Utilising Wikipedia for Text Mining Applications

Muhammad Atif Qureshi

College of Engineering and Informatics
National University of Ireland, Galway

Department of Informatics, Systems and Communications
University of Milano-Bicocca, Milan

A thesis submitted for the degree of
Doctor of Philosophy

2015

Acknowledgements

I am very grateful to my supervisors Dr. Colm O'Riordan and Dr. Gabriella Pasi. Their guidance and support made this day possible as it stands today. In particular, I have found a great friend in Dr. Colm O'Riordan besides being my supervisor, he made my stay in Ireland such a delight and his kindness showed me that I have more to learn from him other than the scientific subject of this thesis. When my mother came to Ireland after the birth of my daughter, Colm visited our place with his family twice, we felt really thankful for his kind gesture and my mother asked me to say thanks in the best possible way for his gentleness on multiple occasions, and hence, from myself and my mother, I am writing a special thanks in this acknowledgement.

I am also thankful to the several suggestions and interesting discussions throughout out my PhD program from different people in the National University of Ireland, Galway and University of Milano-Bicocca, Milan.

I am very grateful to the College of Engineering and Informatics scholarship committee within National University of Ireland, Galway who considered me worthy of the opportunity. It surely would not have been possible to complete the research conducted in this thesis without their support.

I am grateful to my wife Arjumand who partnered me in all walks of life ranging from scientific discussions to home affairs. We have seen different ups and downs of life together and always stood by each other's side; we share workplace, home affairs, and enjoy different hobbies and activities. She developed an interest in SciFi because I am a fan of it, she enjoys watching sports (a problem solved for me) in terms of TV time. Together, we have a beautiful daughter Fareeha Qureshi whose smile makes the world look so easy and pleasant. Fareeha will be 5 months old at the time of submission of this thesis but even this short period is so dear and valuable that words can't explain.

I am very grateful to my mother who has been a source of strength for me throughout my life; she raised me up after the passing of my father at an early age and she showed with her actions that nothing is impossible. I admire her for completing her PhD in the times when she was a single parent and sole earning hand for our family. It is from her that I learned that nothing is impossible to achieve if we have strong motivation, she is my inspiration and my lamp at home that turned me into a person that I am today.

Lastly, and most importantly, I am humbled by the Blessings of the Creator and Sustainer of the Universe, Allah swt. Indeed, it is He who grants us what we do not deserve and none is worthy of praise except Him.

Abstract

The process whereby inferences are made from textual data is broadly referred to as text mining. In order to ensure the quality and effectiveness of the derived inferences, several approaches have been proposed for different text mining applications. Among these applications, classifying a piece of text into pre-defined classes through the utilisation of training data falls into supervised approaches while arranging related documents or terms into clusters falls into unsupervised approaches. In both these approaches, processing is undertaken at the level of documents to make sense of text within those documents. Recent research efforts have begun exploring the role of knowledge bases in solving the various problems that arise in the domain of text mining. Of all the knowledge bases, Wikipedia on account of being one of the largest human-curated, online encyclopaedia has proven to be one of the most valuable resources in dealing with various problems in the domain of text mining. However, previous Wikipedia-based research efforts have not taken both Wikipedia categories and Wikipedia articles together as a source of information.

This thesis serves as a first step in eliminating this gap and throughout the contributions made in this thesis, we have shown the effectiveness of Wikipedia category-article structure for various text mining tasks. Wikipedia categories are organized in a taxonomical manner serving as semantic tags for Wikipedia articles and this provides a strong abstraction and expressive mode of knowledge representation. In this thesis, we explore the effectiveness of this mode of Wikipedia's expression (i.e., the category-article structure) via its application in the domains of text classification, subjectivity analysis (via a notion of "perspective" in news search), and keyword extraction.

First, we show the effectiveness of exploiting Wikipedia for two classification tasks i.e., 1- classifying the tweets[1] being relevant/irrelevant to

[1]Message sent using Twitter.

an entity or brand, 2- classifying the tweets into different topical dimensions such as tweets related with workplace, innovation, etc. To do so, we define the notion of *relatedness* between the text in tweet and the information embedded within the Wikipedia category-article structure. Then, we present an application in the area of news search by using the same notion of *relatedness* to show more information related to each search result highlighting the amount *perspective* or subjective bias in each returned result towards a certain opinion, topical drift, etc. Finally, we present a keyword extraction strategy using community detection over the Wikipedia categories to discover related keywords arranged in different communities.

The relationship between Wikipedia categories and articles is explored via a textual phrase matching framework whereby the starting point is textual phrases that match Wikipedia articles' titles/redirects. The Wikipedia articles for which a match occurs are then utilised by extraction of their associated categories, and these Wikipedia categories are used to derive various structural measures such as those relating to taxonomical depth and Wikipedia articles they contain. These measures are utilised in our proposed text classification, subjectivity analysis, and keyword extraction framework and the performance is analysed via extensive experimental evaluations. These experimental evaluations undertake comparisons with standard text mining approaches in the literature and our Wikipedia framework based on its category-article structure outperforms the standard text mining techniques.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation and Problem Statement

Among the fundamental forms of communication, a popular form is written text or textual data. Human beings have found a great comfort in expressing their viewpoint in writing because of the ability to preserve thoughts for a longer period of time than oral communication. However, textual data may contain the following complexities [2]:

- Lack of contextual and background information
- Ambiguity due to more than one possible interpretation of the meaning of text
- Focus and assertions on multiple topics

The above-mentioned problems mainly arise from the informal nature of day-to-day communications of human beings. However, to be able to automatically process textual data, there is a clear need for effective solutions to the above-mentioned issues.

1.1.1 Textual Data over the World Wide Web

Textual data is a very popular means of communication over the World Wide Web in the form of data on online news websites, social networks, emails, governmental websites, etc. Basically, nearly everything which is present on the World Wide Web has a textual presence. In particular, users over social networks generate their own content and prefer to communicate mostly through text. Textual data has the ability to reach out to a large community, and whenever textual content is read, it can generate a further discussion thereby leading to further generation of textual content.

With so much textual data around us especially on the World Wide Web, there is a motivation to understand the meaning of the data through automated methods for all sorts of computer science applications. By understanding the meaning of textual data the machine can answer different questions such as the following:

- What is the main topic and sub-topics of the written text?
- What are the keywords and entities defining the topics of the text piece?
- What is the underlying context of a certain text piece?

1.1.2 Role of Knowledge Bases in Text Mining Applications

Knowledge bases are playing an increasingly important role in solving the various problems that arise in the domain of text mining. Table 1.1 lists a few of the problems along with the knowledge base used to deal with the problem.

Problem	**Knowledge Base**
Question Answering Applications	YAGO, DBPedia, WordNet [65]
Text Classification and Categorization.	ODP [70].
Query Expansion for Information Retrieval.	ConceptNet [118]
User Profile Creation for Personalized Web Search	ODP [41], YAGO [36]
Cross-Lingual Information Retrieval	Wikipedia [1].

Table 1.1: Text Mining Problems Solved using Knowledge Bases

Of all the knowledge bases, Wikipedia has, so far, proven to be one of the most valuable resources; in fact knowledge bases such as DBPedia [23] and YAGO [195] have been derived from Wikipedia. It is basically an online, collaboratively generated encyclopaedia and one of the largest and most consulted reference works in existence. Wikipedia is written with the goal of human consumption but it contains a certain structure which can be exploited by automated algorithms. This structure is composed of hierarchical categories, and these categories act as semantic tags to different Wikipedia articles. Moreover, each article interlinks with each other using the anchor text within the content. Recent years have seen many significant research questions

being solved with the help of Wikipedia [64, 71, 141, 217] and it has been successfully applied to complement the understanding of different datasets [2].

1.2 Open Challenges

Despite the application of Wikipedia to several text mining problems, there remain a number of open challenges. We list a few of these challenges:

- As mentioned in section 1.1.2, Wikipedia is composed of category hierarchies with the categories linked to Wikipedia articles. To the best of our knowledge, previous research efforts that utilise Wikipedia for knowledge extraction tasks have not taken both Wikipedia categories and Wikipedia articles together as a source of information [64, 141].

- Information access tasks that can benefit immensely from Wikipedia include exploratory search, query expansion, and document clustering to name a few. The various hierarchical categorizations in Wikipedia can aid the identification of various topical threads in a document thereby improving their retrieval ability. However, to the best of our knowledge only a few works [101, 103, 216] have utilised Wikipedia to make such inferences.

- There are several occasions when textual data lacks context and more so in the age of social media. This brings a whole set of new challenges to traditional fundamental research topics in text mining, such as text clustering, text classification, information extraction, and sentiment analysis; unlike standard textual data which has several sentences and hence, a surrounding context whereas social media messages consist of few phrases or sentences. These messages lack sufficient context information for effective similarity measures [166], the basis of many text processing methods [100]. In such a scenario, external knowledge bases such as Wikipedia can help alleviate the semantic gap in textual data (i.e., lack of context problem).

1.3 Research Questions

In this thesis, we present strategies to understand the meaning of text in a document and across the entire document collection by utilising encyclopaedic knowledge in Wikipedia. Wikipedia is an up-to-date, dynamic resource with extensive knowledge on various topics such as politics, sports, science, business, movies, music etc. By

exploiting Wikipedia we can identity the meaning of words/phrases (concepts) mentioned inside textual document using definitions from Wikipedia. Therefore, our core research question is as follows:

How can we effectively utilise the concepts and their interconnections within knowledge bases such as Wikipedia to improve effectiveness of various text mining applications?

The above core research question can be transformed into a specific research question for the utilisation of Wikipedia as follows:

- **How can we effectively use the structure and relationship between Wikipedia categories and articles?**
  Wikipedia categories act as semantic tags for Wikipedia articles and there is a lot of untapped, latent knowledge in this structure. This thesis pursues an exploration into this knowledge in an attempt to make inferences for textual data by making use of both community detection[1] [168, 169] and machine learning approaches [173, 171]. In order to address the above question, we investigate the following sub-questions:

  R1. ***How can Wikipedia be used for the identification of effective keywords that summarize the text collection?***
  Text summarization continues to occupy a central place in text mining on account of its ability to succinctly represent textual documents, and keyword extraction is one fundamental sub-task within text summarization. Wikipedia with its wealth of "knowledge" serves to provide diverse information in the form of Wikipedia articles' hyperlinks, Wikipedia categories, Wikipedia category-article associations etc. Utilisation of this diverse information can lead to extraction of accurate keywords that summarize a given collection of documents.

  R2. ***How can Wikipedia be used for enhanced context representation within an informal text piece?***
  In the age of social media, textual data generated over these media lacks structure and context due to being written in an inherently informal manner[2]. In order to make sense of social media data to be able to derive

[1]The community detection algorithm was applied over the graph of Wikipedia categories and articles.

[2]Facebook status messages and tweets provide sufficient evidence for such phenomenon.

meaningful inferences from it, we consider contextualization of such data as an essential step and Wikipedia on account of its semantic richness enables this as evidenced in previous works [73, 102, 141].

R3. ***How can we identify various topical assertions (both implicit and explicit) in a piece of text?***
We live in the age of controversy with news reporting representing various agenda [33], and most often media outlets exhibit ideological viewpoints in an implicit manner through various topical drifts. Wikipedia categories and articles on account of their huge coverage can aid in the identification of such topical drifts within online news pieces thereby leading to a greater awareness on the part of users.

1.4 Contributions

The general architecture of this thesis can be summarized as in Figure 1.1. As can be seen in the figure, unstructured or semi-structured textual data is extracted from the data sources which is then analysed using the knowledge base (Wikipedia) for different application domains.

In this thesis we make the following contributions to show the effectiveness of using Wikipedia for text mining applications:

A1. We address question 1 through proposing a solution to the problem of keyword extraction from a collection of academic documents of short-text (titles of Web pages). The Wikipedia category taxonomy together with its semantic annotation of Wikipedia articles is passed through a community detection framework to produce domain-specific keywords and these keywords constitute an effective summary for a document collection.

A2. We address question 2 through tackling the task of reputation management of companies over Twitter by filtering tweets related to an entity while also specifying tweets related to different aspects of a company's interest (such as innovation, leadership). Wikipedia is used to provide contextualization for the tweets and a "semantic relatedness" measure based on Wikipedia category-article structure is utilised for the task.

A3. We explore question 3 in the context of the news search domain where agendas by media organizations are driving the published content [33]; we developed a

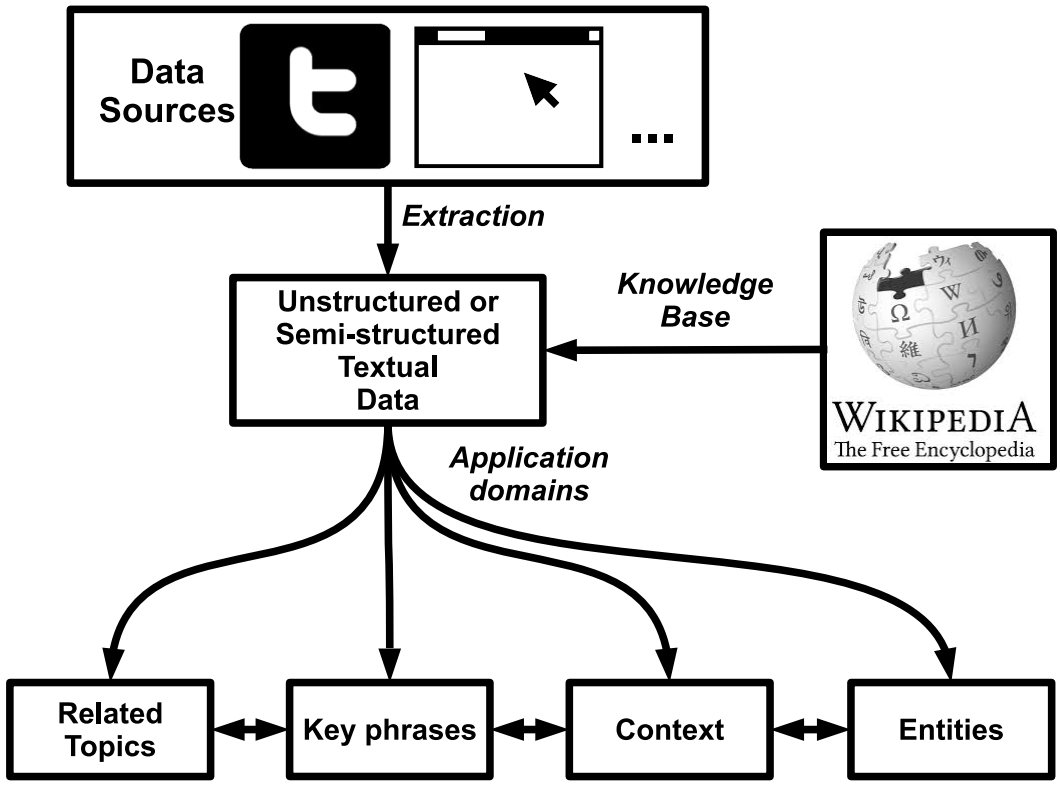

Figure 1.1: The General Architecture of the Thesis

search-engine interface for news articles which shows the amount of perspectives present inside each news result returned by popular search engines [172]. Again, the "semantic relatedness" measure based on Wikipedia category-article structure is utilised.

