mapping micro-blogs into network of topics. We evaluate different components of our framework mainly the components presented in Section 3.3, Section 3.4 and Section 3.5. In the following sections, we describe each component input/output, experiment settings, results, and evaluation metrics. Furthermore the complete framework evaluated by study the End to End framework relatedness.

5.1 Identifying Key-Phrases Clusters

Input: The Testing Data-set (TD) experiment is a sum of **3000** tweets. These are sampled from a total of 30,000 tweets preprocessed data-set.
Output: A set of key-phrases group, We choose to cluster tweet featured terms into 20 group. Each group contains the top high 10 terms that supposed to be related to each others.

5.1.1 Experiments Settings

As we have mentioned on approach Section 3.4, we used LDA for cluster generation. LDA requires a number of parameters: the weight of the words to be considered as feature (Whether TF or TFIDF), the minimum frequency of the term in the entire collection to be considered as part of the dictionary file (support), the nature of word (n-gram where n can be 1 or 2), and or the number of different documents that word mentioned in. Therefore the feature space size and the generated phrases cluster we obtained for each experiment were different (see Table 5.1.1). In order to identify these parameters, we conducted a number of different experiments with different parameters during development phase. In

the experiments 1-7 conducted during development phase, we used a data "Development Data-sets" (DD) of **3000** tweets are sampled from a total of 30,00 tweets preprocessed data-set different from those sampled for testing data-set. In what follows, we describe the experiments we conducted during development and testing phases (see Table 5.1.1). The generated cluster for each experiment can be seen on Appendix Section A.1

Development Phase Experiments:

Experiment 1: In this experiment, we calculated TF-IDF of tweet words to form a dictionary of 2633 distinct featured terms. The term should have at least two occurrences in the collection to be considered as a feature. we did not restrict the document frequency, the word occurrences can be in one tweet (document). Additionally, in this task we considered each word is single token a word (n-gram where n=1). The topical terms for 20 different clusters are shown in Table A.1.

Experiment 2: In this experiment, we weighted featured terms based on TF-IDF measurement to form a word dictionary of 5899 distinct words. However, for the purpose of exploring other hidden relations between words in tweets we considered a word to be n-gram (n = 1 and 2) . subsequently , The dictionary is increased over the first experiment. The key-phrases of 20 cluster resulted from this experiment are shown in Table A.2.

Experiment 3: In this experiment, we weight featured terms based on TF measurement. The word that has a frequency of at least two occurrences will be considered as a featured term. That is the case of all single token which collectively forms a dictionary size of 2633. The resulted topical terms are shown in Table A.3.

Experiment 4: At this experiment we should investigated the case of weighing TF with considering a word as a 2-gram. We obtained a dictionary size of 5899 words and the key-phrases are shown in experiment Table A.4.

Experiment 5: Referring to pervious experiments, there is no major difference between them in term of topical terms extracted. Thus, we try to restrict the dictionary terms by rise the number of occurrences of each term inclemently by 3 then to 4 the dictionary size did not change it has the same as considering only 2 occurrences. By rising the term frequency to at least 5 occurrences for the featured terms, the dictionary size is dropped to 1238 words (up to 2 tokens). We try LDA over the produced dictionary to extract topical terms for 20 topics as shown in Table A.5.

Experiment 6: In this experiment we investigate the topical terms in case the minimum term frequency is increased to 6 occurrences which leads to a dictionary of 940 words even the words are compound of up to 2 single tokens (see Table A.6).

Experiment 7: Form experiment 6 we can notice that the resulted topical terms are more generic and many contextual terms have been excluded , so at this experiment we set the minimum term frequency to 5 to each single or compound tokens (up to 2 tokens). Moreover, for the purpose of more filtering of dictionary words to include only sense terms, we set the min number of documents(tweets) the word needs to be in to 5 distinct tweets (Exclude all words that not mentioned in at least 5 different tweets). Hopefully restricting the document frequency would lead to excluding spam words appeared in the dictionary which has a large frequency in the collection. Accordingly the dictionary size decreases to 1180 distinct. Table A.7 on Section A.1.1 shows the resulted topical terms.

Experiment #	Measurement	Support	n-gram	document frequency	Dictionary Size	Data-Set
Experiment 1	TF-IDF	2	1	1 tweet	2633 different word	DD
Experiment 2	TF-IDF	2	up to 2 tokens	1 tweet	5899 different word	DD
Experiment 3	TF	2	1	1 tweet	2633 different word	DD
Experiment 4	TF	2	up to 2 tokens	1 tweet	5899 different word	DD
Experiment 5	TF	5	up to 2 tokens	1 tweet	1238 different word	DD
Experiment 6	TF	6	up to 2 tokens	1 tweet	940 different word	DD
Experiment 7	TF	5	up to 2 tokens	5 distinct tweet	1180 different word	DD
Experiment 8	TF	5	up to 2 tokens	5 distinct tweet	1446 different word	TD

Table 5.1 Experiments Settings - Topic Detection

Testing Phase Experiment:

The resulted outcomes of development phase's experiment 1-6 as shown in Section A.1.1 of Appendix showed that the key-phrases in each generated cluster are too generic and many contextual terms have been excluded while those generated for experiment 7 are more specific and formative (see Table A.7). Therefore, we applied the same settings of experiment 7 on experiment 8 which was conducted on testing data-set. These parameters are restricted to those featured terms that occur at least 5 times in the corpus, mentioned in distinct 5 tweets, and are single or compound words (n-gram where n= 1 and n=2).
Experiment 8: The input dictionary size is 1446 words resulted from this experiment (see Table 5.1.1). The full list of key-phrases extracted is shown in Table 5.1.1.

5.1.2 Results

Since we do not have a gold-standard which we can refer to for evaluating the quality of the generated clusters in experiment 8, we used two human judges [1]. Human judges

[1] They are web developers with Information Technology major

	Terms
Topic 0	bigdata - enterprise - hadoop - spark - development - applications ...
Topic 1	bigdata - smart - para - sense - virtual - turning ...
Topic 2	bigdata - predictive analytics - ediscovery - health - medicine ...
Topic 3	bigdata - startup - infographic - experts - authentiweb ...
Topic 4	bigdata - internet things - internetofthings - hyped - energy ...
Topic 5	marketing - bigdata - marketing - database - sales ...
Topic 6	bigdata - janitor - hurdle - data scientists - sensors - ..
Topic 7	bigdata - research - machinelearning - datamining - massive ..
Topic 8	bigdata - social - insurance - mistaken - predict - network ..
Topic 9	bigdata - strategy - customer - experience - management - healthcare ..
Topic 10	bigdata- data scientist - chief - data officer ..
Topic 11	bigdata - system - time - data - sports - botmaker ..
Topic 12	bigdata - privacy - critical - ehealth - opendata ..
Topic 13	nosql - tools - mongodb - engineer - nosqlnow - database - oracle - developer
Topic 14	bigdata -mdatadriven - education - analysis - classroom students..
Topic 15	bigdata - business intelligence - market - chain - supply - buzzwords ..
Topic 16	bigdata - government - marketing - data science - analytics - data mining
Topic 17	bigdata - retail - gartner- hype cycle - turn ..
Topic 18	bigdata - cloud - security - mobile - social - daily ..
Topic 19	bigdata analytics - realtime - forecasting - healthcare - banking ..

Table 5.2 Topical Terms of Experiment 8

evaluated whether the key-phrases in one cluster form a coherent set. A coherent set reflects a meaningful, interpretable key-phrases in some context, and related such that one can logically label them in a subject-heading like.

We used Fleiss' kappa [29] to measure the agreement between the two judges. Fleiss' kappa is a well known statistical measure that calculate the degree of agreement in

classifying items and instances. The kappa k, can be defined as:

$$k = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e} \tag{5.1}$$

$\bar{P}$ is defined in Equation (5.2) yet P_i calculates the extent to which raters agree, while $\bar{P}_e$ is defined as in Equation (5.3) and P_j is used to measure the proportion of all assignments.

$$\bar{P} = \frac{1}{n} \sum_{i=1}^{N} P_i \tag{5.2}$$

$$\bar{P}_e = \sum_{j=1}^{k} P_j^2 \tag{5.3}$$

Based on k Coefficient value, the agreement is assessed (see Table 5.1.2).

k	Interpretation
< 0	Poor agreement
0.01 – 0.20	Slight agreement
0.21 – 0.40	Fair agreement
0.41 – 0.60	Moderate agreement
0.61 – 0.80	Substantial agreement
0.81 – 1.00	Almost perfect agreement

Table 5.3 k Coefficient Interpretation

The result of inter-annotators agreement in Table 5.4, where the complete evaluation Table is on Section A.1.1 of Appendix shown in Table A.1.1.

	Useful	Not Useful
Total	34	6
P_j	0.85	0.15
Kappa Coefficient	0.2156	

Table 5.4 The inter-annotator agreements using kappa- coefficient of experiment 8 evaluations

From the Table above we can notice that 85% of the key-phrases were assigned to the right topic. The inter-annotator agreement is 0.2156 is a fair agreement according Fleiss' kappa interpretation.

5.2 Mapping Clusters to Wikipedia

Input: The input of the experiment in the testing phase is a set of key-phrases clusters generated and evaluated from experiment 8 (Key-phrases are shown in Table 5.1.1). As we need to map key-phrases to Wikipedia articles, we first collected a set of articles (see Table5.5).
Output: This component generates a set of relevant Wikipedia articles to each cluster. As described on (Section 4.2.5).

Number of Wikipedia Articles	Around 1268 Page
Size of Wikipedia Dump File	10.2 MB
Size of Wikipedia Index Repository	5 MB Indexed Terms

Table 5.5 Wikipedia Articles Data Set

5.2.1 Experiments Settings

Development Phase Experiments

We studied three different combinations of three parameters : decomposition method,query boosting, and matching model. See Table 5.2.1 describe different settings. The outcomes from setting 1 and 2 are shown in table appendix respectively. We found from the initial evaluation from those two settings that the query Wikipedia article matching is low. We further investigated the effect of using the decomposition and the boost. We decomposed most of key-phrases before structuring and issuing the queries to the indexed Wikipedia repository in experiment 3. In addition, in experiment 3, the scoring method described in Section 3.4.2 was used. We found from this study that settings of experiment 3 yields a better matching articles. Therefore, we applied these settings in our experiment described on the following section.

Settings #	Query Decomposition	Query Boost	Matching Model
Setting 1	No	No	Vector Space Model
Setting 2	Yes	No	Vector Space Model
Setting 3	Yes	yes	Vector Space Model & Binary Model

Table 5.6 Experiments Settings

Testing Phase Experiment

As we have mentioned in pervious section, the settings of experiment 3 were applied on testing data-set. The outcome of this experiment is shown in Table 5.7:

Wikipedia Articles	
Query 0	http://en.wikipedia.org/wiki/Apache_Hadoop http://en.wikipedia.org/wiki/Apache_Spark ...
Query 1	http://en.wikipedia.org/wiki/Oracle_Big_Data_Appliance http://en.wikipedia.org/wiki/Big_data ...
Query 2	http://en.wikipedia.org/wiki/Predictive_analytics http://en.wikipedia.org/wiki/Industry_4.0 ..
Query 3	http://en.wikipedia.org/wiki/Big_structure http://en.wikipedia.org/wiki/Big_data ...
Query 4	http://en.wikipedia.org/wiki/Big_data http://en.wikipedia.org/wiki/Internet_of_Things ..
..	...
..	...
Query 17	http://en.wikipedia.org/wiki/Disaster_recovery_plan ...
Query 18	http://en.wikipedia.org/wiki/Mobile_cloud_computing ...
Query 19	http://en.wikipedia.org/wiki/Health_care_analytics http://en.wikipedia.org/wiki/Predictive_analytics ...