1.5 Thesis Flow and Structure

Figure 1.2 shows the overall pathway of this thesis. As can be seen, text streams from documents in combination with Wikipedia (specifically Wikipedia article titles and redirects) are used in the phrase chunking step. The output from the phrase chunking step comprises a set of candidate phrases which are utilised in combination with Wikipedia category-article structure) for 1) calculation of relatedness, and 2) detection of communities. The relatedness scores and detected communities are used in three separate applications in the context of this thesis. These applications are *online reputation management* (Chapter 5), *perspective-aware search* (Chapter 6), and keyword extraction (Chapter 7).

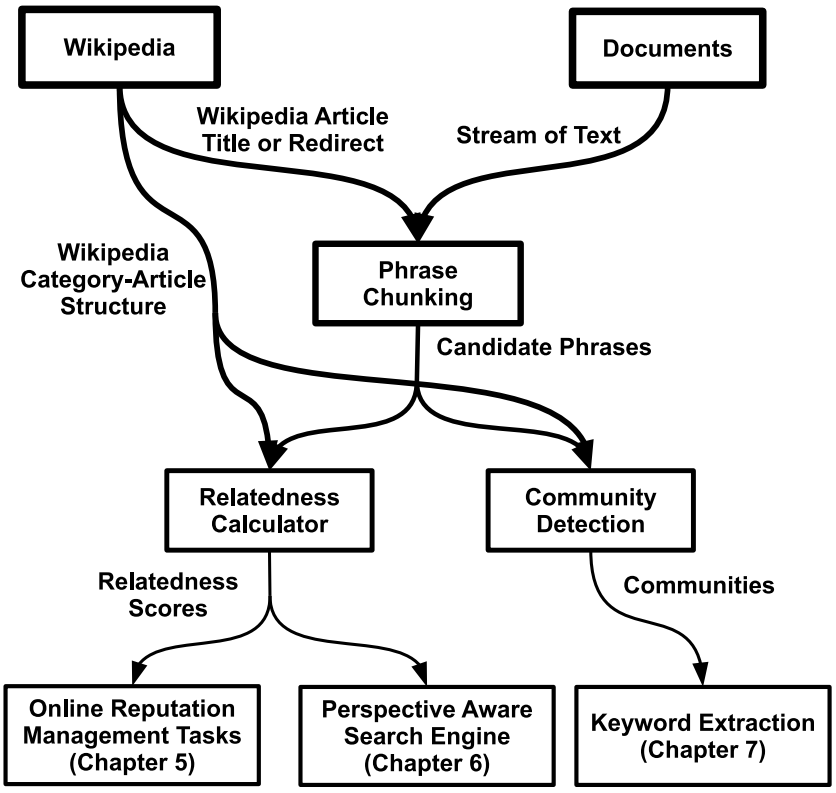

Figure 1.2: Thesis Pathway

This dissertation is structured as follows. In this chapter, we introduced the motivations along with a presentation of the research questions and contributions. Chapter 2 presents some background material to provide a description of text mining tasks relevant to our work in addition to a description of Wikipedia and other knowledge bases. Chapter 3 presents a description of the state of the art related to the core research areas of the thesis. It introduces the most relevant definitions and the related work for the research fields of semantic relatedness, named entity recognition, word sense disambiguation, named entity disambiguation, novel search engine interfaces document summarization, and keyword extraction. Chapters from 4 to 7 are the core chapters of the thesis and include the main contributions.

Chapter 4 presents our framework for measures of "semantic relatedness" built on top of the Wikipedia category graph and Wikipedia category-article structure. We present a detailed explanation on how we generate features through exploitation of the Wikipedia category-article structure; these features are essentially based on relatedness scores that we generate through the use of category hierarchies. Chapter 5 and 6 present two application scenarios where we apply our newly proposed "semantic

relatedness" model.

In particular, chapter 5 presents details of two application scenarios where we describe our participation in the CLEF RepLab 2013 filtering task and CLEF RepLab 2014 reputation dimensions classification task[3] We have exploited multiple Wikipedia category taxonomies to derive separate relatedness scores to describe an entity and the core of our approach centers on the relatedness model explained in chapter 4.

Chapter 6 presents details of a "perspective-aware approach to search" where topical assertions are identified within news search results returned by different search engines. Perspective-aware search is proposed as a means to investigate topical drifts in documents which in some cases can be used to analyse a leaning towards an agenda. We also explain the usefulness of Wikipedia's semantic relatedness model (explained in chapter 4) in identification of various topical drifts in the context of news search results.

Chapter 7 describes our approach for identification of domain-specific keywords from a collection of short-text. We present a novel domain-specific keyword extraction method, which relies on both the notion of n-gram overlap between the titles of Wikipedia articles[4] and those of the short-text collection (titles of Web pages), and on a community detection algorithm that makes use of the Wikipedia Category graph in order to boost the extraction of domain-specific keywords. The output of the proposed method is a set of meaningful keywords (n-grams) that define the topical domains of the considered collection.

Chapter 8 concludes this thesis with a discussion on findings and an outline of future work.

[3]These tasks were organized as a CLEF evaluation task [8, 9] where teams were given a set of entities and for each entity a set of tweets were provided. In RepLab 2013 the challenge was to classify tweets as relevant or irrelevant with respect to the entity, and in RepLab 2014, the challenge was to classify tweets with respect to the reputation dimension of an entity.

[4]And the redirects of the Wikipedia articles.

Chapter 2

Background

In order to cover the related background, we start by giving a review of different text mining models that are related to our proposed research work. The essential component of any text mining process is conversion of input data from its raw format to a structured, easy-to-manipulate format, a document representation, and we begin by presenting an overview of the "vector space model". This is followed by an overview of supervised and unsupervised methods of making inferences from textual data. We then briefly present various types of knowledge bases along with a detailed background on Wikipedia which is the knowledge base upon which the contribution of this thesis rests. We also motivates our choice of Wikipedia for the text mining applications carried out in this thesis. Finally, we present an overview of the microblogging platform Twitter which represents one of the application scenarios to which we apply our semantic relatedness model.

2.1 Text Mining

The term "text mining" was first coined in by Feldman and Dagan in 1995 [63]. It is the process by which textual data is analysed in order to derive high quality information on the basis of patterns. In the context of text mining, there are two popular classes of techniques namely unsupervised learning and supervised learning. We present a brief overview of each in the following subsections. The last subsection covers evaluation measures used to measure the performance of various tasks.

2.1.1 Document Representation Models

The selection of a document representation model for text depends on the selection of meaningful text units. The most commonly chosen "text units" are called terms

and hence, a document is represented as a set of terms. Many different document representation methods exist generating different types of term sets and the choice of term set has a huge effect on the quality of the overall text mining process. The most basic and earliest approach for document representation is the Bag-Of-Words model (BOW), which views the basic units as a single word thereby assuming no significance for grammar or word order in the document. Another approach that builds on BOW's basic idea is the popular and commonly used Vector Space Model (VSM) [187].

2.1.1.1 Vector Space Model

This model allows for partial matching between documents through measuring the degree of similarity between those documents, and currently this model is widely used in information retrieval and text mining tasks. In the vector space model, modelling of a document is performed with a vector with each dimension of the vector corresponding to a separate term, and instead of assigning each term a Boolean value[1] to represent whether it exists in a document or not, each term is assigned a certain weight. The weight is used to denote the contribution of the term to the 'meaning' of the document. Moreover, all the algebraic rules and operations for vectors can be applied to the documents. Therefore, if we have a set of m documents $\{d_i : i = 1, \ldots, m\}$, each document d_i is represented as:

$$\vec{d_i} = (w_{i1}, w_{i2}, ..., w_{in})$$

where n is the total number of terms. The original purpose behind the vector space model was to have a model that could enable the measurement of similarity between two documents: this is possible through measuring the closeness between the vectors representing these documents; the cosine of the angle between the vectors is utilised for this purpose[2]; the similarity S_{ij} between a document d_i and a document d_j can be defined as follows:

$$\vec{S_{ij}} = \frac{\vec{d_i}.\vec{d_j}}{|\vec{d_i}| \times |\vec{d_j}|}$$
$$= \frac{\sum_{k=1}^{n} \vec{w_{ik}} \times \vec{w_{jk}}}{\sqrt{\sum_{k=1}^{n} w_{ik}^2} + \sqrt{\sum_{k=1}^{n} w_{jk}^2}}$$

[1] Standard Boolean model.

[2] Note that cosine of the angle between the vectors is a normalized measure and is not affected by document length.

where $\vec{d_i}.\vec{d_j}$ is the dot product between the document vectors and $|\vec{d_i}|$,$|\vec{d_j}|$ are the norms of the document vectors.

The most common method used to reflect the weight of terms in a document is the use of the *tf-idf* measure, where *tf* represents the *term frequency* of a term in a document and the *idf* represents an *inverse document frequency* of a term in the document collection. There are several ways to calculate the *term frequency*, however, a popular variant of the *term frequency* is defined as:

$$tf_{ij} = \frac{n_{ij}}{\sum_{k=1}^{n_j} n_{kj}}$$

where n_{ij} is the number of occurrences of term t_i in document d_j and n_j is the total number of terms in document d_j. The *term frequency* measure is considered to be a local measure representing how important (discriminative) the term t_i is to the document d_j rather than the other terms in this document.

The other component of *tf-idf* i.e., *idf* is considered to be a global measure representing how discriminative the term t_i is to the document d_j rather than to other documents; similar to the *term frequency* there are also different ways of calculating the *idf*, however, a popular variant of the *idf* is defined as:

$$idf_i = \log\left(\frac{m}{m_i}\right)$$

where m is the total number of documents, and m_i is the number of documents that contain the term t_i. The combination of *tf* and *idf* is used for term weighting in vector space model; it is defined as

$$\textit{tf-idf} = tf \times idf$$

There are other ways to calculate term weighting inspired by the *tf-idf* method such as the *BM25*[179] and the pivoted document length normalization [160, 190].

2.1.2 Unsupervised Learning Methods from Text Data

Unsupervised learning is the name given to the process of finding latent structure in unlabelled data; no supervision implies that there is no human expert who has assigned text documents to classes. The two main unsupervised learning methods commonly used in the context of textual data are clustering and topic modelling. The essential difference lies in whether the membership of a document lies in one cluster (referred to as *hard clustering*), or in several clusters (referred to as *soft clustering*).

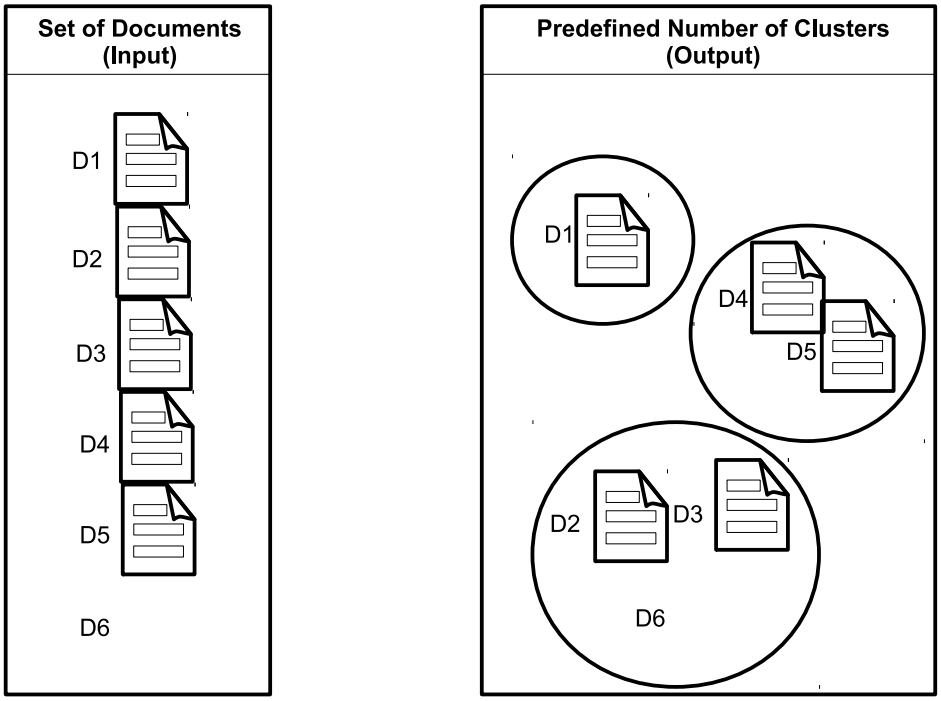

Figure 2.1: Illustration of Text Clustering Process (Hard Clustering)

2.1.2.1 Text Clustering

Text clustering refers to the process of segmenting textual documents into partitions with similar characteristics. The similarity between documents is measured through a similarity function which is basically a distance measure and the cosine distance is the most commonly used measure. Figure 2.1 illustrates the document clustering process. Text clustering algorithms are divided into a wide variety of different types such as hierarchical clustering algorithms [215], and partitioning algorithms [108].

Hierarchical clustering algorithms are a class of clustering algorithms in which a tree-like structure emerges from the hierarchy of created clusters; this tree-like

structure is referred to as a Dendrogram. The root of the tree represents one cluster which contains all the data points (documents). The number of the leaves of the tree is equal to the total number of data points (documents) and each leaf represents a cluster which corresponds to a single data point (document). Partitional clustering algorithms are a class of algorithms which start by potential partitions or clusters of data points, then update these clusters iteratively using some objective function; the objective function is generally representative of the distance between a data point and the cluster's center. The most well-known algorithm of this type is the k-means clustering.

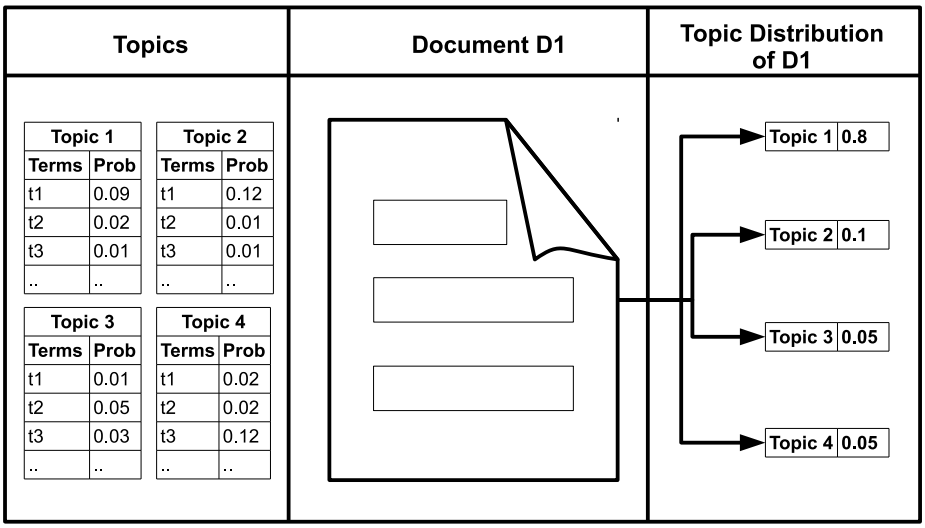

Figure 2.2: Illustration of Topic Modelling Process

2.1.2.2 Topic Modelling

Topic modelling is a popular method which performs document clustering through a probabilistic generative model. The corpus is represented as a function of hidden random variables with the parameters estimated using a particular document collection. The basic ideas behind topic modelling are as follows:

- The n documents in the corpus are assumed to have a probability of belonging to mainly one or more of k topics. A given document generally has a probability of belonging to multiple topics and hence, containing information about multiple subjects (see Figure 2.2). For a given document D_i, and a set

of topics $T_1 \ldots T_k$, the probability that the document D_i belongs to the topic T_j is given by $\mathrm{P}(\mathrm{T}_j \mid \mathrm{D}_i)$; note that the topics are essentially clusters.

- Each topic is associated with a probability vector quantifying the probabilities of the different terms in the lexicon for that topic. For a given topic T_j, and the set of terms $t_1 \ldots t_d$ with d terms in the lexicon, the probability that the term t_l occurs in topic T_j is given by $\mathrm{P}(\mathrm{t}_l \mid \mathrm{T}_j)$.

The values of $\mathrm{P}(\mathrm{T}_j \mid \mathrm{D}_i)$ and $\mathrm{P}(\mathrm{t}_l \mid \mathrm{T}_j)$ are the outputs of the topic modelling algorithm. The two well-known variants for estimation of probabilities are *Probabilistic Latent Semantic Indexing* [99] and *Latent Dirichlet Allocation* [24] with *Latent Dirichlet Allocation* being the more popular choice.

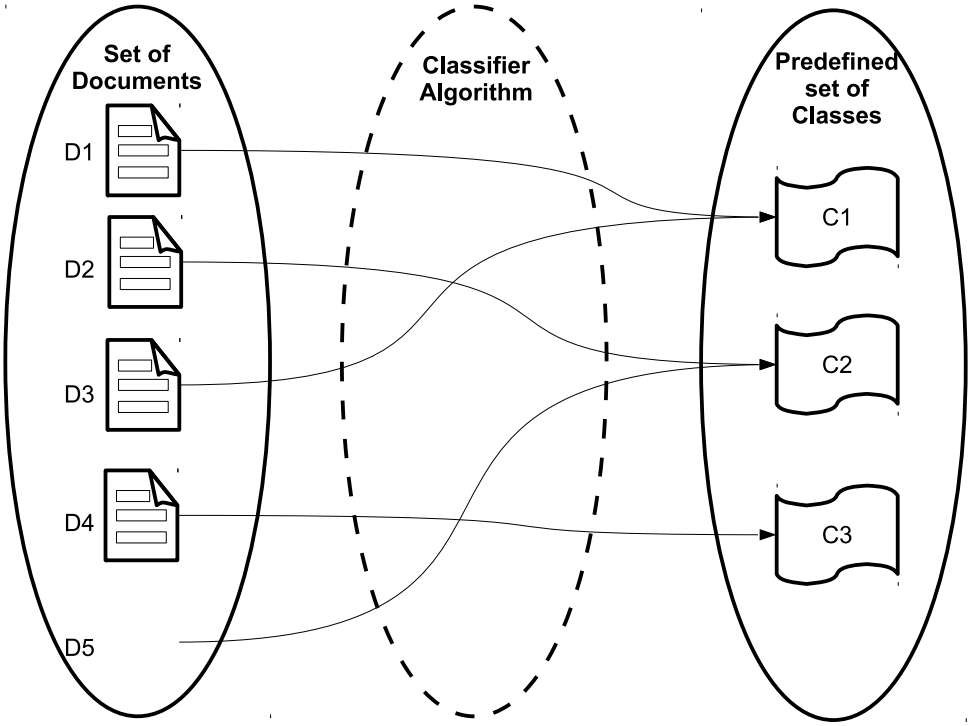

Figure 2.3: Illustration of Text Classification Process

2.1.3 Supervised Learning Methods from Text Data

Supervised learning methods are a category of methods that exploit training data (i.e., pairs of input data points with a label for the corresponding output point). These methods learn a classifier or regression function that can be used to compute predictions on new, unseen data. Generally, supervised learning methods for text data

fall under the domain of text classification: Figure 2.3 shows an illustration of the text classification process. Some key methods commonly used for text classification are decision trees, rule-based classifiers, linear classifiers, neural network classifiers and Bayesian classifiers.

2.1.4 Evaluation Measures

In order to test the performance of text mining algorithms evaluation measures are needed that are concerned with measuring how the predicted data partitions (either clusters or classes) are equivalent to ground-truth partitions which are defined by human annotators. In order to define evaluation measures, we first explain the concepts of *True Positives*, *True Negatives*, *False Positives*, and *False Negatives*. True positives reflect the instances when the actual labelled value of a document and the predicted labelled value both refer to a positive label (i.e., top column on the left in Table 2.1), and false positives reflect the instances when the actual labelled value of a document refers to a negative label and the predicted labelled value refers to a positive one (i.e., bottom column on the left in Table 2.1). True negatives reflect the instances when the actual labelled value of a document and the predicted labelled value both refer to a negative label (i.e. bottom column on the right in Table 2.1), and false negatives reflect the instances when the actual labelled value of a document refers to a positive label and the predicted labelled value refers to a negative one (i.e., top column on the right in Table 2.1).

Based on the above quantities, standard measures of precision, recall, and F-measure are commonly used as well-established measures in the literature. The following equations show these measures, where, TP is True Positives, TN is True Negatives, FP is False Positives, and FN is False Negatives.

$$Precision = \frac{TP}{TP + FP} \tag{2.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.2}$$

$$F\text{-}Measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.3}$$

Evaluations often involve using the metric of precision at k (P@k), average precision (AP), mean average precision (MAP), reciprocal rank (RR), and mean reciprocal rank (MRR). P@k is defined as the ratio of correctly matching results over the first

Table 2.1: True Positives, False Positives, True Negatives, and False Negatives Illustration

		Prediction Outcome		
		p	**n**	**total**
Actual Value	**p′**	True Positive	False Negative	P′
	n′	False Positive	True Negative	N′
	total	P	N	

top-k results. This measure is generally used when the list of returned documents/results is huge, i.e. when relevance judgements for the entire result set cannot be assessed by manual annotators (such as returned result set in Web search engines [3]). AP is the average value of P@k values computed after each correct answer in the result set, while MRR is the mean of AP across different result sets [3]. RR is the multiplicative inverse of the rank of the first correct result whereas MRR is the mean of RR across different result sets [208]. Both AP and RR capture the importance of the ranked results i.e., the value of the measure is higher when the correct result is found in the top order compared to that when the correct result is found in the lower order. Furthermore, when there is only one correct result in the result set AP and RR returns the same value.

$$P@k = \frac{\mid CorrectResults \mid}{\mid Top\text{-}kResults \mid} \tag{2.4}$$

$$AP(q) = \frac{1}{\mid CorrectResults \mid} \sum_{k=1}^{|Results|} P@k \times corr(k) \tag{2.5}$$

where q is the query across which measure is calculated and *corr(k)* returns one if the item at rank k is correct otherwise zero.

$$MAP = \frac{1}{\mid Q \mid} \sum_{q=1}^{|Q|} AP(q) \tag{2.6}$$

where Q is the set of queries across which measurement is made.

$$RR(q) = \frac{1}{rank_{firstcorrect}} \tag{2.7}$$

$$MRR = \frac{1}{\mid Q \mid} \sum_{q=1}^{Q} RR(q) \tag{2.8}$$

Other measures have been proposed in the literature; for example, the official measure of the CLEF 2013 RepLab filtering task for evaluation purposes. These are reliability and sensitivity described in detail by Amigo et al. [11]. The property that makes them particularly suitable for the filtering problem is that they are strict with respect to standard measures, i.e., a high value according to reliability and sensitivity implies a high value in all standard measures. Within binary classification such as in the case of the filtering problem, reliability is the product of precision in both classes (i.e., true positives and true negatives) and sensitivity is the product of recall of both classes. The combined effect of reliability and sensitivity is reported through harmonic mean similar to the standard measures.

$$Reliability = \frac{TP}{TP + FP} \times \frac{TN}{TN + FN} \tag{2.9}$$

$$Sensitivity = \frac{TP}{TP + FN} \times \frac{TN}{TN + FP} \tag{2.10}$$

$$F_1(R, S) = 2 \times \frac{Reliability \times Sensitivity}{Reliability + Sensitivity} \tag{2.11}$$

2.2 Knowledge Bases

We first provide a brief overview of existing knowledge bases followed by a detailed background on Wikipedia. We also explain the motivations behind the use of Wikipedia for the text mining tasks conducted in this thesis.

2.2.1 DBPedia

DBpedia is a knowledge base which extracts various types of structured information from Wikipedia [13]. It is one of the leading projects that defines semantics of data, thus allowing sophisticated queries using RDF triples about relationships and properties associated with available resources. As of 2014, the English version of DBpedia

consists of 4.58 million "things" with 583 million "facts"[3], such as including over 1.4 million persons, and over 0.7 million places, etc.

2.2.2 YAGO: Yet Another Great Ontology

YAGO is a knowledge base which is extracted from Wikipedia, Wordnet, and GeoNames[4] [98]. YAGO is very similar to DBpedia and it also represents knowledge via triples but YAGO enriches leaf categories from Wikipedia by reusing WordNet whereas DBpedia has its own taxonomy which is manually defined. YAGO was found to be above 95% in accuracy when manually evaluated over a sample of facts[5]. Currently[6], YAGO contains over 10 million entities and over 120 million "facts"[7]

2.2.3 Freebase

Freebase is a large knowledge base with broad coverage as it is extracted from various sources [25] such as Wikipedia, open library project[8], Food and Drug Administration, individual user-submitted wiki contributions. However, it is similar to DBpedia except for some differences. It allows users to edit content whereas DBpedia can only be modified through modifications in Wikipedia. It has a different structural organization than DBpedia and it consumes a variety of data sources while DBpedia relies mainly on Wikipedia for its extraction of knowledge. Currently[9], Freebase contains over 2.9 billion "facts" and over 47 million topics.

2.2.4 WordNet

WordNet is a lexical semantic database for English [146] that arranges nouns, verbs, adjectives and adverbs into synsets (i.e., sets of synonyms). These synsets are interlinked with each other by conceptual-semantic and lexical relations (antonymy, hyperonymy, hyponymy). WordNet is similar to a thesaurus; however, there are

[3]http://wiki.dbpedia.org/services-resources/datasets/datasets2014

[4]http://www.geonames.org/

[5]http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/statistics/

[6]In the month of June 2015.

[7]https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/

[8]For books.

[9]In the month of June 2015.

some differences: first, strings of letters are also considered as word forms in WordNet and second, words are also labelled in the semantic relations by WordNet. Latest version of WordNet[10] contains 155,287 unique strings and 117,659 synsets.

2.2.5 Cyc and OpenCyc

Cyc is a proprietary, commercial resource which is a 31 years old[11] artificial intelligence (AI) project [124]. It aimed to provide common sense knowledge in the hope to advance possible human-like reasoning for computer applications. OpenCyc[12] is the open source version of Cyc released in 2008. Recent release of opencyc[13] contains 239,000 concepts and 2,093,000 "facts".

2.2.6 Wikipedia

Wikipedia is a multilingual[14], collaboratively constructed largest free encyclopedia containing containing over 4.4 million articles[15] in English alone. Wikipedia contains articles on a wide range of topics, politics to science, news events to contributions by different people. Research have shown that Wikipedia is reasonably accurate[16] [43] and as accurate as its rival commercial alternates i.e., Encyclopedia Britannica [75] and Encarta [181].

A key difference between various knowledge bases lies in their underlying processing mechanism in terms of how they are read i.e., there exist human-readable and machine-readable knowledge bases. Wikipedia is different from other knowledge bases in terms of being human-readable. We utilise Wikipedia on account of its rich category graph structure; and in order to enable exploitation of the Wikipedia information we develop our own system called WikiMadeEasy. This system exploits Wikipedia dumps in an efficient way (see Appendix A).

Our main motivations behind use of Wikipedia are as follows[17]:

- Wikipedia is a collaboratively constructed resource which is updated extensively and hence, contains fresh knowledge on most topics.

[10] Version 3.0.
[11] Developed by CycCorp since 1984.
[12] http://opencyc.org/
[13] Version 4.0.
[14] Available in 270+ languages.
[15] http://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia
[16] http://en.wikipedia.org/wiki/Reliability_of_Wikipedia
[17] Note that we have essentially utilised the dumps made available by DBPedia. However, despite the fact that DBPedia contains a notable work of semantic annotations, we are not using this additional information.

- The continuous growth over a period of years makes it likely to stay useful over a number of years to come.
- The nature of continuous expansion of Wikipedia has made it truly the de-facto online encyclopedia which is more likely to cover aspects of human knowledge which are uncovered as of now but likely to be covered in future.
- Other knowledge bases chiefly rely on Wikipedia as potential source of knowledge while other sources are only included when Wikipedia lacks to cover them but this gap is more likely to diminish over the passing of time.

Each Wikipedia article contains content that defines a particular concept textually which may be accompanied with with images related to the concept inside a Wikipedia page. Each article has a title that identifies a concept and each article can also be identified with zero or many redirect strings e.g., an article with title 'United States' can be identified by either its title or redirects such as 'USA' or 'US'. Furthermore, there is a possibility of ambiguity among different article titles, e.g., apple can either be a fruit or a company and likewise more than one person can have same names such as 'Michael Jordan' which can refer to the basketball star in NBA or to the Professor at the University of California, Berkeley. To handle such ambiguous needs, Wikipedia has special pages which are called disambiguation pages. The disambiguation pages are special Wikipedia pages that contain one to many relations for ambiguous strings, e.g., the disambiguation page for 'apple' contains references to possible senses such as 'Apple (fruit)', 'Apple Inc. (company)', 'The Apple (1980 film)', etc. The Wikipedia articles are densely inter-connected to each other and each Wikipedia article references on average 22 other articles [149]. Furthermore, each article is mentioned inside different Wikipedia categories and each Wikipedia category generally contains parent and children categories.