Table 5.7 Scored Wikipedia Articles - Testing Experiment

5.2.2 Results

Relevancy is the main measurement used in Information retrieval domain. It denotes how well a retrieved document or set of documents meets the user's needs. Therefore, we evaluate the relevancy of Wikipedia articles retrieved by this component by measuring the user satisfaction of the scored articles compared to cluster key-phrases. We used two human judges. Human judges evaluated whether the cluster key-phrases is related to the Wikipedia article list. We used Fleiss' kappa to measure the agreement between the two judges (see Table 5.8 for evaluation results where the complete inter-annotator agrement Table is in Section A.1.2 of Appendix shown in Table A.13).

From the Table we can notice that 80% of the key-phrases were mapped to the right Wikipedia articles. But we can notice the inter-annotator agreement is 0.0625 which is slight agreement according Fleiss' kappa interpretation. Therefore, another experiments was conducted in order to measure the accuracy of the mapping process. We introduce this experiments in what follows.

	Related	Not Related
Total	64	16
P_j	0.8	0.2
Kappa Coefficient	0.0625	

Table 5.8 The inter-annotator agreements using kappa-coefficient of article's relevancy

Google Results Evaluation

A good Information Retrieval system is the one that has a high relevancy in term of Precision and Recall values. The relevance of the retrieved documents together or the relatively of them to the user query are measured by a well known information retrieval measurements which are Precision and Recall [17]. *Precision* represents the ability to retrieve top-ranked documents that are mostly relevant. On the other hand, *Recall* represents the ability of the search to find all of the relevant items in the corpus. It is not possible to compute the Recall in our case since, the total number of relevant items is not available. Therefore we will consider only the measurements that are based on *Precision* computations.

Precision can be formally defined as follows:

$$Precision = \frac{\#(retrieved \quad relevant \quad documents)}{\#(retrieved \quad documents)} = P(relevant|retrieved) \tag{5.4}$$

In this experiment, we compute the Precision with the help of Google Search Engine. The 20 queries we evaluate in this experiment were sent to Google– with domain of search restricted to Wikipedia. We consider the top-5 ranked documents list returned by Google, and we assume that the relevant documents *R* are any *two documents* of these five, (*R = 2 in our experiments*.) We also assume that our system retrieves two documents– the top-2 ranked list of documents. For each query, we mark each document in the ranked list that is relevant according to the Google top-5 ranked list. We then compute a precision for each position (positions 1, and 2) in the ranked list that contains a relevant document.

Formally, we calculate $Precision@k$ [58], where k is a threshold that is set to 2 in our case since we assume that two documents are retrieved for each query. $Precision@k$ is computed as follows:

$$Precision@k = \frac{\#(retrieved \quad relevant \quad documents)}{k} \tag{5.5}$$

The *Average Precision (AP)* for each query represents the average of the precision values at the points at which each relevant document is retrieved. That is:

$$AP = \frac{Precision@1 + Precision@2 + \cdots + Precision@k}{R} \tag{5.6}$$

The *Mean Average Precision (MAP)* represents the average of the average precision value for a set of queries. Given that the total number of queries is $|Q|$, the MAP can be calculated as follows:

$$MAP = \frac{\sum_{i=1}^{|Q|} AP_i}{|Q|} \tag{5.7}$$

In our experiment, *MAP was 0.8* which is very satisfactory. Full computations are in Table A.14 at Section A.1.2 of Appendix Chapter).

5.3 Constructing the Topic Network

As we build our network mainly with help of Wikipedia Category structure, we will not focus of evaluating the structure of network constructed by our approach. In this section we focus on evaluating the assignment of Wikipedia articles to their topics rather than evaluating the relations between these topics.

Input: The input to the experiment of this component is the set of Wikipedia articles to which our key-phrases were mapped to in experiment described in pervious section (see Table 5.7).

Output: The output of this experiment is A graph/network of topics. More details on graph structure is given in Section 3.5

5.3.1 Experiments Settings

Development Phase Experiment

In the development phase, we studied the informative of the generated graph by the two methods described in approach section. In the following the graph created in tow different method. Figure 5.1 is created based on common parent category between nodes, Figure 5.2 is a graph created by the usage of sub-category relation.

Testing Phase Experiment

Wikipedia articles are assigned to different topics. Remember that the graph nodes represent topics, and these topics are Wikipedia categories. Some of these topics/categories

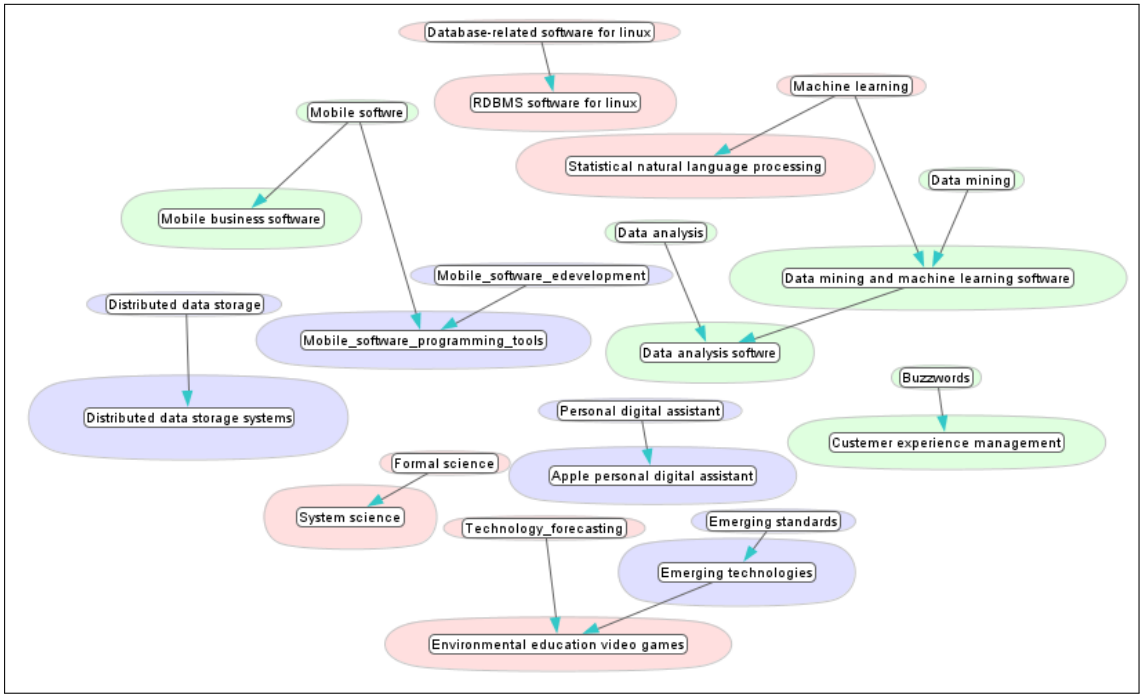

Figure 5.1 Graph using parent relation -method 1

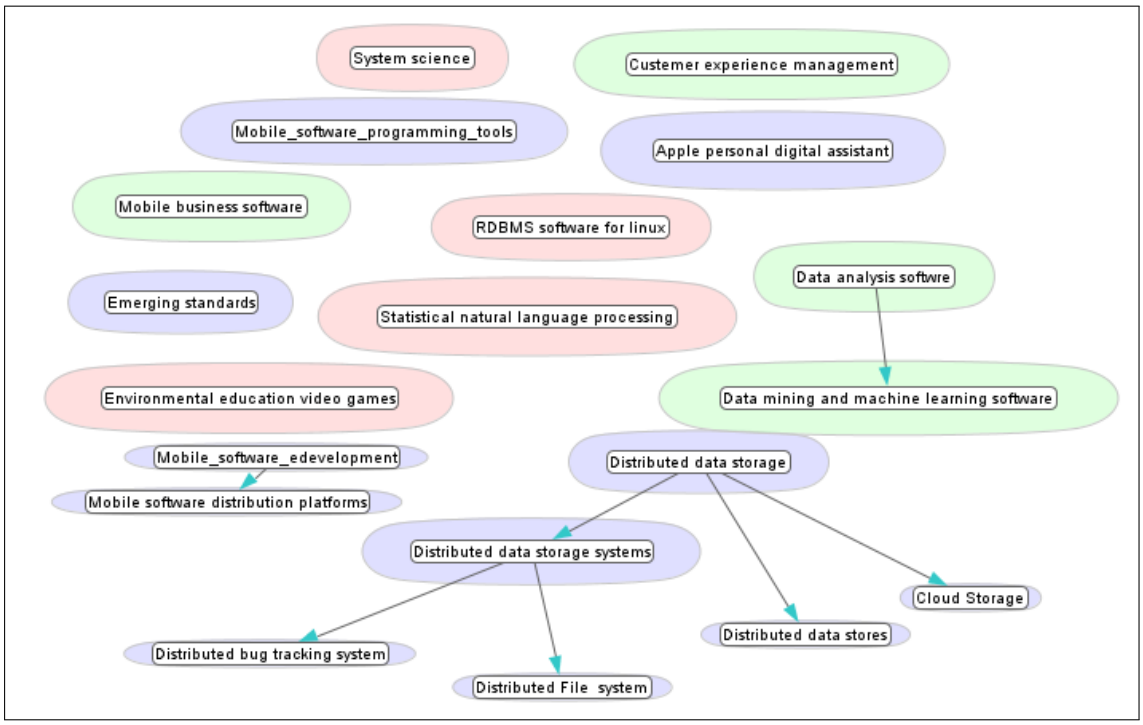

Figure 5.2 Graph using sub-category relation-method 2

are already assigned by Wikipedia to the Wikipedia articles which we need to evaluate. These Topics were excluded from our evaluation. We focus only on evaluating those

Wikipedia articles assigned to topics by our approach but not by Wikipedia.

5.3.2 Results

We evaluate the assignment of Wikipedia articles to their topics shown in Table 5.7. As we do not have a gold-standard we used two human judges. They have been asked to evaluate if the given category *can be* assigned as a topic for the corresponding Wikipedia article. We used Fleiss' kappa to measure the agreement between the two judges. See Table 5.3.2. The complete evaluation is in Appendix on Table A.15.

	Related	Not Related
Total	24	14
P_j	0.63	0.36
Kappa Coefficient	0.337	

Table 5.9 The inter-annotator agreements using kappa-coefficient of article to category relevancy

From the Table we can notice that 63% of Wikipedia articles can be assigned to categories recommended by our approach. These results reflect that we need to work more on finding a more accurate similarity method for this component.

This accuracy rate is quiet reasonable in our tweets collection. We pick the hashtag from our own choice, we did not know actual people engae in learning process using these hashtags. Therefore, most of the tweets are job opportunities and tools and projects promotions. Thus, The diversity of tweets is the main reason of the different key-phrase that lead to different Wikipedia articles.

5.4 End to End Evaluation

We need to evaluate the assignment of tweets – the input of our framework – to topics in the generated topics network – a final output of our framework.
Input: A collection of preprocessed tweets.
Output: The output of this experiment is A graph/network of topics. More details on graph structure is given in Section 3.5.

5.4.1 Experiments Settings

For evaluation experiment, we sampled a set of 100 tweets from the test data-set to evaluate the assignment of tweet to the Wikipedia category chosen as a label for the cluster it belongs to. The Tweet list is shown in Section A.1.4.