Wikipedia categories are organized into a taxonomy structure (see Figure 2.4). Each Wikipedia category can have an arbitrary number of subcategories as well as being mentioned inside an arbitrary number of supercategories (e.g., category C_4 in Figure 2.4 is a subcategory of C_2 and C_3, and a supercategory of C_5, C_6 and C_7.) Furthermore, in Wikipedia each article can belong to an arbitrary number of categories. As an example, in Figure 2.4, article A_1 belongs to categories C_1 and C_9, article A_2 belongs to categories C_3 and C_{10}, while article A_3 belongs to categories C_3 and C_4. In addition to links between Wikipedia categories and Wikipedia articles, there are also links between Wikipedia articles as the dotted lines in Figure 2.4 show

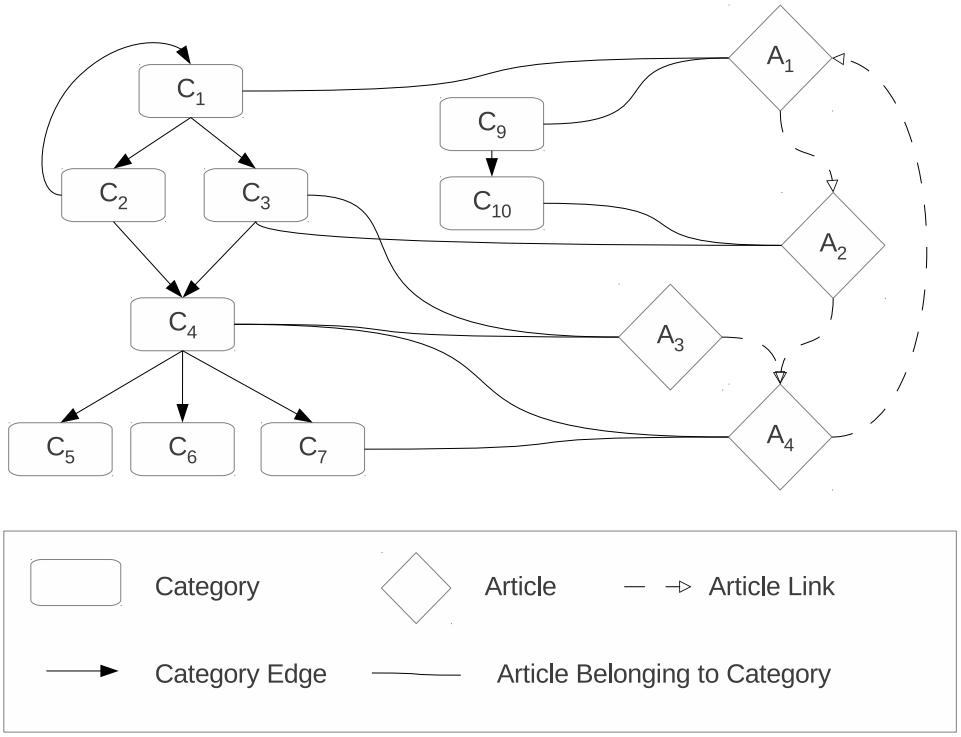

Figure 2.4: Wikipedia Category Graph Structure along with Wikipedia Articles

(e.g., article A_1 outlinks to A_2 and has an inlink from A_4). The Wikipedia categories serve as a semantic tag for the articles to which they link [227]. Similarly, the inlinks and outlinks between Wikipedia articles are organized according to the semantics inside the articles' content (e.g., the article on "Apple Inc." has an inlink from the article on "Steve Jobs" while having an outlink the to article on "iPhone").

Fig. 2.5 shows the truncated Wikipedia category graph corresponding to distinct topical interests of "sociology" and "information science" as leaf categories. From the figure it is evident that different categories narrow down to different range of topics which can be captured for the emergence of topical domain of a collection (as shown in later chapter).

2.3 The Data Source: Twitter

Twitter is a unique social media platform in that it is essentially a microblogging platform and also provides the features of a social networking web service [109, 121]. The micro-blogging nature of Twitter comes from the fact that the user can post a

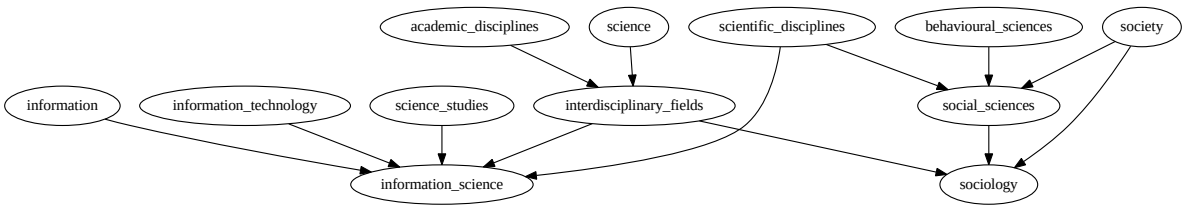

Figure 2.5: Truncated Wikipedia Category Graph

message from his/her profile while its social aspect comes from the fact that a user can follow other users.

Users turn to Twitter for several different reasons among which popular ones include the following:

- Twitter has often surpassed traditional media in reporting breaking news [12, 225].
- Twitter is an effective platform to share thoughts on daily happenings in life.
- Twitter allows easy dissemination of opinionated voices over different sorts of issues such as politics [170], companies, musicians, and other entities [192].
- Twitter is an online medium that supports users in having conversations with friends as was traditionally done with SMS service.

Below we present an overview of the features of Twitter to help the user understand usage of various Twitter features in the application scenarios of subsequent chapters:

- **Following**: A *following* relationship is a directed relationship i.e., user A follows user B while vice-versa is not true until user B decides to follow user A. When user A becomes *follower* of user B it means that user A will see all public tweets of user B.
- ***Tweet:*** The 140-character message that users can post on the web site is called a "tweet". When a user posts a tweet, it is displayed on the user's timeline while also viewable by all of his/her followers. Moreover, the tweet may also be viewed by any other Twitter user searching for keywords matching some content in the tweet.
- **User mentions**: A user can engage in conversation with other users by using '@' followed by the user identifier (name) in the tweets, and the process is called *user mention*. E.g., `Guys you should follow this news @user1 @user2`.

- **User reply**: When a user want to specifically communicate or reply to another user, the user places a mention at the beginning of a tweet. E.g., *@user1 lets celebrate the achievement*.

- **Retweet**: A user can re-produce or share a tweet from another user on his/her profile by using 'RT' and '@username:' before the message and the process is called *re-tweeting*. *Retweet* is simply a copy of a tweet borrowed from some other user because it might be interesting to share. E.g., *RT @user1: it is interesting article http://link*. Similarly sometimes, user tends to modify the content of original by putting 'MT' instead of 'RT'.

- **Hashtags** There is a special feature that makes a tweet a good candidate for retrieval when users search for a particular topic; the user has to put hash symbol '#' before a keyword and these keywords are then referred to as *hashtags* (e.g., #java). This tag is similar to tags on social tagging systems.

2.4 Summary of the Chapter

In this chapter we presented an overview of classical text mining. In particular we first presented the most basic document representation model namely vector space model which treats each term as a vector dimension with an associated weight. We then presented a summary of basic text mining applications that involve unsupervised and supervised methods of learning from textual data. Unsupervised learning methods are those that do not involve any human judgements in the form of assigned classes; instead it clusters documents through the latent structure within the documents. Supervised learning methods on the other hand involve training over pre-assigned classes to discover some structure based on which assignments to unseen textual data can be made. We also presented principal evaluation metrics commonly used for text mining applications; note that these evaluation metrics will be used to present results of our experimental evaluations in later chapters. We presented a brief description of available knowledge bases while motivating our reasons for use of Wikipedia for the contributions of this thesis. This was followed by a detailed overview of the Wikipedia category-article structure and the interconnected structure between them which forms a significant component of the contributions in this thesis. Finally, an overview of various features of Twitter was presented which constitutes our data source for few application scenarios in this thesis.

Chapter 3

Related Research

This chapter presents an overview of the state-of-the-art in text mining applications that are related to the contributions of this thesis. In particular, we present details of research that aims to associate textual data with their semantics and therefore, we begin by presenting in detail the notion of "semantic relatedness". This is followed by covering some research works in the domain of named entity recognition, and disambiguation tasks i.e., word sense disambiguation and named entity disambiguation. We then present an overview of research in the domain of information retrieval with a particular focus on approaches to deal with complex information seeking. We then present a summary of text mining works aimed towards knowledge extraction from text, particularly focusing on document summarization and keyword extraction.

3.1 Semantic Relatedness

The literature has defined semantic relatedness as a means to allow computers to reason about written text [217] whereby the reasoning deals with finding and quantifying the strength of semantic association between textual units [89]. Within the proposed works in the literature the difference lies in the knowledge base employed, the technique used for measurement of semantic distances and the application domain [97, 107, 122, 162]

Within the context of this thesis, we follow the notion of semantic relatedness adopted by Milne and Witten [217] whereby we use it for measuring degree of similarity, and the relationship between different terms. Two examples from Milne and Witten are with respect to relationship between "social networks" and "privacy", and "cars" and "global warming". To clarify further, 'lion' and 'cheetah' are not same but are similar due to belonging to the same biological family i.e., Felidae; likewise word pairs carpenter:wood and mason:stone are relationally similar because both carpenter

and mason are professions while wood and stone represent materials used to carry out the job. Note that we utilise Milne and Witten's definition of semantic relatedness; however, we differ with them in terms of strategy employed since they utilise Wikipedia hyperlinks which in our context fails to show good performance[1].

For the contributions in this thesis, the semantic relatedness framework introduced in Chapter 4 is a core component, and here we present some semantic relatedness frameworks proposed in the literature.

To estimate semantic relatedness Lee [123] and Dagan et al [50] used co-occurrence; Budanitsky and Hirst [34] and Jarmasz [110] used generalization ('is a') relations between words using WordNet based techniques; Jarmasz [110] also used Rogets Thesaurus [180] and showed improvement over a WordNet-based technique; Turney [204, 206] used Latent Relational Analysis for relational similarity (i.e., for word pairs), Sahami and Heilman [183] and Bollegala et al. [26] proposed querying Web Search Engines for measuring similarity of short-text snippets; Metzler et al. [143] used Web Search Logs for measuring similarity of short text, and both Strube and Ponzetto [194] and Gabrilovich and Markovitch [71] used rich encyclopedic knowledge of Wikipedia for Semantic Relatedness.

Strube and Ponzetto [194] made a system called WikiRelate! which calculates the relatedness score of words by finding Wikipedia articles that contain words in their titles. They made use of previously developed measures for WordNet which in their calculation relied on the content of Wikipedia articles and the path distances found along the category taxonomy of Wikipedia. Gabrilovich and Markovitch [71] proposed a technique called Explicit Semantic Analysis (ESA) which calculates Semantic Relatedness between words and text of any length (unlike [194] which operates over words only); the technique bases itself on the vector space model using Wikipedia. The input is represented as a vector and is then scored on the basis of association with documents in the collection i.e., Wikipedia. Even though ESA gathered attention in the research literature [72, 60, 175] it does not exploit the hypergraph of Wikipedia and this was filled by two later approaches [217, 221]. Milne and Witten [217] made use of tf.idf-like measures on Wikipedia links and Yeh et al. [221] made use of random walk algorithm (Personalized PageRank [90]) over the graph driven from Wikipedia's hyperlink structure, infoboxes, and categories.

An empirical analysis by Strube and Ponzetto [194] demonstrates the strength of Wikipedia-based semantic relatedness measures over those based on WordNet. With respect to comparing the performance of various Wikipedia-based semantic

[1]A detailed overview and analysis of a hyperlink-based approach is given in Appendix B.

Type	Sentence
Input(Unannotated)	Joe and Alan worked for Luther Corp. in 1982.
Output(Annotated)	[Joe]$_{PERSON}$ and [Alan]$_{PERSON}$ worked for [Luther Corp.]$_{ORGANIZATION}$ in [1982]$_{TIME}$.

Table 3.1: Example showing application of NER over a sentence

relatedness measures there is inconsistency in results, and one underlying reason for this is the different application scenarios for which they have been devised [46]. We differ from proposed techniques in that we utilise Wikipedia categories in conjunction with Wikipedia articles whereas earlier works utilise either Wikipedia hyperlinks or category hierarchies without taking into account their combination[2].

3.2 Named Entity Recognition

Named Entity Recognition (NER) is a key task in information extraction and forms related work for our contributions made in Chapter 5. NER fundamentally involves annotating a snippet of text with a label from a set of fixed category types such as name of person, location, quantities, percentages, products, time etc. Formally, NER is performed in two steps: first, different block of texts are extracted from a document and then, each block of text is classified into a different range of category types [82]. Table 3.1 shows the application of NER over a sentence, where upper-case words show the annotated category type for the block of text by NER.

The NER task was first defined by Message Understanding Conference (MUC) in the mid 1990s [196] which involved recognition of people, organization, location, date, time, money, percentage, and quantity. Since then, NER has been addressed by approaches based on supervised, semi-supervised, and unsupervised methods.

The main idea behind supervised approaches is to study features of positive and negative examples of named entities over a large collection of annotated documents and design rules to capture instances of a given named entity type. In supervised approaches different researches have used a number of classifiers. One of the earliest algorithms by Bilkel et al. [21] employs a hidden markov model (which is a generative model) where each word could be assigned exactly one label i.e., either a class from fixed classes or no class at all. Works that followed [28, 49] utilised a maximum

[2]We give a detailed explanation of our semantic relatedness framework in Chapter 4.

entropy model where the model learns feature weights with conditional probabilities as opposed to joint probabilities with generative models. McNamee and Mayfield [137] utilised support vector machines where the task was modelled as a binary decision task. McCallum and Wei [136] utilised conditional random fields which is a statistical model known for pattern recognition.

Semi-supervised approaches are based on bootstrapping where a small amount of training is needed which act as seeds for the classification. Using the seed examples, the classifier learns the context with which the classified terms appear in order to judge the unseen data. Collins and Singer [44] observed a pattern for NER e.g., a proper noun followed by noun phrases (through the application of parts of speech tagging), Brin [31] utilised regular expressions in order to generate book titles paired with authors, and the works in [178, 47, 161] utilised mutual bootstrapping that kept growing a set of entities and a set of contexts.

Unsupervised approaches are generally based on clustering [151] where on the basis of similarity of context one can cluster named entities in a group. Furthermore, other approaches are based on exploiting external sources of evidence such as WordNet [7], Wikipedia [200], using pointwise mutual information (PMI) over large Web corpus where a high PMI value means that expressions co-occur usually [62].

Supervised approaches to named entity recognition continue to dominate research within this area through their high accuracy. However, this high accuracy is achieved through a huge amount of training data makes it impractical in large-scale settings [151]. Semi-supervised approaches are now able to achieve accuracy closer to supervised approaches, and are the prevalent technique in use. The task we address in Chapter 5 despite being somewhat related to named entity recognition is not directly solvable through simple identification of named entities as it primarily involves filtering tweets with respect to a given entity[3].

3.3 Disambiguation Problem

In this section we provide an overview of works that deal explicitly with associating terms/phrases/entities with their meanings, and these associations constitute a key phase for the task we contribute to and present in Chapter 5. We first present an overview on Word Sense Disambiguation which is followed by an overview on Named Entity Disambiguation in long texts and in tweets.

[3]We give a detailed explanation of the task in Chapter 5.

Type	Sentence
Input(Unannotated)	They like grilled bass.
Output(Annotated)	They like/ENJOY grilled/COOKED bass/FISH.[4]

Table 3.2: Example showing application of WSD over a sentence

3.3.1 Word Sense Disambiguation (WSD)

It is a NLP task which deals with the assignment of correct word senses to words when words carry multiple meanings or senses [152]. WSD makes use of knowledge resources because without knowledge it would be impossible for machines (as well as for humans) to identify the meaning of words. These knowledge resources can be corpora of text, machine-readable dictionaries, thesauri, glossaries, ontologies or lexical resources such as WordNet [152, 58]. Table 3.2 shows the application of WSD over a sentence, where upper-case words refer to the sense of the word separated by a '/'. Word 'bass' can also refer to low-frequency tones of sound.

The knowledge-based Word Sense Disambiguation algorithms that have been proposed in recent years generally fall into two main groups: similarity-based methods and graph-based methods. Similarity-based methods operate by computing the similarity between each of the possible senses of a "word" and the words in the surrounding context [71, 163]. Graph-based methods operate by building a graph that represents all available senses of all of the words being disambiguated [4, 153]. The nodes in the graph represent the different senses while the edges represent the semantic relations such as synonymy, antonymy, hyperonymy, etc. between them. Graph centrality algorithms are then applied to determine the important nodes, which are then considered to be the correct senses of the target words.

3.3.2 Named Entity Disambiguation (NED)

Similar to WSD there is another NLP task termed NED which deals with the identification of entities inside text, where an entity is an uniquely identifiable "thing" or "object", e.g., people, companies, products, locations, etc [140]. Linking entities mentioned in text is generally achieved with the help of a Knowledge Base (KB) [85] such as Wikipedia; therefore the task is also called Wikification [40] due to linking of entities in text with Wikipedia pages (called concepts). Recently, Li et at. [126] argued that KB is not enough for NED task and they proposed a hybrid model making

use of both a KB and external sources of evidence.

There are some works which perform NED without using a KB. Davis et al. [51] addressed NED in streaming data on Twitter since the content of a single tweet is too short for performing entity disambiguation. In order to overcome the problem of short context in a tweet, the authors propose to merge thousands of tweets together in a stream (per second) for defining context, which is then exploited for NED through the application of supervised learning (they used Expectation-Maximization algorithm for classification).

The task of NED using KB mainly involves three steps: mention detection, link generation, and best candidate selection. In the first step, linkable phrases or spots from a text are generally identified using Named Entity Recognition (NER), different rule based approaches, or different measures such as key phraseness or link probability i.e., how much the phrase is linkable inside the KB [141, 142, 48, 144, 139, 64, 84]. In the second step, all possible candidates for linkable phrases are identified, the linkable phrase which contains only one possible candidate is an unambiguous entity while the linkable phrases which contain more than one candidate are defined as ambiguous linkable phrases. In this step, candidates for entities are topically identified by a direct reference inside the KB (i.e., if a phrase matches the phrase of entity inside KB) or by different strategies of machine learning or measures like commonness i.e., how often a particular phrase links to different entities inside the KB (e.g., "world cup" anchor text within Wikipedia referring to Wikipedia articles "FIFA World Cup", "World Cup (men's golf)", etc.) [141, 144, 139, 147, 85]. In the final step, disambiguation is performed for the linked phrases which contain multiple candidates for entities in the KB. In this step unambiguous linked phrases define the context for ambiguous linked phrases that are disambiguated by similarity functions [142, 48, 144, 35] and semantic relatedness [64, 147, 74, 119].

Most of the Wikification methods require clean and grammatically correct texts showing good results in experimental evaluations for long texts but they have been shown to perform poorly on tweets as described by Meij et al. [141]. In spite of the great significance of extracting commercially useful information from tweets, the amount of research dedicated to entity name disambiguation in tweets is very limited. Three serious efforts have been undertaken which are by Ferragina and Scaiella [64], Meij et al. [141], and Habib and Keulen [85]; all these approaches use Wikipedia[5] for the task at hand. The TagMe system [64] uses the hyperlink structure of Wikipedia

[5]Habib and Keulen [85] used Yago KB which is built on Wikipedia and they also used Google API.

by exploiting the links between Wikipedia pages and the anchor texts of the links to those Wikipedia pages. Disambiguation is performed by application of a collective agreement function (i.e., a voting function) among all senses associated to anchors detected on the input texts and similar to the work of Meij et al [141], unambiguous anchors are utilised to boost the selection of these senses for the ambiguous anchors. Meij et al. [141] employ supervised machine learning techniques for refinement of a list of candidate Wikipedia concepts that are potentially relevant to a given tweet. The candidate ranking list is generated by matching n-grams in the tweet with anchor texts in Wikipedia articles, taking into account the hyperlink structure in Wikipedia to compute the most probable Wikipedia concept for each n-gram.

The named entity disambiguation approaches in the literature are mostly reliant upon anchor texts and Wikipedia hyperlinks. However, on the application of Wikipedia hyperlink structure we found less than optimal performance whereas Wikipedia category-article structure provides optimal performance for the task addressed in Chapter 5. This can be on account of the fact that named entity disambiguation is the task that disambiguates an entity used within the text whereas the task we address in Chapter 5 differs in that the entity is pre-defined.

3.4 Seeking Information for Complex Needs

The past few years have witnessed a rapid growth of the World Wide Web (WWW) and an accompanying "information overload" [19]. The immense growth of World Wide Web coupled with the diverse user base has made the process of information seeking overly complex [91, 94]. Research attempts that aim to satisfy users' complex information needs have been along the following dimensions:

- Search result diversification
- Exploratory search

We present a brief overview of each of these research dimensions in the following subsections. Note that research attempts for addressing complex information-seeking is a key task in information extraction and forms related work for our contributions made in Chapter 6.

3.4.1 Search Result Diversification

The tendency of users to search for information covering different facets of their information need gave birth to the field of "search result diversification" which has emerged as a means of avoiding over-specialization and homogeneity in search results [57]. The fundamental goal of result diversification is to tackle the issue of query ambiguity on the user side [188]. Furthermore, search result diversification aims to avoid the "filter bubble" effect wherein the user is presented with the same type of information over and over again [156].

The research literature defines the definition of diverse results of an information filtering or information retrieval system into three different categories based on:

- Content: Content-based definitions of diversity aim to maximize the distance between items in a result set of a recommendation system or search engine [229, 230].
- Novelty: Novelty-based systems seek to maximize the freshness of the returned items/search results by avoiding redundancy in terms of not selecting documents previously seen [42].
- Coverage: Coverage-based definitions of diversity attempt to select resultant items so as to maximize the covered aspects of the users' information need [5].

3.4.2 Exploratory Search

Exploratory search attempts to go beyond the query-response paradigm and can be considered a specialization of information exploration [134]. Broadly speaking it represents a research area concerned with the design of systems that support the user in his/her journey of satisfying an information need through browsing within the information space. Exploratory search systems go beyond single-session lookup tasks and support complex search scenarios [214].

Examples of exploratory search systems include the following

- Information visualization systems: These systems attempt to incorporate graphical designs that provide information summaries of the search results so as to assist the user in the information seeking process [32, 80].
- Document clustering and browsing systems: These systems group search results into clusters in order to offer users with the ability to sieve through the result set for better exploration of the information space [201, 226].

- Intelligent content summarization systems: These systems complement search results with textual summaries in order to provide the user with an overview of the major themes present in encountered information [55, 117].

3.5 Knowledge Extraction

Knowledge extraction aims to preserve the meaning of textual units of information by providing a concise representation of documents. Specifically, it consists of approaches for document summarization and keyword extraction. It is important to note that document summarization is not a direct related work to any of the contributions made in this thesis whereas keyword extraction is presented as a contribution in Chapter 7. However, due to it being a closely investigated area related to keyword extraction, we include it as a related work in the thesis for a better presentation of the related research. First, we present an overview of document summarization; it is the task that generates a summary of a document in a few words while retaining the important points of the document. Then, we present an overview of keyword extraction; it is the task that extracts most important keywords which represent the gist of a document while omitting the sentence based structure of the document.

3.5.1 Document Summarization

There are two ways to summarize a text document [212]: extraction based methods and abstraction based methods.

In extraction based methods, actual sentences or snippets from documents are extracted and scored using a combination of statistical and linguistic features such as the position of a snippet in the document, cue-phrases, formatting, and frequency [212]. According to research [15, 30, 38, 120, 198] the position of the first occurrence of a phrase is an important feature for the summarization of news articles and scientific reports. Similarly, Lin and Hovy [127] reported different optimal positions for document summarization for the documents belong to different domains. Research also indicates that cue-phrases i.e., phrases proceeded by 'in conclusion' and 'significantly' are an important indicator for generating a summary because this terminology emphasizes the importance of text that follows it [59, 120, 198], similarly words such as 'impossible' and 'hardly' are examples of sentences which are not important[6] for summaries in scientific articles [120]. Research by Edmundson [59] and Teufel and

[6]More likely to be not important.

Moens [198] concludes that formatting when used as a feature helps with summarization e.g., sentences in bold, title or head. Research [61, 132, 174] also shows that features based on word frequency (such as term frequencies, tfidf scores) are particularly noticeable for summarization. Research by Kupiec et al. [120] made use of sentence length and presence of upper-case words and found these features useful. Generally the proposed approaches in the extraction based methods make use of document structure and are therefore, fine-tuned according to the nature of the document collection such as scientific articles, conversational or meeting points, etc.

In abstraction based methods, the internal semantic structure of a document is first realized which is then exploited to generate summaries closer to the human generated summaries using natural language processing techniques. This method generates paraphrased summaries of the document which may include sentences which were never used in the original document. To achieve abstraction of summaries which is a relatively less explored research area compared to extraction based methods, later research [177] proposes to fill predefined templates by generation of summaries through extracting information which would fit in the predefined templates but this idea is too domain-specific[7]. Another research work [116] proposes a compression algorithm for text using *expectation maximization* which reduces sentences to shorter lengths using syntactic parse trees. Furthermore, exploiting similarity and repetition of sentences helps uncover topics which can then lead to generation of paraphrased fusion of sentences [18, 37, 133].

3.5.2 Keyword Extraction

Recent years have seen keyword extraction as a dominant technique for summarizing the contents of a document with numerous applications in various information access tasks such as exploratory search and query expansion, and in various text mining tasks such as document classification and document clustering to name a few.

Due to the differences in the nature of textual documents, generally four document specific factors have influenced keyword extraction techniques i.e., length of document, structural consistency of document, possibility of topic change within the document, and possibility of topic correlation among topics within the document [88]. The longer the document, the more candidate keywords are available (e.g. scientific articles and technical reports compared to news articles and emails). A well structured document contains certain sections (fields) and formatting that can be exploited

[7]Because of predefined templates.

for keyword extraction, such as a scientific paper's title and abstract [115], and meta-data of webpages [222]. Documents such as conversational texts, logs of open-ended meetings generally contain several topics in sequence[8], and in such type of documents a topical change happens as the discussion moves[9] [113]. Documents such as news articles and scientific articles possess a topical correlation (i.e., interconnected topics) in the entire flow of the article unlike informal chat, and in these type of documents the keywords are usually related with each other [145, 203].

Several approaches have been proposed in the literature to address the problem of keyword extraction from a piece of text. However, keyword extraction is generally performed in two steps, first a list of candidate keywords are extracted using some heuristics, and then each candidate is scored using either a supervised or an unsupervised strategy. A candidate keyword is usually extracted on the basis of n-grams [69, 104, 138, 218], words with specific parts of speech tags (nouns, verbs, adjectives) [128, 145, 210], noun phrases [17, 202, 219], words other than stopwords [130], and n-grams appearing as Wikipedia articles titles [81]. Scoring each candidate keyword in a supervised strategy is influenced by the selection of different features and by the process of task re-definition while scoring in an unsupervised strategy is addressed by graph based approaches and topical clustering.

3.5.2.1 Supervised strategies

Supervised strategies generally use statistical, structural, and syntactic features from documents, and features based on external sources of evidence to infer a function from labelled training data. Most renowned statistical features are tf-idf [179, 186], distance of a phrase[10], probability of a phrase being a keyword in the training-set. These are the feature-sets used by [69, 218]. Moreover, these features have been shown to perform well on different sources of documents [115, 222]. Some other statistical features are the phrase length and the number of words between the first and last occurrence of a phrase in a document [88]. The document structure helps in identifying keywords; for example the usage of phrases in different sections of scientific articles carries a different emphasis (e.g., abstract, introduction) [155], and the location of usage in a Web page (e.g., title) [39, 222]. Among the syntactic features, sequences of parts of speech tags (nouns, verbs, adjectives) assigned to word sequences and suffix sequences (morphological suffixes) are commonly used [114, 155,

[8] As in talking points.

[9] E.g., first topic can be about cleaning, second can be related to cooking, etc.

[10] It is defined as words preceding the first occurrence of the phrase normalized by total number of words in the document.

222]. Research has also exploited external sources of evidence such as search engine query logs [222], Web [203], and Wikipedia [138]. If a phrase is discovered in the query logs then it makes sense to consider it as a potential indicator of being a keyword [222]. Likewise, if a phrase is found in databases of terminologies such as scientific papers [131] then it is also an indicator of being a keyword. Semantic relatedness scores were calculated as a feature by [203] using the Web as an external source of evidence; if a candidate keyword is not semantically related to predicted keywords then it is unlikely to be a keyword in technical reports [88]. A measure based on Wikipedia (called *keyphraseness*) is used by [138], which scores the possibility of a phrase to have a linkable article on Wikipedia; once the score[11] is learned using training data, both seen/unseen phrases can be classified using the measure *keyphraseness*.

Initially, supervised strategies for keyword extraction redefined the task as a classification task [69, 202, 205, 218] i.e., a binary decision whether or not the candidate keyword is actually a keyword instead of ranking the keyword with a continuous value; in these approaches keywords are uniformly important i.e., no keyword is more important than the others [105]. Later on, this issue was addressed by ranking each keyphrase by learning a ranker function [111].