5.4.2 Results

We evaluate the assignment of tweets to topics in a sample of 100 tweets. As we do not have a gold-standard we used two human judges. They have been asked to evaluate if the given category *can be* assigned as a topic for the tweet. We used Fleiss' kappa to measure the agreement between the two judges (see Table 5.4.2). For the complete evaluation see Table A.16 and Section A.1.4 for the complete tweet list evaluated. From the Table we

Wikipedia Category	Related	Not Related	Pi
Total	174	26	88
Pj	0.87	0.13	
Kappa Coefficient	0.469		

Table 5.10 Tweet two Wikipedia Categories Evaluation

can notice that 87% of Tweets assigned to ultimate categories chosen by our approach as a label for the cluster. The inter-annotator agrement 0.469 is moderate. Therefore, the results are satisfactory.

5.5 Summery

Component	Evaluation Method	Criterion	Result
Identifying Key-Phrases Clusters	Topic Usefulness	Human Judge	85%
Mapping Clusters to Wikipedia 1	Human Judge	User Satisfaction	80%
Mapping Clusters to Wikipedia 1	Mean Average Precision (MAP)	Relevancy	0.8
Constructing the Topic Network	Human Judge	Article to Category Assignment	63%
End to End Evaluation	Human Judge	Tweet to Category Relatedness	87%

Table 5.11 MAMINT-Experiments Evaluation

6

Conclusions and Future work

In this research, we studied the possibility of mapping microblogs generated in learning activities into **a network of topics** by utilizing methods from Microblogs Analytic and Learning Analytic. This network would provide essential feedback about what topics are being learnt by people involved in this learning process, the size of interest in a particular topic and how well these topics are being learnt.

We developed MAMINT framework that collects and processes tweets in order to extract features and key-phrases, cluster them, map these clusters to Wikipedia articles, and finally construct a network of topics representing what people are discussing. At the applied part of the thesis, we collect of tweets using a set of active hash-tags. We extract the featured words based on different word weight methods. The complete set of words used by topic model method to cluster a set of key-phrases clusters. Each cluster mapped to Wikipedia articles. The mapping depends on using a repository of Wikipedia categories processed, indexed and scored by Vector Space Model approach. The set of scored Wikipedia articles form the source for each key-phrases cluster's label. Each label 'Wikipedia Category' and its sub-category and parent category are forms the nodes on the network of topics. The ultimate topic network is enriched by different methods using Wikipedia category system.

We conducted several experiments to evaluate MAMINT and its different components. The End-to-End evaluation of MAMINT shows that it has an accuracy of 87%. This is a very satisfactory results compared to complexity of this research problem.

At future work we will work on enhancing the accuracy of our system, try to leverage Wikipedia structure and category system for finding labels of each key-phrases clusters and on integrating it with other components for sentiment analysis to complete our envisioned system.

Bibliography

[1] Introductionp twitter4j.

[2] Welcome to JGraphT - a free Java Graph Library jgrapht, 2011.

[3] What is LingPipe? lingpipe, 2011.

[4] Welcome to Apache Flume apache flume, 2012.

[5] Apache Jena apache jena, 2014.

[6] vCenter Server vmware, 2015.

[7] Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. Semantic enrichment of twitter posts for user profile construction on the social web. In *The Semanic Web: Research and Applications*, pages 375–389. Springer, 2011.

[8] Charu Aggarwal and ChengXiang Zhai. A survey of text clustering algorithms. In *Proceedings of Mining Text Data*, pages 77–128. Springer, 2012.

[9] Hughes Amanda and Palen Leysia. Twitter adoption and use in mass convergence and emergency events. In *Proceedings of the 6th International ISCRAM Conference*, volume 6, pages 248–260. Inderscience, 2009.

[10] Aneesha. SNAPP social networks adapting pedagogical practice, 2011.

[11] Kimberly Arnold, Matt Bethune, and Matthew Pistilli. Signals: Using Academic Analytics to Promote Student Success educause review online, 2012.

[12] Joanne Berg, Kathy Christoph, and Lori Berquam. Social networking technologies: A "poke" for campus services. 42:32–44, 2007.

[13] David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006.

[14] Caruso Borreson, Ellison Nicole, Nelson Mark, and Salaway Gail. The ecar study of undergraduate students and information technology. Technical report, 2008.

[15] Davison Brian and Hong Liangjie. Empirical study of topic modeling in twitter. In *Proceedings of the First Workshop on Social Media Analytics*, pages 80–88. ACM, 2010.

[16] Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479, 1992.

[17] Michael K. Buckland and Fredric C. Gey. The relationship between recall and precision. *JASIS*, 45(1):12–19, 1994.

[18] Amparo E Cano, Andrea Varga, Matthew Rowe, Fabio Ciravegna, and Yulan He. Harnessing linked knowledge sources for topic classification in social media. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, pages 41–50. ACM, 2013.

[19] Yan Chen, Hadi Amiri, Zhoujun Li, and Tat-Seng Chua. Emerging topic detection for organizations from microblogs. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 43–52. ACM, 2013.

[20] Shui-Lung Chuang and Lee-Feng Chien. A practical web-based approach to generating topic hierarchy for text segments. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 127–136. ACM, 2004.

[21] Ian Clark. Formative assessment: Assessment is for self-regulated learning. 24:205–249, 2012.

[22] Michael Crow. No More Excuses educause review online, 2012.

[23] Blei David, Ng Andrew, and Jordan Michael. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[24] Nadeau David and Sekine Satoshi. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30:3–26, 2007.

[25] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[26] Twitter Huffington Post eMarketer. Twitter Statistics statistic brain, 2013.

[27] Demir Engin, Demirbas Murat, Ferhatosmanoglu Hakan, Fuhry Dave, and Sriram Bharath. Short text classification in twitter to improve information filtering. In

Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 841–842. ACM, 2010.

[28] Miao Fan, Qiang Zhou, and Thomas Fang Zheng. Mining the personal interests of microbloggers via exploiting wikipedia knowledge. In *Computational Linguistics and Intelligent Text Processing*, pages 188–200. Springer, 2014.

[29] Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.

[30] The Apache Software Foundation. Welcome to Apache Hadoop hadoop, 2012.

[31] The Apache Software Foundation. Welcome to Apache Lucenelucene, 2014.

[32] The Apache Software Foundation. What is Apache Mahout? apache mahout, 2014.

[33] Evgeniy Gabrilovich and Shaul Markovitch. Wikipedia-based semantic interpretation for natural language processing. *Journal of Artificial Intelligence Research*, 34(2):443, 2009.

[34] Yan Gao, Jin Liu, and PeiXun Ma. The hot keyphrase extraction based on tf* pdf. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on*, pages 1524–1528. IEEE, 2011.

[35] Yegin Genc, Winter Mason, and Jeffrey V Nickerson. Semantic transforms using collaborative knowledge bases. *Howe School Research Paper*, (2013-23), 2013.

[36] Yegin Genc, Winter A Mason, and Jeffrey V Nickerson. Classifying short messages using collaborative knowledge bases: Reading wikipedia to understand twitter. In *# MSM*, pages 50–53, 2013.

[37] Yegin Genc, Yasuaki Sakamoto, and Jeffrey V Nickerson. Discovering context: classifying tweets through a semantic transform based on wikipedia. In *Foundations of Augmented Cognition. Directing the Future of Adaptive Systems*, pages 484–492. Springer, 2011.

[38] Yuhang Guo, Bing Qin, Ting Liu, and Sheng Li. Microblog entity linking by leveraging extra posts. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 863–868, 2013.

[39] Al-Khalifa Hend. Finding a place for twitter in higher education. 2010, 2010.

[40] Hsiu-Ting Hung and Steve Chi-Yin Yuen. Educational use of social networking technology in higher education. 15:703–714, 2010.

[41] IBM. Analytics for Education ibm, 2013.

[42] Blackboard Inc. About Blackboard Analytics blackboard analytics.

[43] Rapid Insight Inc. Rapid Insight Veera rapid insight, 2013.

[44] SAS Institute Inc. SAS for Higher Education sas.

[45] Twitter Inc. FAQs about Trends on Twitter help center, 2013.

[46] Instructure. Hundreds of Features .All Working Together canvas, 2012.

[47] Ted Thibodeau Jr. The DBpedia Knowledge Base dbpedia, 2014.

[48] Reynol Junco, Greg Heiberger, and Eric Loken. The effect of twitter on college student engagement and grades. *Journal of Computer Assisted Learning*, 27:119–132, 2010.

[49] SB Kotsiantis, D Kanellopoulos, and PE Pintelas. Data preprocessing for supervised leaning. *International Journal of Computer Science*, 1(2):111–117, 2006.

[50] Akshi Kumar and Teeja Sebastian. Sentiment analysis on twitter. *International Journal of Computer Science Issues (IJCSI)*, 9, 2012.

[51] Darrell Laham, Peter Foltz, and Thomas Landauer. An introduction to latent semantic analysis. *Discourse Processes*, 25:259–284, 1998.

[52] Debra Laverie, Shannon Rinaldo, and Suzanne Tapp. Learning by tweeting: Using twitter as a pedagogical tool. 33:193–203, 2011.

[53] Zhenhui Li, Ding Zhou, Yun-Fang Juan, and Jiawei Han. Keyword extraction for social snippets. In *Proceedings of the 19th international conference on World wide web*, pages 1143–1144. ACM, 2010.

[54] Anna Liddo, Sándor Ágnes, and Shum Buckingham. Contested collective intelligence: Rationale, technologies, and a human-machine annotation study. *Computer Supported Cooperative Work*, 21:417–448, 2012.

[55] Dan Liebling, Daniel Ramage, and Susan Dumais. Characterizing microblogs with topic models. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, pages 130–137. AAAI, 2010.

[56] Ellucian Company L.P. Ellucian Course Signals ellucian, 2013.

[57] D2L Ltd. Desire2Learn Insights desire2learn, 2013.

[58] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[59] Cheong Marc and Lee Vincent. Integrating web-based intelligence retrieval and decision-making from the twitter trends knowledge base. In *Proceedings of the 2Nd ACM Workshop on Social Web Search and Mining*, pages 1–8. ACM, 2009.

[60] Edgar Meij, Wouter Weerkamp, and Maarten de Rijke. Adding semantics to microblog posts. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 563–572. ACM, 2012.

[61] Pablo N Mendes, Alexandre Passant, Pavan Kapanipathi, and Amit P Sheth. Linked open social signals. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 1, pages 224–231. IEEE, 2010.

[62] Matthew Michelson and Sofus A Macskassy. Discovering users' topics of interest on twitter: a first look. In *Proceedings of the fourth workshop on Analytics for noisy unstructured text data*, pages 73–80. ACM, 2010.

[63] Dunworth Moira. Supporting students through social networking. *Journal of Practice Teaching & Learning*, 9:64–80, 2009.

[64] Moodle™. Welcome to the Moodle community! moodle, 2013.

[65] Hieu V Nguyen and Li Bai. Cosine similarity metric learning for face verification. In *Computer Vision–ACCV 2010*, pages 709–720. Springer, 2011.

[66] Jeffrey Nickerson, Yasuaki Sakamoto, and Yegin Genc. Discovering context: Classifying tweets through a semantic transform based on wikipedia. In *Proceedings of the 6th International Conference on Foundations of Augmented Cognition: Directing the Future of Adaptive Systems*, pages 484–492. Springer-Verlag, 2011.

[67] Douglas Oard and Tan Xu. Wikipedia-based topic clustering for microblogs. In *Proceedings of the American Society for Information Science and Technology*, volume 48, pages 1–10. Association for Computational Linguistics, 2011.