3.5.2.2 Unsupervised strategies

In unsupervised strategies graph-based techniques have been a popular choice for keyword extraction [29, 145, 211, 212, 213, 228]. These techniques are inspired by the PageRank algorithm [159]; in these techniques the words of a document are modelled as nodes, while the edges between them define the relatedness between them [29, 209, 211]. A word is important when it is linked by other important words [135, 145], and this importance is estimated using the PageRank algorithm. TextRank [145] is one of the well-known algorithms for unsupervised keyword extraction [81, 88], this technique makes use of the PageRank algorithm where words are nodes and these nodes are connected whenever there is a co-occurrence in the text. ExpandRank [210, 211] is an extension to TextRank by augmenting the graph with highly similar documents. ExpandRank requires an input parameter which is a small number of neighbouring documents[12] of the considered document. However, to discover these documents, the technique uses the cosine similarity, which is computationally expensive and practically inapplicable to large datasets. Furthermore, the

[11] Threshold.
[12] Number of similar documents.

exploitation of a neighbourhood of documents may produce a topic drift resulting in the extraction of noisy terms for the considered document [129].

Topical clustering is another unsupervised strategy that arranges candidate keywords into topics before extracting keywords for a document [81, 130]. The system described by [130] clusters semantically similar candidates by using both co-occurrence statistics and Wikipedia; while extracting keywords it gives to each topic a uniform importance, which is a drawback [88]. Similarly, Grineva at al [81] extract keywords from the top-k topics (i.e., k needs to be manually defined) while ignoring the influence of the rest of topics.

Instead of utilising the document features our technique makes use of Wikipedia category-article structure which makes it unsupervised and able to operate over short-text. We compare our technique with above-explained unsupervised keyword extraction strategies in Chapter 7.

3.6 State-of-the-Art in Lieu of Thesis Contributions

The subsequent chapters of this thesis cover our semantic relatedness framework along with its use in various subtasks of "online reputation management" (Chapters 4 and 5), "perspective-aware search" (Chapter 6), and "keyword extraction" (Chapter 7). The works described in previous sections served as significant background to aid the reader in understanding the current state of the field wherein the role of knowledge bases such as Wikipedia is limited.

The state-of-the-art presented in previous sections touches upon various aspects of our contributions. Named entity recognition is considered closely related to the task we address within the context of reputation management. One fundamental difference however lies in the nature of how entities need to be identified within ambiguous tweets and simple categorization into various entity types does not address the problem given the complete tweet. Disambiguation, and particularly named entity disambiguation, is closely related to the filtering task addressed in Chapter 5. The techniques for named entity disambiguation that we presented in Section 3.3.2 do not achieve maximum contextualization for entity mentions; we on the other hand in an attempt to resolve the second and third research question obtain a suitable set of related terms corresponding to an entity through the Wikipedia category-article structure[13].

[13]Detailed explanation follows in Chapters 4, 5, and 6.

Complex information-seeking ties in with the third research question as it introduces a class of search whereby the user is facilitated for investigating various facets of his information need via search result diversification or exploratory search. Chapter 6 proposes a novel exploratory search interface that is both an information visualization system and a content summarization system (refer to Section 3.4.2).

Finally, knowledge extraction approaches aim to achieve preservation of the meaning of a textual piece traditionally via document summarization or keyword extraction. In an attempt to demonstrate Wikipedia's usefulness in preservation of the meaning of a textual piece and thereby tying in with the first and second research question we focus on knowledge extraction approaches. We differ from presented works in that instead of relying on document features we utilise Wikipedia category-article structure in a community detection framework[14].

3.7 Summary of the Chapter

This chapter presented an overview of works related to ours within the text mining domain. In an attempt to present a summary of research efforts within the domain of text semantics, we began with an introduction to semantic relatedness. This was followed by an overview of named entity recognition, word sense disambiguation, and named entity disambiguation. We then presented a summary of research efforts that attempt to address the problem whereby complex information seeking scenarios arise and we fundamentally focused on search result diversification and exploratory search. Finally, we presented details of research works focusing on knowledge extraction efforts. Within the domain of knowledge extraction, we covered document summarization and keyword extraction. In conclusion, we attempted to position related work with the contributions of this thesis that we explain in subsequent chapters (specifically Chapters 4, 5, 6, and 7).

[14]Detailed explanation follows in Chapter 7.

Chapter 4

Wikipedia Based Semantic Relatedness Framework

This chapter presents the proposed semantic relatedness framework which constitutes the core of the methodology for chapter 5 and 6. This framework is composed of two steps as explained in this chapter.

Our view on semantic relatedness follows from the definitions in Chapter 3 whereby we define it as a means to offer reasoning over textual units. In the context of this thesis, semantic relatedness serves as a means for inference of a relationship between textual units, and these textual units are the candidate phrases explained in Section 4.1 with the relationships not being limited to similarity. Instead we model semantic relatedness as explicit and implicit connections between the concepts representing textual units and therefore, our notion of semantic relatedness is not restricted to identification of relationships such as musician1:musician2[1] [162] but can also identify relationships like microsoft:windows10[2].

In the following sections, we first explain the process of candidate phrase generation performed through the chunking of textual data into variable-length phrases using Wikipedia. This is followed by an explanation of the strategy to produce relatedness scores through the exploitation of the Wikipedia category-article structure.

4.1 Generation of Candidate Phrases

Candidate phrases in the context of our thesis contributions are the phrases extracted from textual data which constitute the fundamental building blocks upon which our semantic relatedness framework is built. In the context of Chapter 5 the considered

[1] musician1 and musician2 are two different musicians such as Madonna and Lady Gaga.
[2] Microsoft is a company whereas Windows10 is a product of Microsoft.

phrases are from within tweets and we calculate semantic relatedness between these phrases and the pre-defined entity[3]. On the other hand, in the context of Chapter 6 the considered phrases are from within news articles and we calculate semantic relatedness between these phrases and a "perspective concept".

4.1.1 Variable-Length Phrase Chunking

In the literature, phrase chunking has been traditionally performed through part-of-speech tagging [45]. As we explained earlier (refer to Section 2.3 of Chapter 2 of this thesis), tweets which form an essential component for the contributions in this thesis, are written in an informal manner and hence, lack proper grammatical structure due to which part-of-speech tagging fails [76]. Additionally, extracting a phrase through part-of-speech tagging does not guarantee a match with a Wikipedia article which is an essential requirement for the proposed framework. Therefore, we devise our strategy for variable-length phrase chunking by making two intuitive assumptions as follows:

- A phrase that contains more words is usually more informative than a phrase that contains less words, e.g., 'computer science' is more informative than 'science'.
- A single term which is not a stopword is more informative than a single term which is a stopword, e.g., 'science' is more informative than the stopword 'of'.

Figure 4.1 shows the phrase chunking strategy that we employ. In the first step, the textual content (say, a sentence or a tweet) is converted into lowercase (to avoid case-sensitivity). Then, phrase boundaries (such as commas, semi-colons, sentence terminators etc.) are used for chunking the content into phrases. In the case of tweets, phrase boundaries also include tweet-specific markers (such as @, RT etc.). Finally, the extracted phrases are further reduced to those that match a Wikipedia article title or redirect. Preference is given to the extraction of the longest phrase. In the final step, there is an exception rule to ignore a phrase or word which matches exactly a stopword. Figure 4.1 shows the removal of stopwords such as 'i', 'over', etc, and it also shows extracted phrases such as 'samsung s5', 'htc', etc. The pseudo-code for the variable-length phrase chunking step is shown in Listing 4.1.

[3]It is this pre-defined entity corresponding to which the tweet has to be disambiguated as explained in Chapter 5.

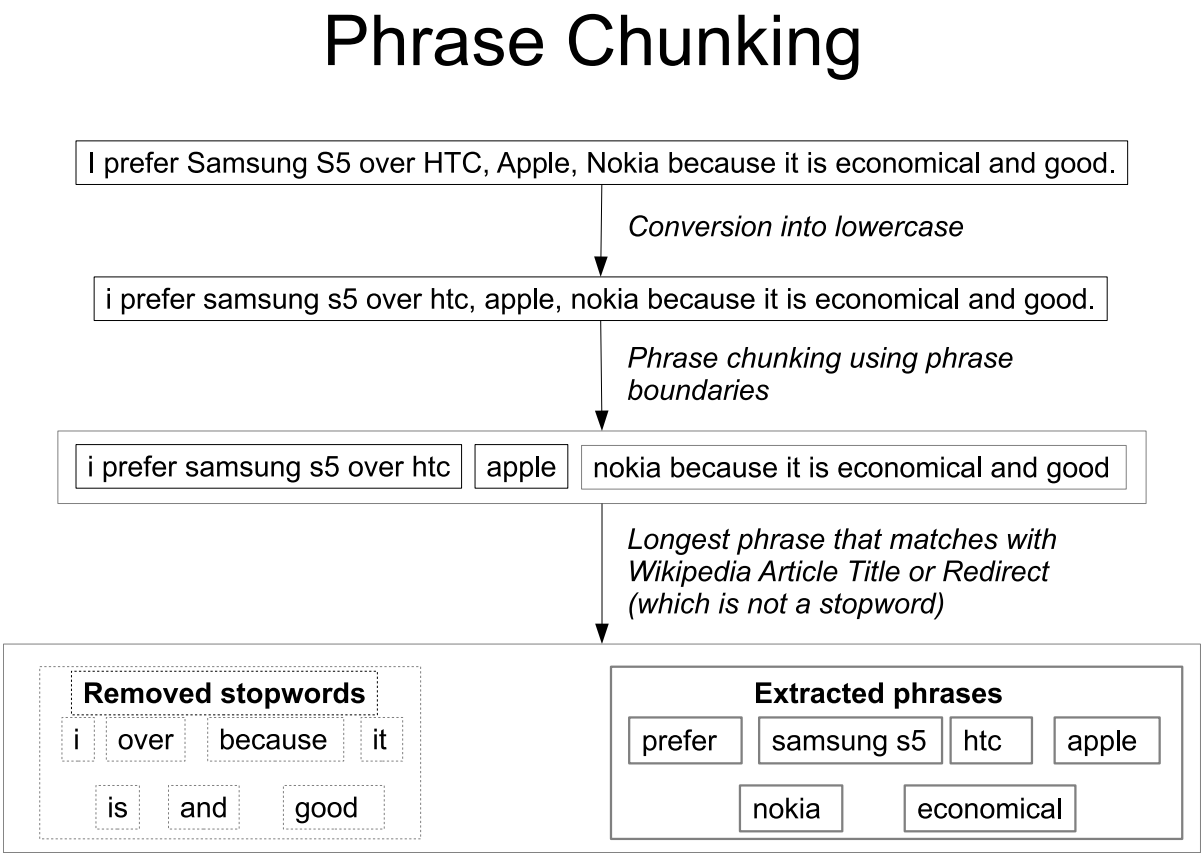

Figure 4.1: Strategy of phrase chunking using Wikipedia

Listing 4.1: Psuedo-Code for Phrase Chunking using Wikipedia

```
1  def main():
2   extractVariablePhrases(text) # main functionality
3
4  def longest_possible_phrases(phrase):
5   ngram_lst = extract_allpossible_ngram(phrase)
6   order_ngram_lst = ngram_lst order by highest n-size first
7   matched_lst = []
8   for ngram in order_ngram_lst: # by default all ngrams are
       not deactive
9    if ngram.deactive:
10      continue
11   if matches(ngram, wiki_article_title) or matches(ngram,
       wiki_article_redirect):
12     matched_lst.append(ngram)
13     ngram.substring_branches(deactive) # all substrings of
          this ngram's branch gets deactivated
14  return matched_lst
```

```
15
16 def extractVariablePhrases(text):
17  text = text.lower() # (in order to avoid case-sensitivity).
18  phrase_boundaries = [',', '.', ..., '@', '#', etc.] # (i.e
        ., commas, semi-colons, etc.) & tweet-specific markers (
        such as @, #, etc)
19  phrase_lst = text.split(reg_ex(phrase_boundaries)) # split
        text using phrase boundaries
20  matched_phrases = []
21  for phrase in phrase_lst:
22   ph_lst = longest_possible_phrases(phrase)
23   for ph in ph_lst:
24    if not ph==stopword:
25     matched_phrases.append(ph)
26  return matched_phrases
```

One limitation of our proposed phrase chunking strategy is its tendency to possibly miss out some terms on account of the matching strategy looking for exact matches with Wikipedia article titles and redirects[4].

4.2 Relatedness Scores Using Wikipedia Category Hierarchies

In this section we present our strategy for generating relatedness scores which uses the Wikipedia category-article structure. Note that the relatedness scores are generated for textual phrases (i.e. candidate phrases as explained in section 4.1) with respect to a certain entity where an entity is a thing or concept with an independent existence such as a brand, company, celebrity, topical interest etc. For example, our aim can be to measure the relatedness of a piece of text to some real-world entity. Having extracted phrases from the text, we wish to score these phrases in terms of relatedness. In order to do so, we exploit the Wikipedia category taxonomies and the articles that are mentioned inside those category taxonomies as explained in the following subsection. Note that we utilise WikiMadeEasy API for querying Wikipedia which facilitates efficient access to Wikipedia data; more details of WikiMadeEasy API are given in Appendix A.

[4]We aim to overcome this limitation as part of future work; a detailed explanation follows in Chapter 8.

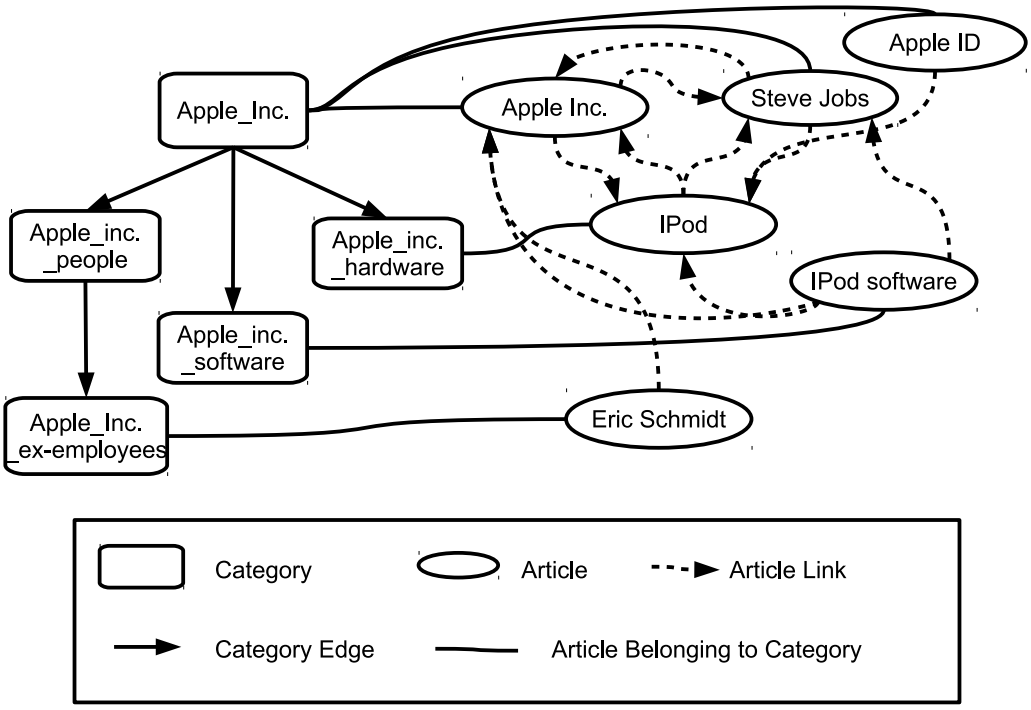

Figure 4.2: Truncated Category-Article Structure for Concept "Apple Inc."

4.2.1 Generation of Relatedness Scores

Wikipedia contains a huge and diverse amount of semantics pertaining to all entities in the form of related terms, article redirects, article hyperlinks, infoboxes[5], parent and child categories etc. Semantics is a broad term mainly used to represent the meaning and useful connections behind entities which is normally built upon extensive knowledge pertaining to an entity. As an example, the entity "Steve Jobs" represents the founder of company "Apple Inc."; however, to make this connection about entity "Steve Jobs" one would have to posses knowledge about entity "Apple Inc."[6]. Wikipedia categories (i.e., parent and child categories) are particularly useful in that they can be used to infer or derive additional information pertaining to an entity. In fact, the Wikipedia category taxonomy can be representative of an entity; note that the choice of chosen category taxonomies to represent an entity is dependent upon the application scenario and we separately explain this process in Chapter 5 and 6 for the different applications under consideration. Here, for the sake of simplicity, we assume that a category taxonomy for which the relatedness score

[5]An infobox is a fixed-format table designed to be added to the top right-hand corner of Wikipedia articles to consistently present a summary of some unifying aspect pertaining to the articles.

[6]An example category taxonomy for Apple Inc. can be seen on left side of Fig. 4.2.

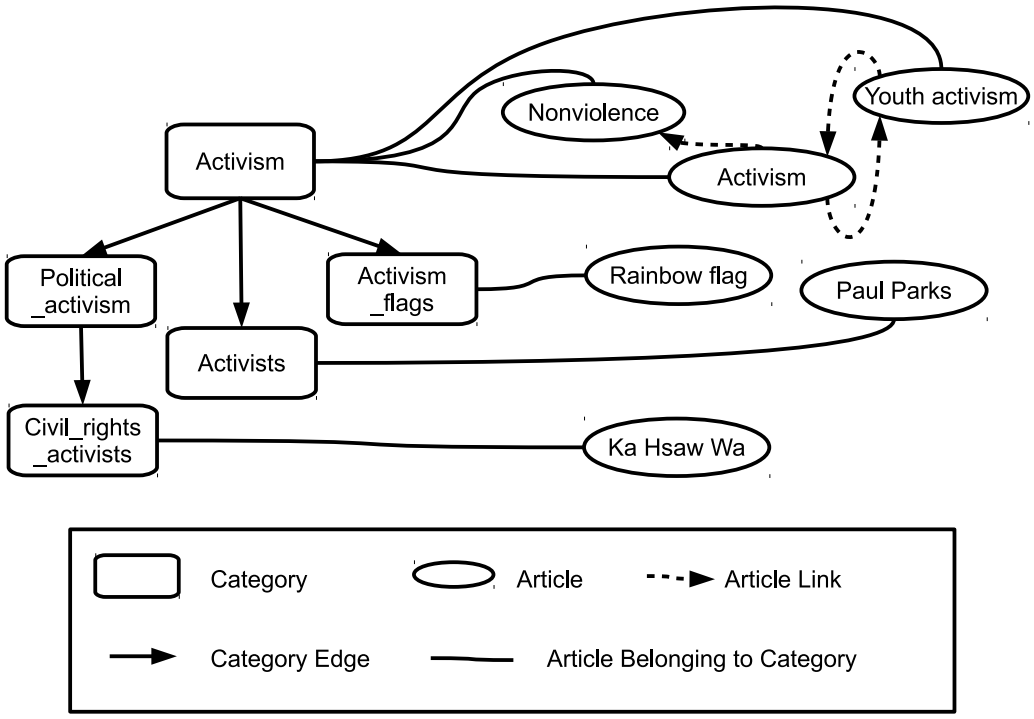

Figure 4.3: Truncated Category-Article Structure for Concept "Activism."

is to be calculated is an arbitrary category along with its sub-categories to a depth of two[7]. A depth of two is utilised as the optimal setting for inference of a relationship between a candidate phrase and Wikipedia category. Going further down in the depth is computationally expensive due to heavy interlinking of Wikipedia categories, whereas with a depth count of two we observed reasonable evaluation results without degrading performance. Figure 4.2 shows the truncated Wikipedia category-article structure for the entity "Apple Inc." and Figure 4.3 shows the truncated Wikipedia category-article structure for the concept "Activism" up to a depth count of two. The inter-connections between the Wikipedia categories and Wikipedia articles are utilised in our semantic relatedness framework as explained below.

Each category taxonomy has exactly one parent category and usually several sub-categories. We refer to all these categories as RC (i.e., it contains all related categories in a hierarchy from depth count of zero to two). These categories RC contain different Wikipedia articles, we refer to these articles as $Articles_{RC}$. These articles $Articles_{RC}$ are also mentioned in categories other than RC and we retrieve all categories that contain $Articles_{RC}$ and refer to them as WC; note that RC is a subset of WC. Table 4.1 summarizes the above-explained conventions. Note that in Figures 4.2 and 4.3

[7]It is important to note that a category representative of the entity is selected at this phase.

Convention	Explanation
RC	Set of parent category and subcategories to depth of 2 (i.e., list of categories in a hierarchy)
$Articles_{RC}$	Set of Wikipedia articles which are mentioned in at least one category from RC
WC	Set of all Wikipedia categories that mention Wikipedia articles in $Articles_{RC}$, therefore $RC \subset WC$

Table 4.1: Conventions

all the Wikipedia categories that are shown (using the rounded rectangle symbol) represent RC and all the Wikipedia articles that are shown (using the oval symbol) represent $Articles_{RC}$.

The candidate phrases extracted from phrase chunking (explained in Section 4.1) that match an article title or redirect in $Articles_{RC}$ are called matched phrases. We use these matched phrases to calculate the relatedness score. In the next section, we summarize the factors which contribute in calculating the relatedness score of a candidate phrase using the Wikipedia category-article structure.

4.2.2 Relatedness Measures

As explained earlier, we aim to capture the relationship between the concepts represented by two textual units and in doing so we capture how related they are within the Wikipedia category-article structure. The relatedness measures introduced in this section have been devised to capture the closeness between two concepts within the Wikipedia category taxonomy via the relatedness measure related to depth significance, and the number of common categories between two concepts via the relatedness measure related to category significance. Moreover, the significance of the phrase itself is taken into account so as not to overemphasize relatedness when the phrase itself is insignificant.

The traditional semantic relatedness measures found in the text mining literature make use of path length between Wikipedia categories or Wikipedia articles without taking into account both simultaneously [96, 194, 217]. The uniqueness of our measures lies in their ability to capture implicit relationships between concepts and this is on account of direct utilisation of Wikipedia categories where a corresponding match

occurs with a Wikipedia article representing a candidate phrase (further details in Section 4.2.2.1). Furthermore, our relatedness measures utilise the notion of category overlap (further details in Section 4.2.2.2) which to the best of our knowledge has not been done previously. One potential limitation however is the introduction of noisy relationships as we aim to increase coverage of relationships between concepts[8].

Below, we discuss three separate relatedness measures; these relate to depth, number of categories, and phrase frequency. Finally, we present the aggregation of these measures into a single measure. Note that we use non-normalized versions of relatedness measures as the range of values for Wikipedia category-article based heuristics is not wide, and moreover, we wish to capture even subtle relationships between the concepts represented by the textual units[9]. In the formulations presented below we use the notation of 1) p to denote the candidate phrase for which a relatedness measure is to be calculated, and 2) cat_t to denote the category taxonomy corresponding to the entity under consideration.

4.2.2.1 Heuristic 1: $Depth_{significance}$

$Depth_{significance}$ denotes the significance of category depth at which a matched phrase occurs. Each potential branch in a category is of a certain depth; the further down the category the greater the specialization. As we move further up the category, we are potentially moving further away from the context expressed in the original subcategory (e.g., automata $\subset$ computer science $\subset$ science $\subset$ knowledge). One intuition that follows from this is that the deeper the match occurs in the taxonomy the less its significance to the entity under consideration. This means that the matched phrases in the parent category of the entity under investigation are more likely to be relevant to the entity than those at depth of two. To capture this heuristic, we introduce the notion of depth and we assign $Depth_{significance}$ as a measure of relatedness.

$$Depth_{significance}(p, cat_t) = \sum_{cat \in RC \cap p_{categories}} \frac{1}{depth_{cat} + 1} \tag{4.1}$$

In the above formula, $p_{categories}$ denotes the categories in which the matched phrase appears. A $Depth_{significance}$ score is computed for each $p_{category}$ in RC, and an overall score for the considered matched phrase is obtained by summing up all the obtained significance scores. For an intuitive understanding of the $Depth_{significance}$

[8]We comment on these aspects further in Chapters 5 and 6.

[9]Normalizing a subtle relationship may result into mathematical zero due to small fraction and storing a low fraction with high precision is not an efficient choice.

score, consider the Wikipedia article "Eric Schmidt" belonging to Wikipedia category "Apple Inc. ex-Employees" (refer to Figure 4.2); the phrase "Eric Schmidt" is not highly related with the entity "Apple Inc." and this is also signified by its match with a Wikipedia category deeper in the hierarchy and hence, our formulation for $Depth_{significance}$ in above equation assigns a lower score to this phrase. Furthermore, with respect to the concept "Activism" in Figure 4.3 note that our technique is able to discover a significant phrase "Rainbow Flag"[10] and assigns it a high score according to $Depth_{significance}$ due to it being close to the parent category node.

4.2.2.2 Heuristic 2: $Cat_{significance}$

$Cat_{significance}$ denotes the significance of the matched phrase as expressed by the number of categories containing it. Intuitively, a matched phrase is more related to an entity when the Wikipedia categories of a matched phrase coincide with the categories in the category taxonomy of the considered entity. Therefore, the more categories of a matched phrase in RC, the higher the significance of that particular matched phrase with respect to the entity. To capture this heuristic, we introduce the notion of category and we assign $Cat_{significance}$ as a measure of relatedness.

$$Cat_{significance}(p, cat_t) = \frac{|RC \cap p_{categories}|}{|WC \cap p_{categories}|} \times log(|RC \cap p_{categories}| + 1) \tag{4.2}$$

$Cat_{significance}$ in the semantic relatedness model rewards the matched phrases which are densely inter-connected within the categories in RC.

4.2.2.3 Heuristic 3: $Phrase_{significance}$

$Phrase_{significance}$ is a combination of phrase word length and frequency of the phrase within the textual block from where it's extracted[11]. Intuitively, the greater the phrase length[12], the more informative or important it becomes, likewise the more frequent the phrase is in the textual block from where it's extracted, the more importance it assumes. To capture this heuristic, we introduce the notion of phrase and we assign $Phrase_{significance}$ as a measure of relatedness.

$$Phrase_{significance}(p, cat_t) = log(wordlen(p) + 1) \times p_{frequency} \tag{4.3}$$

[10]The phrase "Rainbow Flag" has a relatively high association with the concept of "Activism" as is obvious from many LGBT protests carrying the flag.

[11]This could be a paragraph, sentence or tweet.

[12]Number of words in a phrase.

$$p_{frequency} = log(freq + 1) \tag{4.4}$$

4.2.2.4 Summary of Relatedness Scores

We presented three separate relatedness scores to capture the relatedness of a matched phrase with a category taxonomy that is representative of an entity. The presented measures were $Depth_{significance}$, $Cat_{significance}$, and $Phrase_{significance}$. The measure of $Depth_{significance}$ captures the distance measure that depicts how far apart our entity of interest is from the candidate phrase being investigated thereby serving as a replacement of path-based semantic relatedness measures from within the literature [227]. The measure of $Cat_{significance}$ captures the category overlap between our entity of interest and the candidate phrase being investigated potentially expressing the richness of the candidate phrase with respect to the entity. Finally, the measure of $Phrase_{significance}$ captures the information content of the candidate phrase. Each of these measures captures a different aspect of semantic relatedness whereby each helps in uncovering a different type of relationship between the concepts represented by textual units as follows:

- $Depth_{significance}$ is able to highlight how closely the categories of a phrase match the categories of the category taxonomy representative of the entity of interest in terms of path length. The lesser the path length, the more significant the matched phrase with respect to the entity of interest.

- $Cat_{significance}$ is able to highlight the fraction of categories matched between the phrase and the category taxonomy representative of the entity of interest. The more the intersections, the more significant the matched phrase with respect to the entity of interest[13].

- $Phrase_{significance}$ is able to highlight the importance of the matched phrase in terms of its frequency and its length. The higher the frequency or the word length, the more important the phrase is.

We combine the three separate relatedness scores of $Depth_{significance}$, $Cat_{significance}$, and $Phrase_{significance}$ to give a unique relatedness score. More than one approach is possible for the aggregation of these measures, however we adopt[14] the following.

[13]This measure does not consider the path length.

[14]Empirically this aggregation performs reasonably well during the evaluations as shown in the later chapters.

$$Relatedness(p, cat_t) = Depth_{significance}(p, cat_t) \times Cat_{significance}(p, cat_t) \times Phrase_{significance}(p) \tag{4.5}$$

So far we have discussed generation of relatedness scores for matched phrases. These matched phrases are essentially taken from a piece of text where the piece can be a tweet, a scientific article, a news article, etc. The combined effect of $Depth_{significance}$, $Cat_{significance}$, $Phrase_{significance}$, and combined $Relatedness$ is applied over the entire text via the following summations:

$$Depth_{significance}(text, cat_t) = \sum_{p \in MatchedPhrases} Depth_{significance}(p, cat_t) \tag{4.6}$$

$$Cat_{significance}(text, cat_t) = \sum_{p \in MatchedPhrases} Cat_{significance}(p, cat_t) \tag{4.7}$$

$$Phrase_{significance}(text) = \sum_{p \in MatchedPhrases} Phrase_{significance}(p) \tag{4.8}$$

$$Relatedness(text, cat_t) = \sum_{p \in MatchedPhrases} Relatedness(p, cat_t) \tag{4.9}$$

Here, *MatchedPhrases* is used to denote the set of matched phrases that occur in a given piece of text i.e., *text* in Equations 4.6 - 4.9.