[68] Brendan O'Connor, Ramnath Balasubramanyan, Bryan Routledge, and Noah Smith. From tweets to polls : Linking text sentiment to public opinion time series. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, pages 122–129. AAAI Press, 2010.

[69] Brendan O'Connor, Michel Krieger, and David Ahn. Tweetmotif: Exploratory search and topic summarization for twitter. In *ICWSM*, 2010.

[70] Gephi org. The Open Graph Viz Platform gephi.

[71] Achananuparp Palakorn, He Jing, Jiang Jing, Li Xiaoming, Lim Ee-Peng, Song Yang, and Zhao Xin. Topical keyphrase extraction from twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 379–388. Association for Computational Linguistics, 2011.

[72] Juceviciene Palmira and Valineviciene Gintare. A conceptual model of social networking in higher education. 102, 2010.

[73] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2:1–135, 2008.

[74] Luo Ping, Shen Wei, Wang Jianyong, and Wang Min. Linking named entities in tweets with knowledge base via user interest modeling. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 68–76. ACM, 2013.

[75] prefuse. the prefuse visualization toolkit prefuse, 2012.

[76] Oxford University's Press. OUP Dictionary Team monitors Twitterer's tweets oxford university's press.

[77] Eric Prud'Hommeaux, Andy Seaborne, et al. Sparql query language for rdf. *W3C recommendation*, 15, 2008.

[78] He Qi, Lim Ee-Peng, Jiang Jing, and Weng Jianshu. Twitterrank: Finding topic-sensitive influential twitterers. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, pages 261–270. ACM, 2010.

[79] Kira Radinsky, Sagie Davidovich, and Shaul Markovitch. Learning causality for news events prediction. In *Proceedings of the 21st international conference on World Wide Web*, pages 909–918. ACM, 2012.

[80] Hugo Rosa, João Paulo Carvalho, and Fernando Batista. Detecting a tweet's topic within a large number of portuguese twitter trends. *InProceedings OASIcs*, page 4569, 2014.

[81] Claire Ross, Melissa Terras, Claire Warwick, and Anne Welsh. Enabled backchannel: conference twitter use by digital humanists. *Journal of Documentation*, 67(2):214–237, 2011.

[82] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[83] Bieke Schreurs and Maarten Laat. Network awareness tool - learning analytics in the workplace: Detecting and analyzing informal workplace learning. In *Proceedings of the 2Nd International Conference on Learning Analytics and Knowledge*, pages 59–64. ACM, 2012.

[84] JI Sheeba and K Vivekanandan. Improved keyword and keyphrase extraction from meeting transcripts. *International Journal of Computer Applications (0975–8887)*, 52(13):11–15, 2012.

[85] Saeedeh Shekarpour, Axel-Cyrille Ngonga Ngomo, and Sören Auer. Keyword-driven resource disambiguation over rdf knowledge bases. In *Semantic Technology*, pages 159–174. Springer, 2013.

[86] Yongwook Shin, Chuhyeop Ryo, and Jonghun Park. Automatic extraction of persistent topics from social text streams. *World Wide Web*, pages 1–26.

[87] Greg Sterling. Pew: 94% Of Teenagers Use Facebook marketing land.

[88] Michael Strube and Simone Paolo Ponzetto. Wikirelate! computing semantic relatedness using wikipedia. In *AAAI*, volume 6, pages 1419–1424, 2006.

[89] Horiguchi Susumu, Nguyen Le-Minh, and Phan Xuan-Hieu. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th International Conference on World Wide Web*, pages 91–100. ACM, 2008.

[90] Anwar Tarique. *A Supervised Learning Approach for Automatic Keyphrase Extraction*. PhD thesis, Jamia Millia Islamia (A Central University), 2010.

[91] Loralyn Taylor and Virginia McAleese. Using Targeted Analytics to Improve Student Success educause review online, 2012.

[92] Yee W Teh, David Newman, and Max Welling. A collapsed variational bayesian inference algorithm for latent dirichlet allocation. In *Advances in neural information processing systems*, pages 1353–1360, 2006.

[93] Ann Thompson and Denise Lindstrom. Social networking as a tool in teacher education courses. *Journal of Digital Learning in Teacher Education*, 27, 2010.

[94] Ken Thompson. Programming techniques: Regular expression search algorithm. *Communications of the ACM*, 11(6):419–422, 1968.

[95] Stella Tian, Angela Yu, Douglas Vogel, and Ron Kwok. The impact of online social networking on learning : a social integration perspective. *International Journal of Networking and Virtual Organisations*, 8:264–280, 2011.

[96] Ding Tu, Ling Chen, and Gencai Chen. Wordnet based multi-way concept hierarchy construction from text corpus. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

[97] Princeton University. About WordNet princeton university, 2010.

[98] Andrea Varga, Amparo Elizabeth Cano Basave, Matthew Rowe, Fabio Ciravegna, and Yulan He. Linked knowledge sources for topic classification of microposts: A semantic graph-based approach. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2014.

[99] E. Wagner. Argunaut research publications, 2012.

[100] Ellen Wagner and Phil Ice. Data Changes Everything: Delivering on the Promise of Learning Analytics in Higher Education educause review online, 2012.

[101] Chi Wang, Marina Danilevsky, Nihit Desai, Yinan Zhang, Phuong Nguyen, Thrivikrama Taula, and Jiawei Han. A phrase mining framework for recursive construction of a topical hierarchy. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 437–445. ACM, 2013.

[102] Tim Weninger, Yonatan Bisk, and Jiawei Han. Document-topic hierarchies from document graphs. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 635–644. ACM, 2012.

[103] Huang Wenyi, Liu Zhiyuan, Sun Maosong, and Zheng Yabin. Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 366–376. Association for Computational Linguistics, 2010.

[104] Edutech Wiki. Learning management system edutech, 2014.

[105] WikiMedia. A Java API to parse Wikipedia XML dumps wikixmlj, 2012.

[106] WikiMedia. API:Main page wikipedia, 2012.

[107] Wikipedia.org. Wikipedia:About wikimedia foundation.

[108] WikiProject. Wikipedia talk:WikiProject Categories wikipedia, 2015.

[109] Shirley A Williams, Melissa M Terras, and Claire Warwick. What do people study when they study twitter? classifying twitter related academic papers. *Journal of Documentation*, 69(3):384–410, 2013.

[110] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 334–342. ACM, 2001.

A Appendix

A.1 Evaluation

A.1.1 Topics Identification

Terms	
Topic 0	data - science - analytics - mobile - bigdata - interesting - sense - impact - predictive - media - platform - reality - virtual - life - mind
Topic 1	analytics - data - personal - latest - decisions - chief - heres - statistics - bigdata - human - finance - officer - deliver - tool - touch
Topic 2	data - bigdata - wrangling - value - tools - system - analytics - insurance - learning - case - technologies - machine - problem - market - justice
Topic 3	research - infographic - bigdata - dataviz - businessintelligence - panel - funding - businesses - ediscovery - guide - lead - coming - challenge - view - recherche
Topic 4	business - innovation - bigdat - service - register - potential - become - product - development - press - advantage - skills - study - available - chinese
Topic 5	cloud - scientist - learn - hadoop - bigdata - silicon - valley - devops - facebook - days - code - named - articles - cost - internetofthings
Topic 6	nosql - mongodb - startup - database - support - matters - community - algorithms - tomorrow - socialmedia - solution - wireless - bigdata - role - barcelona
Topic 7	cloudcomputing - digital - companies - cloud - intelligence - field - insight - bigdata - risk - data - technologymirror - analytics - technews - text - infrastructure
Topic 8	time - customer - information - analytics - data - company - bigdata - strategy
Topic 9	bigdata - join - machinelearning - solutions - part - databases - power - launches - working - mining - cybersecurity - financial - data - mlatmsft - leaders

Topic 10	experience - customers - hiring - software - performance - systems - jobs - improve - health - hadoop - opportunity - teams - team - bigdata - firms
Topic 11	bigdata - process - data - challenges - fast - realtime - faster - watch - cleaning - ebola - critical - datawrangling - distributed - article - building
Topic 12	internet - things - technology - hyped - internetofthings - bigdata - official - data - understand - network - target - users - organizations - collecting - predict
Topic 13	python - bigdata - talent - focus - want - series - nosql - raises - opensource - pour - couchbase - java - metrics - workplace - apply
Topic 14	healthcare - future - analysis - bigdata - success - data - parkinsons - strategy - intel - opendata - disease - apps - healthit - services - wearables
Topic 15	privacy - bigdata - sales - nice - love - myths - data - create - chicago - money - reddit - tips - anonymity - mean - uses
Topic 16	insights - scientists - janitor - hurdle - datascience - bigdata - data - times - york; myth - competitive - dataquality - datamanagement - datascientist - datascientists
Topic 17	social - blog - enterprise - architecture - people - management - dublin - lambda - nosql - smart - bigdata - benefits - datacenter - tickets - principles
Topic 18	marketing - bigdata - analytics - video - startups - data - change - para - projects care - sharing - interview - datadriven - security - requires
Topic 19	tech - hype - google - cycle - gartner - trends - bigdata - city - increase - fashion - melbourne - wants - sexy - story - partnership

Table A.1: Topical Terms of Experiment 1

	Terms
Topic 0	insights - scientists - janitor - hurdle - hurdle insights - bigdata scientists - scientists janitor - janitor hurdle - data scientists - scientists hurdle - insights janitor - times - datascience
Topic 1	analytics - data - personal - latest - decisions - chief - heres - statistics - bigdata - human - finance - officer - deliver - tool - touch
Topic 2	data - bigdata - wrangling - value - tools - system - analytics - insurance - learning - technologies - case - machine - problem - market - justice
Topic 3	research - infographic - bigdata - dataviz - businessintelligence - panel - funding - businesses - ediscovery - guide - lead - coming - challenge - view - recherche

Topic 4	business - innovation - bigdata - service - register - potential - become - product - development - press - advantage - skills - study - available - chinese
Topic 5	cloud - scientist - learn - hadoop - bigdata - silicon - valley - devops - facebook - days - code - named - articles - cost - internetofthings
Topic 6	nosql - mongodb - startup - database - support - matters - community - algorithms - tomorrow - socialmedia - solution - wireless - bigdata - role - barcelona
Topic 7	cloudcomputing - digital - companies - cloud - intelligence - field - insight - bigdata - technews - data - technologymirror - analytics - risk - text - infrastructure
Topic 8	time - customer - information - analytics - data - company - report - bigdata - retail - center - strategy - start - knowledge - spend - loyalty
Topic 9	bigdata - join - machinelearning - solutions - part - databases - power - launches - working - mining - cybersecurity - financial - data - mlatmsft - leaders
Topic 10	experience - customers - hiring - software - performance - systems - jobs - improve - health - hadoop - opportunity - teams - team - bigdata - firms
Topic 11	bigdata - process - data - challenges - fast - realtime - faster - watch - cleaning - ebola - critical - datawrangling - distributed - article - building
Topic 12	internet - things - technology - hyped - internetofthings - bigdata - official - data - understand - network - target - users - organizations - collecting - predict
Topic 13	python - bigdata - talent - focus - want - series - nosql - raises - opensource - pour - couchbase - java - metrics - workplace - apply
Topic 14	healthcare - future - analysis - bigdata - success - data - parkinsons - strategy - intel - opendata - disease - apps - healthit - services - wearables
Topic 15	privacy - bigdata - sales - nice - love - myths - data - create - chicago - money - reddit - tips - anonymity - mean - uses
Topic 16	insights - scientists - janitor - hurdle - datascience - bigdata - data - times - york - datascientist - competitive - dataquality - datamanagement - myth - datascientists
Topic 17	social - blog - enterprise - architecture - people - management - dublin - lambda - nosql - smart - bigdata - benefits - datacenter - tickets - principles
Topic 18	marketing - bigdata - analytics - video - startups - data - change - para - projects - care - interview - datadriven - security - requires - sharing
Topic 19	tech - hype - google - cycle - gartner - trends - bigdata - city - increase - fashion - melbourne - wants - sexy - story - partnership

Table A.2: Topical Terms of Experiment 2

	Terms
Topic 0	bigdata - data - research - video - care - companies - want - days - wearable - reddit - money - mean - devices - hpbigdata - challenges
Topic 1	bigdata - data - strategy - value - success - para - firms - project - loyalty - trends
Topic 2	internet - things - technology - bigdata - hyped - hype - official - cycle - gartner - emerging - internetofthings - tech - technologies - bluemix - ediscovery
Topic 3	bigdata - cloud - company - data - cloudcomputing - services - technews - interview - interesting - technologymirror - build - security - telco - fujitsu - provider
Topic 4	bigdata - mobile - marketing - data - experience - customer - information - improve - science - sports - analytics - teams - impact - digital - field
Topic 5	bigdata - data - change - future - sharing - sense - visualization - state - statistical - thats - model - economy - relationships - benefits - fight
Topic 6	business - bigdata - intelligence - data - blog - system - opendata - problem - justice - press - researchers - datacenter - watch - management - poor
Topic 7	bigdata - learn - data - increase - facebook - cloudcomputing - love - chicago - market - events - manage - worth - businessintelligence - rapid - platform
Topic 8	datascience - scientist - data - hiring - bigdata - python - jobs - nosql - matters - databases - dublin - statistics - challenge - articles - programming
Topic 9	nosql - hadoop - mongodb - performance - database - community - java - support - bigdata - opensource - sept - guide - text - webcast - cassandra
Topic 10	bigdata - machinelearning - join - data - chief - tomorrow - officer - life - meet - microsoft - info - mlatmsft - come - lead - tweetchat
Topic 11	bigdata - data - city - fashion - melbourne - socialmedia - algorithm - myth - dataquality - datascientist - partnership - sign - brands - insight - send
Topic 12	analytics - bigdata - power - infographic - team - machine - disease - learning - parkinsons - predictive - applications - businesses - data - ready - building
Topic 13	bigdata - analytics - healthcare - tech - industry - start - insurance - human - decisions - healthit - datadriven - utility - future - funding - realtime
Topic 14	data - systems - bigdata - dataviz - privacy - health - knowledge - financial
Topic 15	startup - google - customers - retail - data - opportunity - privacy - personal - media - management - tech - heres - focus - bigdata - content
Topic 16	bigdata - enterprise - architecture - sales - projects - process - lambda - datastax - software - patient - reality - virtual - storage - product - tools

Topic 17	bigdata - analysis - internetofthings - innovation - silicon - valley - devops - data - talent - named - sponsor - launches - startups - session - sentiment
Topic 18	data - bigdata - time - social - wrangling - understand - network - part - spend - users - target - collecting - types - tools - raises
Topic 19	insights - scientists - janitor - hurdle - bigdata - data - times - york - datawrangling - story - offers - datascientists - bigdatabrasil - brasil - piece

Table A.3: Topical Terms of Experiment 3

Terms	
Topic 0	bigdata - data - science - customers - data science - part - projects - opportunity - analytics - impact - trends - change - reddit - critical - market
Topic 1	data - social - privacy - sense - analytics - network - understand - teams - data social - socialmedia - target - users - improve - reality - virtual
Topic 2	bigdata - data - bigdata cloud - time - data bigdata - fast - fast data - cases - become - code - data brasil - brasil - bigdatabrasil - crie - love
Topic 3	nosql - data - community - center - bigdata - matters - dublin - support - database - data center - knowledge - tickets - changes - nosql matters - agenda
Topic 4	bigdata - google - data - management - facebook - startup - want - wants - wireless - sexy - partnership - vegas - bigdata privacy - sale - businessintelligence
Topic 5	bigdata - data - data bigdata - hadoop - systems - insight - video - content - measure - mean - money - startup - give - future - hope
Topic 6	bigdata analytics - analytics - nosql - hadoop - mongodb - hiring - python - jobs - performance - java - bigdata nosql - service - databases - software - bluemix
Topic 7	bigdata - cloudcomputing - data - para - tech - bigdata cloudcomputing - technews - technologymirror - technews technologymirror - datascience - risk - negocio - dangerous - cloudcomputing datascience - rentables
Topic 8	data - scientist - machinelearning - data scientist - bigdata - information - bigdata datascience - join - learning - machine - chief - machinelearning bigdata - datascience - process - tomorrow
Topic 9	internet - bigdata - things - technology - internet things - hyped - hyped technology - official - internetofthings - bigdata hyped - things bigdata - internetofthings bigdata - official internet - data - things data

Topic 10	analytics - healthcare - data - datascience - health - analytics bigdata - startups - industry - bigdata - care - future - healthit - apps - patient - statistics
Topic 11	bigdata - data - analytics - companies - bigdata data - decisions - talent - time - people predict - conference - quality - media - talent analytics - analyze
Topic 12	bigdata - mobile - learn - research - sales - parkinsons - innovation - intel - disease - sports - field - wearables - enterprise - parkinsons disease - datastax
Topic 13	insights - scientists - janitor - hurdle - hurdle insights - bigdata scientists - scientists janitor - janitor hurdle - data scientists - data - wrangling - datascience - scientists hurdle - insights janitor - insights bigdata
Topic 14	bigdata - business - marketing - intelligence - business intelligence - digital - opendata - platform - bigdata marketing - startup - bigdata business - solution - pour - myths - fujitsu
Topic 15	bigdata - data - architecture - system - lambda - lambda architecture - article - problem - insurance - analytics - changing - test - justice system - justice - principles
Topic 16	cloud - cloud bigdata - bigdata - hype - internetofthings - tech - business - cycle - hype cycle - dataviz - silicon - silicon valley - valley - devops - gartner
Topic 17	bigdata - analysis - power - city - fashion - melbourne - data - city melbourne - review - pitfalls - emerging - pitfalls nowcasting - nowcasting - emerging pitfalls - sentiment
Topic 18	bigdata - customer - data - experience - value - blog - retail - personal - services - blog bigdata - customer experience - management - loyalty - finance - interview
Topic 19	bigdata - data - analytics - data analytics - strategy - start - analytics bigdata - success - utility - firms - capital - industries - data strategy - planning - bigdata strategy

Table A.4: Topical Terms of Experiment 4

	Terms
Topic 0	analytics - data analytics - analytics bigdata - cloudcomputing - startup - bigdata - startups - predictive - talent - platform - bigdata cloudcomputing - datadriven - technews - launches - predictive analytics
Topic 1	bigdata - business - data - customers - start - opportunity - decisions - smart - want - utility - requires - become - helps - watch - opportunity customers
Topic 2	nosql - mongodb - hiring - python - performance - database - software - jobs - java - matters - dublin - databases - community - bigdata nosql - service

Topic 3	internet - things - technology - internet things - hyped - internetofthings - hyped technology - hype - official - bigdata hyped - cycle - hype cycle - gartner - things bigdata - internetofthings bigdata
Topic 4	data - bigdata - data bigdata - innovation - field - bigdata innovation - care - lake - data lake - manage - energy - mean - money - champions - workplace
Topic 5	bigdata - datascience - data - scientist - data scientist - bigdata datascience - data-science bigdata - machine - join - statistics - learning - tomorrow - ediscovery - machinelearning - solutions
Topic 6	cloud - customer - cloud bigdata - bigdata cloud - google - silicon - sense - valley - silicon valley - internetofthings - devops - bigdata internetofthings - valley cloud - customer experience - named
Topic 7	bigdata analytics - social - analytics - science - article - data - part - data science - process - media - network - understand - data social - target - users
Topic 8	data - intelligence - value - bigdata - experience - improve - business intelligence - system - teams - chief - problem - sports - chief data - justice - justice system
Topic 9	data - hadoop - bigdata - bigdata data - future - systems - projects - hadoop bigdata - data wrangling - project - guide - code - webinar - wrangling - bigdata projects
Topic 10	bigdata - privacy - power - research - video - parkinsons - latest - intel - team - disease - focus - insight - case - panel - parkinsons disease
Topic 11	bigdata - data - marketing - time - blog - management - tools - increase - opendata - blog bigdata - data marketing - press - bigdata marketing - myths - datacenter
Topic 12	bigdata - mobile - learn - socialmedia - impact - data - businessintelligence - learn bigdata - content - come - meet - company - build - lead - weve
Topic 13	bigdata - data - infographic - interesting - para - change - report - insurance - changing - ebola - life - negocio - creates - rentables - negocio rentables
Topic 14	data - bigdata - strategy - retail - success - personal - interview - mining - capital - working - firms - applications - financial - competitive - data strategy
Topic 15	insights - scientists - janitor - hurdle - hurdle insights - bigdata scientists - scientists janitor - janitor hurdle - data scientists - scientists hurdle - insights janitor - insights bigdata - times - york - york times
Topic 16	bigdata - sales - enterprise - city - fashion - melbourne - datastax - algorithms - company - city melbourne - chicago - product - helping - trust - vision

Topic 17	bigdata - digital - companies - architecture - facebook - lambda - lambda architecture - storage - reddit - principles - view - microsoft - programming - architecture principles - going
Topic 18	bigdata - tech - analysis - machinelearning - dataviz - trends - center - services - google - challenges - machinelearning bigdata - data center - model - test - knowledge
Topic 19	healthcare - health - information - wrangling - time - support - spend - healthit - spend time - conference - patient - bigdata - datas - predict - million

Table A.5: Topical Terms of Experiment 5

Terms	
Topic 0	bigdata - tech - hype - hiring - python - cycle - hype cycle - jobs -gartner - software - java - team - challenges - technologies - emerging
Topic 1	insights - scientists - janitor - hurdle - hurdle insights - bigdata scientists - scientists janitor -janitor hurdle - data scientists - scientists hurdle - insights janitor - times - york - insights bigdata - york times
Topic 2	data - data bigdata - data analytics - science - part - data science - process - article - critical - analytics - cleaning - mining - streaming - energy - touch
Topic 3	business - bigdata - startup - google - intelligence - systems - management - business intelligence - requires - bigdata business - smart - data - helps - wants - businessintelligence
Topic 4	bigdata - privacy - parkinsons - intel - disease - platform - panel - wearables - parkinsons disease - become - market - test - wearable - million - creates
Topic 5	bigdata - hadoop - company - increase - opendata - hadoop bigdata - insight - human - press - step - data - pour - product - brief -opportunities
Topic 6	bigdata - insurance - service - data - webinar - webcast - leading - going - august - benefits - leveraging - workplace - state - inmemory - visualization
Topic 7	data - time - scientist - wrangling - data scientist - information - socialmedia - data wrangling - bigdata - tools - spend - spend time - time data - data janitor - algorithm
Topic 8	datascience - bigdata - customer - learn - machinelearning - experience - bigdata datascience - join - datascience bigdata - city - teams - machine - learning - melbourne - fashion

Topic 9	nosql - technology - things - hyped - internet things - internet - hyped technology - official - mongodb - bigdata hyped - performance - database - things bigdata - official internet - sports
Topic 10	bigdata - cloudcomputing - research - startups - dataviz - media - bigdata cloudcomputing - funding - technews - technologymirror - technews technologymirror - risk - bigdata dataviz - case - applications
Topic 11	bigdata - strategy - digital - data - success - companies - capital - tips - working - firms - ways - finance - reddit - series - financial
Topic 12	analytics - analytics bigdata - healthcare - value - industry - power - sense - bigdata - healthit - potential - apps - patient - reality - virtual - mind
Topic 13	bigdata - bigdata analytics - analytics - infographic - report - latest - services - para - bluemix - nice - ediscovery - datadriven - ediscovery bigdata - solutions - trust
Topic 14	cloud - internetofthings - internet - cloud bigdata - things - innovation - internet things - bigdata cloud - bigdata - silicon - internetofthings bigdata - valley - silicon valley - devops - bigdata internetofthings
Topic 15	bigdata - data - analysis - health - video - care - interview - want - focus - perspective - events - hpbigdata - mean - money - manage
Topic 16	data - bigdata - blog - enterprise - center - talent - blog bigdata - project - guide - data center - knowledge - review - solution - pitfalls - datacenter
Topic 17	bigdata - mobile - data - bigdata data - future - sales - change - facebook - heres - decisions - bigdata nosql - datastax - conference - predict - life
Topic 18	social - start - system - network - understand - data social - target - databases - utility - users - problem - changing - faster - collecting - challenge
Topic 19	data - marketing - bigdata - customers - retail - architecture - opportunity - lambda - interesting - chief - lambda architecture - realtime - data marketing - loyalty - chief data

Table A.6: Topical Terms of Experiment 6

	Terms
Topic 0	bigdata - business - analysis - intelligence - systems - future - para - business intelligence - facebook - businessintelligence - bigdata business - panel - mobile - measure - negocio

Topic 1	bigdata - innovation - customers - python - opportunity - projects - bigdata innovation - opendata - focus - ediscovery - platform - press - opensource - solutions - opportunity customers
Topic 2	datascience - marketing - healthcare - bigdata - cloudcomputing - bigdata datascience - industry - impact - datascience bigdata - bigdata cloudcomputing - healthit - apps - technews - patient - data marketing
Topic 3	bigdata - social - science - data - data science - process - understand - network - data social - users - target - collecting - life - solution - critical
Topic 4	data - data bigdata - health - value - field - increase - care - start - machine - learning - bigdata - loyalty - algorithms - market - review
Topic 5	bigdata - analytics bigdata - infographic - interesting - trends - data - decisions - talent - chief - chief data - launches - helping - officer - security - talent analytics
Topic 6	internet - things - technology - internet things - hyped - internetofthings - hyped technology - hype - tech - official - bigdata hyped - cycle - hype cycle - gartner - things bigdata
Topic 7	bigdata - machinelearning - research - information - sense - change - interview - machinelearning bigdata - insurance - funding - reality - virtual - datas - mind - sharing
Topic 8	bigdata - startups - tools - video - report - nice - bluemix - become - smart - competitive - love - applications - advantage - bigdata startups - paas
Topic 9	analytics - bigdata analytics - data analytics - system - predictive - human - datadriven - problem - predictive analytics - justice - justice system - businesses - bigdata - bringing - punished
Topic 10	privacy - startup - google - management - social - media - city - services - databases - melbourne - fashion - bigdata - wants - test - city melbourne
Topic 11	bigdata - learn - power - enterprise - parkinsons - intel - disease - wearables - parkinsons disease - learn bigdata - product - wearable - cybersecurity - industries - creates
Topic 12	data - bigdata - mobile - part - series - financial - money - mean - leaders - bigdatabrasil - brasil - data brasil - growth - picture - crie
Topic 13	bigdata - data - blog - center - software - realtime - blog bigdata - data center - knowledge - risk - bigdata bigdata - dangerous - changes - datamanagement - dataquality

Topic 14	nosql - mongodb - hiring - performance - database - bigdata nosql - jobs - dataviz - sales - community - support - bigdata - team - datastax - nosql database
Topic 15	insights - scientists - janitor - hurdle - hurdle insights - bigdata scientists - scientists janitor - janitor hurdle - data scientists - scientists hurdle - wrangling - insights janitor - insights bigdata - times - york
Topic 16	bigdata - retail - socialmedia - personal - latest - matters - heres - algorithm - online - sept - content - days - finance - ebola - million
Topic 17	cloud - hadoop - cloud bigdata - strategy - bigdata cloud - join - success - silicon - silicon valley - valley - devops - internetofthings - bigdata - bigdata internetofthings - hadoop bigdata
Topic 18	bigdata - customer - scientist - experience - data scientist - digital - article - companies - architecture - improve - teams - lambda - lambda architecture - customer experience - sports
Topic 19	data - bigdata - time - bigdata data - mining - conference - predict - lake - data lake - insight - quality - energy - manage - summit - comes

Table A.7: Topical Terms of Experiment 7

	Terms
Topic 0	bigdata - people - customers - hadoop - enterprise - want - article - bigdata hadoop - problem - workshop - spark - development - understand - reactive - applications
Topic 1	data - bigdata - data bigdata - smart - para - fast - sense - virtual - turning - companies - working - fast data - smart data - data data - youre
Topic 2	bigdata - predictive - start - apply - industry - intel - helps - predictive analytics - ediscovery - smarter - data - health - update - case - medicine
Topic 3	bigdata - data - startup - infographic - challenge - experts - huge - consumer - protect - water - explain - value - authentiweb - congress - grandmother
Topic 4	internet - things - bigdata - internet things - technology - internetofthings - hyped - internetofthings bigdata - tech - things bigdata - energy - hyped technology - bigdata hyped - official - talent
Topic 5	marketing - bigdata - register - times - jobs - code - bigdata marketing - talking - datastax - time - database - sales - succeed - steve - solution

Topic 6	bigdata - insights - scientists - power - janitor - hurdle - hurdle insights - bigdata scientists - data scientists - janitor hurdle - scientists janitor - raises - farmlink - sensors - company
Topic 7	bigdata - research - machinelearning - deal - opportunity - datos - project - pour - datamining - bigdata datamining - machinelearning bigdata - pivotal - process - mobile - massive
Topic 8	bigdata - data - future - changing - social - cant - insurance - mistaken - assumptions - mistaken assumptions - storage - comes - uses - predict - network
Topic 9	bigdata - hadoop - blog - strategy - customer - join - experience - improve - bigdata strategy - care - management - conference - blog bigdata - mobile - healthcare
Topic 10	data - bigdata - scientist - data scientist - chief - officer - overhyped -chief data - actual - drive - actual scientists - advantage - list - overhyped actual - data officer
Topic 11	bigdata - system - going - heres - time - data - sports - stay - suite - relevant - watch - lives - field - botmaker - details
Topic 12	bigdata - learn - data - privacy - services - google - success - interesting - critical - bigdata privacy - ehealth - learn bigdata - opendata - whitepaper - available
Topic 13	nosql - tools - mongodb - engineer - software - nosqlnow - database - hiring - support - jobs - java - apps - ready - oracle - developer
Topic 14	bigdata - tech - datadriven - education - analysis - classroom - modern - students - teachers - modern classroom - students teachers - classroom students - part - bigdata tech - teachers datadriven
Topic 15	bigdata - business - intelligence - business intelligence - biggest - brings - market - chain - tech - ways - intelligence bigdata - supply - buzzwords - biggest buzzwords - supply chain
Topic 16	bigdata - data - datascience - bigdata datascience - datascience bigdata - science - mining - opportunities - government - marketing - rstats - media - data science - analytics datascience - data mining
Topic 17	bigdata - data - hype - retail - gartner - cycle - hype cycle - turn - datas - turn data - reasons - helping - avoid - plan - gartner hype
Topic 18	bigdata - cloud - cloud bigdata - bigdata cloud - security - data - webinar - mobile - search - devops - social - impact - daily - change - grow
Topic 19	analytics - bigdata analytics - bigdata - data analytics - analytics bigdata - realtime - video - forecasting - weather - leveraging - healthcare - enables - weather forecasting - banking - healthcare bigdata

Table A.8: Topical Terms of Testing Experiment

Topic Identification Evaluation

Topic #	Useful	Not Useful	P_i
Topic 0	2	0	1
Topic 1	1	1	0
Topic 2	2	0	1
Topic 3	1	1	0
Topic 4	2	0	1
Topic 5	2	0	1
Topic 6	2	0	1
Topic 7	2	0	1
Topic 8	1	1	0
Topic 9	2	0	1
Topic 10	2	0	1
Topic 11	0	2	1
Topic 12	2	0	1
Topic 13	2	0	1
Topic 14	2	0	1
Topic 15	2	0	1
Topic 16	2	0	1
Topic 17	1	1	0
Topic 18	2	0	1
Topic 19	2	0	1
Total	34	6	
P_j	0.85	0.15	
Kappa Coefficient	0.2156		

Table A.9: The inter-annotator agreement by k-coefficient of experiment 8

A.1.2 Mapping Clusters to Wikipedia

Wikipedia Articles	
Query 0	http://en.wikipedia.org/wiki/Agile_Business_Intelligence http://en.wikipedia.org/wiki/Business_analytics ...
Query 1	http://en.wikipedia.org/wiki/Stealth_mode http://en.wikipedia.org/wiki/Grant_management_software http://en.wikipedia.org/wiki/Context_analysis
Query 2	http://en.wikipedia.org/wiki/Industry_or_market_research http://en.wikipedia.org/wiki/Marketing_strategy http://en.wikipedia.org/wiki/Marketing_effectiveness
..	...
..	...
Query 17	http://en.wikipedia.org/wiki/Cloud_database http://en.wikipedia.org/wiki/Cloud_computing_issues http://en.wikipedia.org/wiki/Hue_(Hadoop) http://en.wikipedia.org/wiki/Cloud_management
Query 19	http://en.wikipedia.org/wiki/Real-time_business_intelligence http://en.wikipedia.org/wiki/Data_warehouse ...

Table A.10: Scored Wikipedia Articles - Setting 1

Wikipedia Articles	
Query 0	http://en.wikipedia.org/wiki/Mobile_business_intelligence http://en.wikipedia.org/wiki/Business_intelligence ...
Query 1	http://en.wikipedia.org/wiki/Focus_group http://en.wikipedia.org/wiki/Prime_Focus_Technologies_(PFT)
Query 2	http://en.wikipedia.org/wiki/Marketing_mix_modeling http://en.wikipedia.org/wiki/Marketing_plan http://en.wikipedia.org/wiki/Marketing_performance_measurement_and_management
..	...
..	...
Query 17	http://en.wikipedia.org/wiki/Cloud_database http://en.wikipedia.org/wiki/Apache_Hadoop ...

Query 19	http://en.wikipedia.org/wiki/Data_mining http://en.wikipedia.org/wiki/Quality_of_Data_(QoD)

Table A.11: Scored Wikipedia Articles - Setting 2

Wikipedia Articles	
Query 0	http://en.wikipedia.org/wiki/Mobile_business_intelligence http://en.wikipedia.org/wiki/Business_intelligence ...
Query 1	http://en.wikipedia.org/wiki/Data_discovery http://en.wikipedia.org/wiki/Big_data ...
Query 2	http://en.wikipedia.org/wiki/Cloud_computing_issues http://en.wikipedia.org/wiki/Health_care_analytics ..
Query 3	http://en.wikipedia.org/wiki/Social_media_mining http://en.wikipedia.org/wiki/Social_BI ...
Query 4	http://en.wikipedia.org/wiki/Big_data http://en.wikipedia.org/wiki/Machine_learning ..
..	...
..	...
Query 17	http://en.wikipedia.org/wiki/Internet_of_Things ...
Query 19	http://en.wikipedia.org/wiki/Lambda_architecture http://en.wikipedia.org/wiki/Quality_of_Data_(QoD) http://en.wikipedia.org/wiki/Digital_strategy ...

Table A.12: Scored Wikipedia Articles - Setting 3

Mapping Clusters to Wikipedia - Evaluation

Topic #	Wikipedia Article	Related	Not Related	P_i
Topic 0	Article 1	2	0	1
	Article 2	2	0	1
Topic 1	Article 1	1	1	0
	Article 2	0	2	1
Topic 2	Article 1	1	1	0
	Article 2	1	1	0
Topic 3	Article 1	2	0	1

	Article 2	1	1	0
Topic 4	Article 1	1	1	0
	Article 2	2	0	1
Topic 5	Article 1	1	1	0
	Article 2	2	0	1
Topic 6	Article 1	2	0	1
	Article 2	2	0	1
Topic 7	Article 1	2	0	1
	Article 2	2	0	1
Topic 8	Article 1	2	0	1
	Article 2	0	2	1
Topic 9	Article 1	2	0	1
	Article 2	2	0	1
Topic 10	Article 1	2	0	1
	Article 2	2	0	1
Topic 11	Article 1	1	1	0
	Article 2	1	1	0
Topic 12	Article 1	1	1	0
	Article 2	1	1	0
Topic 13	Article 1	2	0	1
	Article 2	2	0	1
Topic 14	Article 1	2	0	1
	Article 2	2	0	1
Topic 15	Article 1	2	0	1
	Article 2	2	0	1
Topic 16	Article 1	2	0	1
	Article 2	2	0	1
Topic 17	Article 1	2	0	1
	Article 2	1	1	0
Topic 18	Article 1	2	0	1
	Article 2	1	1	0
Topic 19	Article 1	2	0	1
	Article 2	2	0	1
Total		64	16	

P_j	0.8	0.2
Kappa Coefficient		0.0625

Table A.13: The inter-annotator agreement by k-coefficient of article's relevancy

Query#	Ranked Articles	Relevant	Precsion@1	Precision@2	AveragePrecision
1	1	1	1	1	1
	2	1			
2	1	1	1	0	0.5
	2	0			
3	1	1	1	1	1
	2	1			
4	1	1	1	1	1
	2	1			
5	1	1	1	1	1
	2	1			
6	1	1	1	0	0.5
	2	0			
7	1	1	1	0	0.5
	2	0			
8	1	1	1	1	1
	2	1			
9	1	1	1	1	1
	2	1			
10	1	1	1	0	0.5
	2	0			
11	1	1	1	0	0.5
	2	0			
12	1	1	1	0	0.5
	2	0			
13	1	1	1	0	0.5
	2	0			
14	1	1	1	1	1
	2	1			

15	1 2	1 1	1	1	1
16	1 2	1 1	1	1	1
17	1 2	1 1	1	1	1
18	1 2	1 0	1	0	0.5
19	1 2	1 1	1	1	1
20	1 2	1 1	1	1	1
				MAP	**0.8**

Table A.14: Google Evaluation using MAP

A.1.3 Constructing the Topic Network

Identifying Nodes: Mapping Miro-blogs to Topics

Wikipedia Category	Wikipedia Article	Related	Not Related	P_i
Customer experience management	Digital_strategy	2	0	1
	Lambda architecture	2	0	1
Apple personal digital assistants	Social_media_mining	2	0	1
	Social_BI	2	0	1
Statistical natural language processing	Analytics	2	0	1
Data analysis software	Social_BI	2	0	1
	Wearable_computer	0	2	1
	Supplier_Risk_Management	1	1	0
Systems science	Big_data	1	1	0
	Social_BI	0	2	1
	Data_mining	2	0	1
	Quality_of_Data_(QoD)	1	1	0
Distributed data storage systems	Big_data	2	0	1
Data mining and machine learning software	Data_discovery	1	1	0

	Health_care_analytics	1	1	0
RDBMS software for Linux	MongoDB	2	0	1
Environmental education video games	Competitive_intelligence	0	2	1
Emerging standards	Analytics	1	1	0
Mobile software programming tools	Data_discovery	0	2	1
Total		24	14	
P_j		0.63	0.36	
Kappa Coefficient			0.337	

Table A.15: Article to Wikipedia Category Evaluation

A.1.4 End To End Evaluation

ID	Wikipedia Category	Related	Not Related	Pi
1	Customer experience management	2	0	1
2	Statistical_natural_language_processing	2	0	1
3	Data_analysis_software	2	0	1
4	RDBMS software for Linux	2	0	1
5	Data_mining_and_machine_learning_software	2	0	1
6	Data_analysis_software	2	0	1
7	Data_mining_and_machine_learning_software	1	1	0
8	RDBMS software for Linux	2	0	1
9	Data_mining_and_machine_learning_software	2	0	1
10	Statistical_natural_language_processing	0	2	1
11	Data_mining_and_machine_learning_software	2	0	1
12	Data_mining_and_machine_learning_software	2	0	1
13	Customer experience management	1	1	0
14	Data_mining_and_machine_learning_software	1	1	0
15	Data_mining_and_machine_learning_software	2	0	1
16	Data_analysis_software	2	0	1
17	Data_mining_and_machine_learning_software	2	0	1
18	Apple personal digital assistants	2	0	1
19	RDBMS software for Linux	2	0	1
20	Distributed data storage systems	2	0	1
21	Customer experience management	2	0	1

22	Data_analysis_software	2	0	1
23	RDBMS software for Linux	2	0	1
24	Emerging standards	2	0	1
25	Customer experience management	2	0	1
26	Statistical_natural_language_processing	2	0	1
27	Mobile_business_software	1	1	0
28	Distributed data storage systems	1	1	0
29	Distributed data storage systems	2	0	1
30	Emerging standards	2	0	1
31	Statistical_natural_language_processing	2	0	1
32	Systems science	2	0	1
33	Environmental_education_video_games	2	0	1
34	Customer experience management	2	0	1
35	Emerging standards	2	0	1
36	Data_mining_and_machine_learning_software	2	0	1
37	Data_analysis_software	2	0	1
38	Customer experience management	0	2	1
39	Customer experience management	0	2	1
40	Systems science	2	0	1
41	Environmental_education_video_games	1	1	0
42	Emerging standards	2	0	1
43	Distributed data storage systems	2	0	1
44	Data_mining_and_machine_learning_software	2	0	1
45	Emerging standards	2	0	1
46	Data_mining_and_machine_learning_software	2	0	1
47	Distributed data storage systems	2	0	1
48	Data_analysis_software	2	0	1
49	Data_mining_and_machine_learning_software	2	0	1
50	Distributed data storage systems	2	0	1
51	Data_mining_and_machine_learning_software	2	0	1
52	Data_mining_and_machine_learning_software	0	2	1
53	Customer experience management	0	2	1
54	Data_mining_and_machine_learning_software	2	0	1
55	Systems science	2	0	1

56	Data_mining_and_machine_learning_software	1	1	0
57	Data_mining_and_machine_learning_software	2	0	1
58	Data_mining_and_machine_learning_software	2	0	1
59	Data_mining_and_machine_learning_software	1	1	0
60	Data_mining_and_machine_learning_software	2	0	1
61	RDBMS software for Linux	2	0	1
62	Distributed data storage systems	2	0	1
63	Emerging standards	2	0	1
64	Mobile_software_programming_tools	2	0	1
65	Mobile_business_software	2	0	1
66	Customer experience management	2	0	1
67	RDBMS software for Linux	2	0	1
68	Environmental_education_video_games	2	0	1
69	Data_analysis_software	2	0	1
70	Emerging standards	2	0	1
71	Customer experience management	2	0	1
72	Data_mining_and_machine_learning_software	2	0	1
73	Emerging standards	2	0	1
74	Environmental_education_video_games	1	1	0
75	Environmental_education_video_games	1	1	0
76	Systems science	2	0	1
77	Mobile_software_programming_tools	2	0	1
78	Customer experience management	2	0	1
79	Customer experience management	2	0	1
80	Mobile_software_programming_tools	2	0	1
81	Apple personal digital assistants	2	0	1
82	RDBMS software for Linux	2	0	1
83	Data_analysis_software	2	0	1
84	Distributed data storage systems	1	1	0
85	Data_mining_and_machine_learning_software	2	0	1
86	Data_analysis_software	2	0	1
87	Apple personal digital assistants	2	0	1
88	Data_mining_and_machine_learning_software	2	0	1
89	Data_mining_and_machine_learning_software	2	0	1

90	Systems science	1	1	0
91	Data_analysis_software	2	0	1
92	Statistical_natural_language_processing	2	0	1
93	Environmental_education_video_games	0	2	1
94	Statistical_natural_language_processing	2	0	1
95	Apple personal digital assistants	2	0	1
96	Customer experience management	2	0	1
97	Emerging standards	2	0	1
98	Data_analysis_software	2	0	1
99	Customer experience management	0	2	1
100	Statistical_natural_language_processing	2	0	1
	Total	174	26	88
	Pj	0.87	0.13	
	Kappa Coefficient	0.469		

Table A.16: Tweet to Category Evaluation

Tweet Collection Evaluated

ID	Tweet
1	Good Read >> Content Marketing Campaign Crushes Kapostâ€™s Average Lead Gen Totals By 452% Liz #BigData #Analytics http://t.co/NY8fXphE1u
2	Better data could save lives and make sense of our world, says academic http://t.co/Z0dWRm5jkw #bigdata
3	RT @ICOnews: We've released our first #BigData report. Find out why #DPA shouldn't be seen as a barrier to innovation http://t.co/iiRjeKEnPA
4	EnterpriseDB Makes Agile NoSQL Development Easy With New Postgres AWS ... - Marketwired (press... http://t.co/fyOkzWJ8aX #nosql #bigdata
5	Is it sad that I would love to see a @googleanalytics for a Tesco self service machine. #stats #bigdata
6	RT:@openrsrchhttp://t.co/EUeeQAnCqo #joe_vassily_02 #bigdata #big_data
7	Good #Data better than #BigData: http://t.co/TdsoYy5Udp #google #forecasting #machinelearning
8	RT @DataMdlRockStar: Are there connections between #DataGovernance and #NoSQL? Take the next #data modeling challenge to find out: http://tâ€¦
9	#BigData in the classroom: Why learning will never be the same! http://t.co/BwnxlluFwh #education #data via @LinkedIn , @BernardMarr
10	#BigData and The Crucial Need for #Information #Governance http://t.co/zXmND5x3oG #infogov #records #management #ECM
11	RT:@atoms2bitsWhat does #BigData mean for #health #care? http://t.co/w2fMoI8t4X
12	RT @cammyv: New to #machinelearning? Azure and our new MLU can help! http://t.co/Tqi9MRSrIJ #MLatMSFT #azureml #bigdata #datascience
13	New tech toys in the UK office. Who said #Bigdata and #Analytics was no fun?! #BI #data #BIOffice http://t.co/XDZD9IfY3q
14	Early creative skills indicate intelligence in teen years http://t.co/Og1OHwlPuh #BI #BigData
15	Machine Learning predicts heart attacks 4 hours before doctors! http://t.co/8d3SqqflrO #BigData
16	Contact me & #Learn how #BigData #analytics within IT #Operations can make you a #Hero at our events in #Sydney & #Melbourne this #September
17	RT @davidtalby: #BigData #Analytics Algorithm Predicted West Africaâ€™s Ebola Outbreak Before WHO http://t.co/Tp7zUixo4N
18	#BigData doesn't need to be #BigDrama: incident management and pipeline management are made easy with QlikView https://t.co/gPQ2RBSRQp
19	RT @bobson_pl: Zapraszam na now? konferencj?! http://t.co/5zojpWCr1r #linux #database #dba #db2 #mongodb #postgresql #mysql #nosql #plug â€¦
20	RT @TopQuadrant: TopQuadrant's @bobdc is presenting at #SemTechBiz - "Semantic Web Standards & Variety 'V' #BigData" 8/20 10:15-11am http:/â€¦
21	Turn shapeless #data into actionable insights: @Capgemini #BI, #bigdata and #analytics News on #LinkedIn http://t.co/DlK1mPZj6c
22	Can #BigData cure #cancer? No. But it can help many patients beat their disease.shttp://bit.ly/1yTKMv3 http://t.co/askfTWJJ4K
23	EnterpriseDB chucks devs free tools for building NoSQL web apps with ... - Register http://t.co/AHWRykWurp #nosql

24	RT @ITredux: It's Official: The Internet Of Things Takes Over Big Data As The Most Hyped ... - Forbes http://t.co/ZaDQgU9FwH #bigdata
25	Marketers Speak out on Data, Email, and the Customer Experience - Direct Marketing News http://t.co/nQ1tBO8Ek6 #TCE #bigdata
26	For many, it's computational power, for some, it's data formats - what's your #bigdata bottleneck? http://t.co/wp15WUlr7v
27	The value of big data â€“ Part 1: Big board games http://t.co/leKjTva4Fl #bigdata
28	#BigData in #HR: Finding In-House Talent In The Digital Age | @scoopit http://t.co/V6vRY8kSiK
29	RT @ComptelCorp: MT @shateley: The next step in #BigData is to make information more intelligent and more human http://t.co/duMA2qtP0X #anaâ€¦
30	can your data center handle the Internet of Things? http://t.co/7U7waa2OKp via @orangebusiness #BigData #IoT #M2M
31	Eliminate Spam by using our Spam Detection #API: http://t.co/8WIxIG1mZA #machinelearning #bigdata
32	How Baidu and UNDP plan to use big data to tackle development issues http://t.co/VuojXGItFQ #BigData
33	Watch what you're wearing... Tumblr to scan photos for brands. http://t.co/GnI9zyab7n MT .@GlenGilmore .@StevenVBe .@estherbouw #BigData
34	Pragmatic Programming Techniques: Lambda Architecture Principles http://t.co/WZGfwsj68V #bigdata #architecture
35	A year of tech industry hype in a single graph: http://t.co/viy9ZTyC2I #BigData
36	RT:@xgumara#BigData is not about the 3V's; it's about rapid increase in public awareness that data is a valuable resource http://t.co/hpnBF
37	#BigData Technologists Transition to Customer-Facing Roles http://t.co/s49l9MVqHQ
38	NASA Investigating Climate Impacts of Arctic Sea Ice Loss http://t.co/xhHCHKO8g9 #bigdata #analytics
39	Bringing the human touch to #bigdata #analytics http://t.co/pKiFyX76EY via @itnewsafrica
40	Success tips for Big Data: culture of analytics, data scientists and customer focus http://t.co/Z28PeFRhRX via @YourStoryCo #bigdata
41	"Tools And Tactics For Leveraging Big Data - In The Arizona Desert" http://t.co/I3nDkKm0GM #bigdata #smartdata
42	RT:@AmalioFDâ€œ@simonlporter: It's Official: The #InternetOfThings Takes Over #BigData As The Most Hyped #Technology - http://t.co/ep7FaDCkBNÂ
43	Statisticians have boasted of the benefits of #bigdata. Now theyâ€™re discovering the weaknesses http://t.co/skUByPK70C
44	What would happen if #researchers lose all their data? Read @ http://t.co/TKRCC6USZc #DataScience #openscience #openscience #BigData
45	#IoT The Internet of Things calls for a different mindset #InternetOfThings http://t.co/93pLTpFaj1 #bigdata
46	Using Big Data To Understand Migrations http://t.co/qP8Ce1YyA5 #Dubai #Mydubai #UAE #Data #BigData #Analysis #Analytics
47	New Trend in Data Centre Management: Virtualization http://t.co/aV7I5usQ39 #datacentre, #datacenter , #blog, #bigdata,
48	The janitors "need better tools so we can spend less time on data wrangling and get to the sexy stuffÂ #bigdata http://t.co/59oH6YvhBd

49	RT @DavidCh27992090: Posthumanism & #bigdata - aspiration is not for (thinking) computers that are more like humans but for humans to 'thinâ€¦
50	RT @VicZagorsky: Data needs a leader. The new hero of big data and analytics : Chief Data Officer http://t.co/GGkokzYMW2 #dataofficer #bigâ€¦
51	If you rely on real-time streams of performance and marketing #BigData, #network #monitoring should be a priority http://t.co/HPq66KmLRu
52	Chinese Hackers Steal 4.5 Million Patient Records http://t.co/kfgb6P3H9H #itsecurity #bigdata #hacked #pwnt #china #theft
53	RT @AccentureDigiUK: Can we harness #bigdata effectively? http://t.co/E4dyXww92R #data #analytics via @eyeforpharma http://t.co/PP9pLiKdy4
54	RT @BRIDGEi2i: How A Computer Algorithm Predicted West Africaâ€™s Ebola Outbreak Before It Was Announced http://t.co/nbs17Dzh5o #ebola #bigdaâ€¦
55	RT@askbahar: Big Data talks in #Dhaka - #BigData #Science and #Cloud Computing http://t.co/xvMd03QaIE
56	RT:@nivfJoin @PyramidAnalytic & other #BI experts as they discuss #BigData and more at #Microsoft Israel on Aug 27!... http://t.co/keiy
57	Never underestimate the power of social connectedness. #SocialMedia #DigitalMarketing #BigData
58	Top story from #BigData Organizations For Big-Data Scientists, â€˜Janitor Workâ€™ Iâ€¦ http://t.co/c89kATzFqG, see more http://t.co/xdXUg37lIi
59	RT @valerietyson: The Emerging Pitfalls Of Nowcasting With #BigData http://t.co/VWkyshXLqv >@TechReview
60	Health Care Industry talks most about Big Data http://t.co/wGJr3rYXle #bigdata http://t.co/VZQjh71qO1
61	RT @mongodbinc: Choose #NoSQL database MongoDB to make your organization faster, better, and leaner. Download the complete guide: http://t.â€¦
62	Interesting >> DataGravity says its time for your storage to smarten up already Paula Long, co-founder o #BigData http://t.co/CS8u6Qeh6t
63	Big Data is finally start turning into normal data. Gartner Emerging Technology Hype Cycle 2014 http://t.co/ShUrRQg4vB #bigdata #gartner_inc
64	Interested in a career with Couchbase? Apply today! http://t.co/wIoOhpwFrs #NoSQL #DB #Python #Java #PHP #OpenSource #BigData #TechJobs
65	Big Data in Banking and Financial Services http://t.co/zvDbpK7LId via @insideBigData #bigdata
66	RT @DellDP: A holistic perspective of the data lifecycle can help you efficiently address immediate goals. http://t.co/oRDJZ98AtM #bigdata
67	Skyscanner turns to NoSQL to deliver flight comparison #BigData - http://t.co/TuLZ2nPqVD
68	#BigData as a day in your life video. Nice share! @CBC_Digital
69	EqualLogic Co-Founder Paula Long's Latest Startup, DataGravity, Launches - Wall Street Journal (blog) http://t.co/nBrpki7yKQ #bigdata
70	@GilPress: It's Official: The #IoT Takes Over #BigData As The Most Hyped Technology, new entry: #DataScience http://t.co/t9dczQsLD2
71	TechnoVision 2014: Technology Building Blocks for Digital Transformation by @Capgemini #bigdata http://t.co/zCEHO3E2jm via @SlideShare
72	RT @Datumbox: Machine Learning explained in simple words: http://t.co/vgkAQUsDCE #machinelearning #bigdata #statistics #tutorials

73	Lol RT @HugoR Its Official: #InternetOfThings Takes Over #BigData As The Most Hyped Technology http://t.co/sBUK9xwBCg #IoT #design #business
74	RT @siliconrepublic: .@Sbootcamp is looking to take on 10 #startups for its new #IoT and #bigdata programme in Barcelona. http://t.co/nr9Udâ€¦
75	Tumblr to Scan Photos for Brands http://t.co/75S9WW8x9h RT @StevenVBe @estherbouw #smm #BigData http://t.co/qn3m6MDSOj
76	Open Frameworks Around Pentaho Providing Blueprints for Big Data Success http://t.co/AGFWqN4A21 #BigData
77	More restructuring for Blackberry: QNX, Project Ion, and crypto software combined... http://t.co/wEmlgfggbL #bigdata #API #IoT #innovation
78	#BigData #Analytics helping now with #Personalshopper experience - http://t.co/6iP0tqGNzD
79	Train Reading: Can Stocks and Bonds Both Be Right? http://t.co/ENXRhZsf5o #BigData #Commute
80	. @naftaliharris it is a continuum. Tableau, R, Python, SQL... MapReduce.. #bigdata aint as fun as #smalldata, harder to work with
81	Which PIMS (Personal Information Management Services) opportunity should we go for? / Ctrl-Shift #bigdata #privacy http://t.co/k3tTsHLHwV
82	As DBMS wars continue, PostgreSQL shows most momentum:http://t.co/NBdTsfurvC #MongoDB #NoSQL #Oracle #MySQL http://t.co/ALniTQvXL3
83	RT @BIG_DATA_News: Why big data scientists need to do the 'janitor' work: The field known as big data offers a contemp... http://t.co/SfPâ€¦
84	The Different Types of #Data Each #SocialNetwork Is Collecting To Understand Users And Target Ads http://t.co/5sNoUMafvI #analytics #bigdata
85	Getting Real About Management and "Big Data" http://t.co/UZurdqQiKL #BigData #Hadoop
86	Why big data scientists need to do the 'janitor' work: The field known as big data offers a contemp... http://t.co/SfPRAsgh2O #BigData
87	Google can now track when your online clicks lead to phone calls #google #tech #bigdata http://t.co/7PzNBehxGE
88	Business Intelligence toolkit en oplossingen http://t.co/tPMukGGSia #BigData #BI @1BigData
89	Business Intelligence, bringing the right information at the right time is a 5-step process http://t.co/u9VTbke8Ee #BI #BigData @1BigData
90	#BigData The practices and technology that close the gap between the data available and the ability to turn that data into business insight
91	On the Verge: Big Data takes 'Dangerous' risk http://t.co/BZ5X8GDr7J via @usatoday #Bigdata #datascience
92	Opensource Naive Bayes Text Classifier in #JAVA: http://t.co/WvTFkLZoQO #machinelearning #bigdata #nlproc #opensource
93	Robots helped inspire #DeepLearning and might become its killer app http://t.co/zfl5uxrEt6 #BigData #MachineLearning
94	Text Analytics vs. Other Research Methods [VIDEO] : http://t.co/pOroPTOUrF #bigdata
95	RT @Swiftstories: Today's panel on #BigData and #privacy w/@Google privacy counsel David Lieber and FTC's @MOhlhausenFTC: https://t.co/WTm9â€¦
96	Speaking at the #Martech conference in #Boston tomorrow on #BigData and today's ability to Understand your Customer with Data in Motion