4.3 Summary of the Chapter

In this chapter, we presented two phases of our proposed semantic relatedness framework upon which we further build two chapters relevant to this thesis. We explained in detail the process of candidate phrase generation which involved conversion to lowercase followed by elimination of phrase boundaries and reduction via matching between the extracted phrases and Wikipedia article titles/redirects. The candidate phrases are then used in conjunction with the Wikipedia category-article structure for the calculation of relatedness scores. These relatedness scores are calculated with the help of Wikipedia category taxonomies representative of an entity; the parent category and sub-categories until a depth count of two (referred to as RC; see Table 4.1) are utilised by making use of articles that occur in these categories (referred to as

$Articles_{RC}$; see Table 4.1) together with all the other Wikipedia categories mentioning these articles (referred to as WC; see Table 4.1). Candidate phrases that occur in $Articles_{RC}$ referred to as matched phrases are utilised in three separate measures relating to category depth, category intersections, and phrase frequency to obtain relatedness scores. It is these relatedness scores that are utilised as features for "Online Reputation Management" tasks of Chapter 5 and "Perspective-Aware Search" of Chapter 6.

Chapter 5

Entity Filtering and Reputation Dimensions Classification for Online Reputation Management

This chapter presents two application scenarios which utilise the proposed semantic relatedness framework described in Chapter 4. The considered application scenarios arise in the context of online reputation management which is a research area emanating from the marketing domain. We begin the chapter with an introduction to the domain of online reputation management followed by an overview of the significant subtasks within this domain that we deal with; note that these tasks are addressed in the context of CLEF evaluation campaigns[1] called RepLab which were specifically devoted to online reputation management. These tasks are the filtering task and the reputation dimensions classification task; the filtering task involves determining whether or not a tweet is relevant to an entity while the reputation dimensions classification task involves classification along various dimensions related to the various facets of an entity's reputation. We then explain the various explicit and implicit challenges involved in the subtasks related to online reputation management. Finally, details of our methodology to address the two tasks are presented which utilises as core features Wikipedia-based relatedness scores described in Chapter 4. Extensive experimental evaluations and their results conclude the chapter.

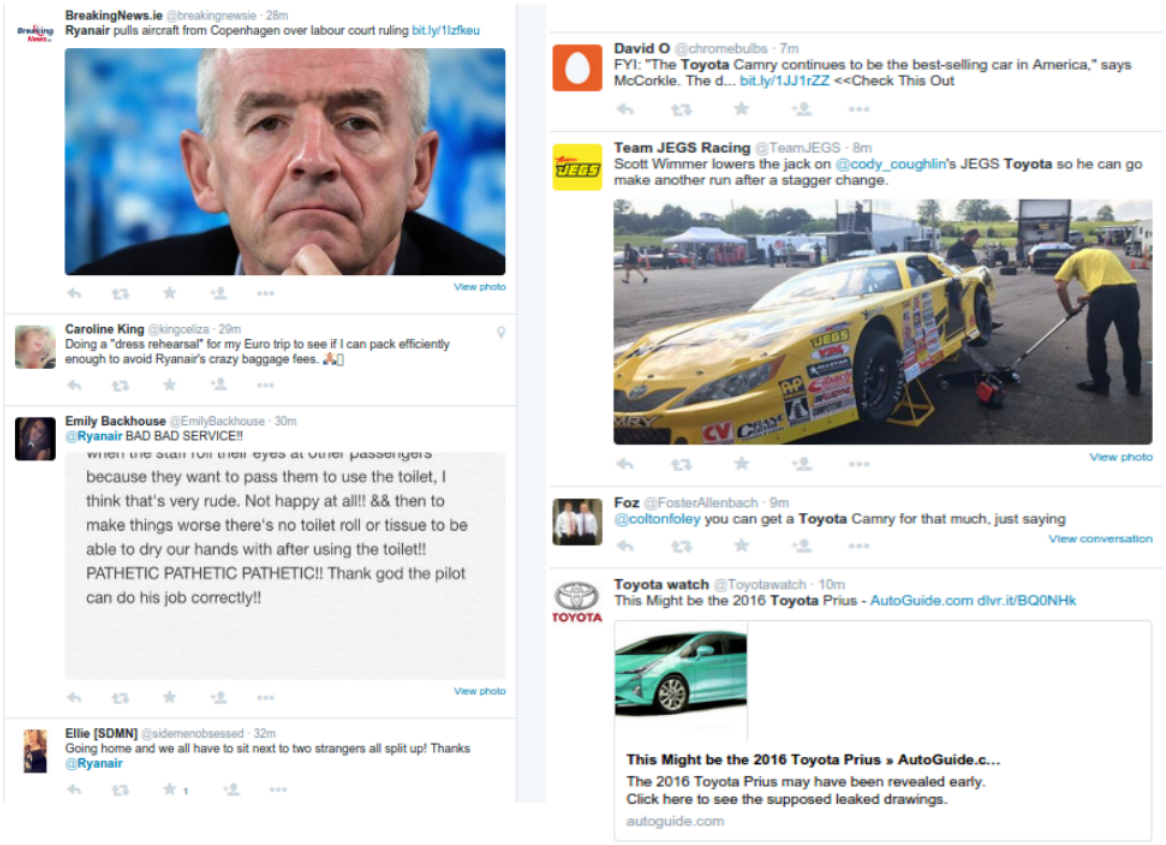

Figure 5.1: Examples of Tweets Expressing Opinions on Entity "Ryanair" (left) and Entity "Toyota" (right)

5.1 Introduction to Online Reputation Management

The area of "reputation management", emanating from the domain of "public relations", is concerned with managing the influence of an individual's or business's reputation [67]. Studies have concluded that it is a driving force behind Fortune 500 corporate public relations since the beginning of the 21st century [106]. It essentially comprises 1) monitoring the reputation of an entity[2], and 2) addressing content potentially damaging to the reputation of an entity.

With the growing popularity of social media the meaning of reputation management has shifted to online portals such as blogs, forums, opinion sites, and social

[1]More specifically, CLEF 2013 and CLEF 2014.

[2]In the context of reputation management, an entity may refer to a celebrity, company, organization or brand.

networks. Companies are increasingly making use of social media for their promotion and marketing. At the same time social media users voice their opinions about various entities/brands (e.g., musicians, movies, companies) [53, 79]. This has recently given birth to "online reputation management" within the marketing domain where automated and semi-automated methods facilitate monitoring reputation of entities instead of relying completely on the manual reputation management by an expert (or a group of experts) as was traditionally done [8]. Twitter serves as the most popular social media source for online reputation management [109] due to its nature of enabling fast dissemination of information. Figure 5.1 shows a typical example of people expressing opinions on the airline "Ryanair" and the automotive manufacturer "Toyota."

5.2 Significant Subtasks within Online Reputation Management

This thesis explores tasks in the context of CLEF 2013 RepLab and CLEF 2014 RepLab evaluation campaigns. The campaigns were organized as CLEF evaluation tasks for three consecutive years i.e. RepLab 2012 [10], RepLab 2013 [8] and RepLab 2014 [9] where teams were given a set of entities and for each entity a set of tweets were provided. The challenge was to perform mining over these tweets for various tasks relevant to online reputation management of the given entities. We present in this section a brief overview of two tasks we deal with, namely the filtering task and the reputation dimensions classification task.

5.2.1 Filtering Task

A significant subtask within online reputation management arises in the form of identifying whether or not a particular social media post, and in our case, a tweet is related to the entity. This subtask is called the filtering task where tweets are classified as either relevant or irrelevant for an entity. For example, a term "Apple" appears in a tweet and the tweet is to be marked as either relevant or irrelevant for the company "Apple". This task can be seen as an application of "entity name disambiguation" due to need to distinguish between the fruit "Apple" and the company "Apple" when the term "Apple" appears.

More specifically, given a set of tweets collected after issuing the entity name as a query, the task is to determine which of the tweets are related to the entity and which are not. As a motivating example, consider a query 'apple' issued using the Twitter

search API which retrieves the tweets containing the term 'apple'; the system has to determine if the tweet refers to the entity 'Apple Inc.' which is a company founded by 'Steve Jobs' or not. So the decision of the system is to mark a tweet as 'relevant' or 'irrelevant'.

5.2.2 Reputation Dimensions Classification Task

An entity has various *aspects* or *dimensions* that affect its reputation. As an example, consider the following scenarios:

- A smartphone company releasing a new phone and creating hype around the product release.
- A pharmaceutical company in trouble due to release of a new drug without adequate testing.

In the first example above, the company's *"products/services"* are under discussion while in the second example the company's *governance* aspect is being examined.

Keeping these different dimensions in view, the task of reputation dimensions classification was first introduced within RepLab 2014 [9]. The task involves classification of tweets according to the reputation dimensions which requires identification of various aspects significant to a company's reputation and Table 5.1 shows the standard dimensions used[3]. Basically, the task involves multi-class classification where given a tweet about an entity of interest and a set of reputation dimensions (in this case the ones shown in Table 5.1), the goal is to automatically classify the tweet to the single reputation dimension that the tweet relates.

Finally, a dimension known as "Undefined" was included by RepLab 2014 organizers in cases where a tweet fails to fall into any of the seven dimensions mentioned in Table 5.1.

5.3 Challenging Nature of Task

The nature of the tasks arising in the context of online reputation management are challenging and the fact that we aim to perform the task for tweets increases the difficulty of the task. This section discusses explicit and implicit challenges of the tasks so as to enable a proper understanding of the techniques proposed.

[3]Note that these are the standard dimensions provided by the Reputation Institute.

Dimension	Description
Products & Services	Products and services offered by the company or reflecting the consumers' satisfaction
Innovation	Innovativeness shown by the company, nurturing novel ideas and incorporating them into products
Workplace	Employees' satisfaction or the company's ability to attract, form and keep talented and highly qualified people
Citizenship	Company acknowledgement of community and environmental responsibility, including ethical aspects of the business: integrity, transparency, and accountability
Governance	The relationship between the company and the public authorities
Leadership	The leading position of the company
Performance	The company's long term business success and financial soundness

Table 5.1: Description of Reputation Dimensions of an Entity

5.3.1 Explicit Challenges

Some explicit challenges of the task as defined in CLEF are reported as follows:

- Tweets are multi-lingual i.e., generally in English and Spanish.
- There are four[4] broad types or domains of entities namely automotive, banking, universities, and music each with differing characteristics.
- Tweets often contain poor language. Issues relating to spelling mistakes and inaccurate grammar are very common.

5.3.2 Implicit Challenges

Some of the implicit challenges are listed as follows:

- Tweets are just 140 characters in length causing a user to provide information within a short window of text thereby limiting the surrounding context. Sometimes the user may use non-standard abbreviations which makes it even more

[4]All four were used in the filtering task while only automotive and banking were used for the reputation dimension classification task.

Figure 5.2: Examples of Tweets after GoDaddy Outage

challenging to make inferences. Figure 5.2 shows an example of tweets with a non-standard vocabulary after the outage of web hosting company "GoDaddy" on 10th September 2012[5]; note that the company name has been changed to NoDaddy due to the outage.

- Some of the tweets can be "spam" and yet refer to an entity but are not deemed to be relevant.
- Despite the organizers stating that the tweets will be in two languages, the nature of the query term causes some of the returned tweets to include tweets in some other languages.

[5] `http://techcrunch.com/2012/09/10/godaddy-outage-takes-down-millions-of-sites/`

- Some of the tweets contain a mix of different languages as the users tend to write in an informal way.
- A knowledge base can suffer from limited coverage problem and this can make it difficult to infer meanings for a non-famous entity or non-famous (rare or specific) information referring to an entity.

5.4 Overview of Our Approach

This section presents a brief overview of our approach for the filtering task and the reputation dimensions classification task. Fundamentally, the approach is aimed at enhanced context representation for tweets in order to filter them with respect to entities and/or reputation dimensions; this is done in an effort to address the second research question raised in Section 1.3 (Chapter 1). The strength of our approach consists of the exploitation of the encyclopaedic knowledge in Wikipedia which is an up-to-date and dynamic resource with extensive knowledge on various subjects as explained in Chapter 2.

5.4.1 Filtering Task

The task of filtering tweets is performed through supervised learning by training the classifier using the following feature types:

- Relatedness scores for several (entity-related) Wikipedia category taxonomies
- Topical scores corresponding to each tweet obtained via topic modelling
- Twitter-specific features obtained using the Twitter API[6]

The fundamental constituent of the technique is the Wikipedia-based features which make use of the Wikipedia category-article structure that describes the entity to obtain a suitable set of related terms corresponding to an entity. A few approaches relying on Wikipedia have been proposed in the literature related to the task of entity filtering. Among these approaches the work of Peetz et al. [164] has failed to show competitive results at the CLEF RepLab Filtering task; this approach is based on an established system defined by Meij et al. [141] for entity linking using the Wikipedia hyperlink structure (refer to Section 3.3.2 of Chapter 3 for a detailed description of the approach by Meij et al.). Peetz et al. utilise as a feature proposed by Meij et al.

[6] `https://dev.twitter.com`

namely commonness whereby candidate phrases for entities are topically identified by how often a particular phrase links to different entities inside the KB (e.g., "world cup" anchor text within Wikipedia referring to Wikipedia articles "FIFA World Cup", "World Cup (men's golf)", etc.). Moreover, their technique fundamentally constitutes an active learning system whereby some tweets are manually inspected during the learning process for updating the model and taking into account new labelled data. Despite the promise shown by active learning for the filtering task in context of online reputation management their technique fails to show reasonable performance, and this is primarily on account of hyperlink-based features which fail to provide maximum contextualization for entity disambiguation. We also experimented with one such approach with details in Appendix B; our system based on Wikipedia hyperlinks does not exhibit optimal performance whereas the one based on Wikipedia category-article structure that we explain in this shows a performance comparable to the one exposed by the best systems participating in the filtering task.

5.4.1.1 Baseline System for the Filtering Task

The baseline approach consists of tagging tweets (in the test set) with the same tags of the closer tweet in the (entity) training set according to the Jaccard word distance. In other words, the baseline system is a simple version of memory-based learning. Note that the baseline system was provided by CLEF RepLab 2013 organizers on account of its ease of use, and ability to exploit training data per entity.

5.4.2 Reputation Dimensions' Classification Task

The task of reputation dimensions classification is performed through supervised learning by training the classifier using the following feature types:

- Relatedness scores for several (reputation classes related) Wikipedia category taxonomies
- Statistical features which we further categorize into tweet-specific features, language-specific features, and word-occurrence features described in the following

The fundamental constituent of the technique is the Wikipedia-based features which make use of the Wikipedia category-article structure that describes a reputation dimension to obtain a suitable set of related terms corresponding to that dimension.

5.4.2.1 Baseline System for the Reputation Dimensions' Classification Task

Similar to the filtering task, the baseline system for the reputation dimensions' classification task was provided by the organizers of CLEF RepLab 2014. It essentially comprises a simple Bag-of-Words (BoW) classifier; the classifier used was Support Vector Machine with linear kernel and multiple classes were trained corresponding to each entity.

5.5 Methodology

In this section we present the proposed methodology that we have defined for the tasks of filtering and reputation dimensions' classification. Note that the first step of the proposed methodology comprises of the steps outlined in section 4.1 and section 4.2 of Chapter 4.

5.5.1 Filtering Task

The following subsections present an explanation of the features used for the filtering task as discussed in section 5.4.1.

5.5.1.1 Feature Set Based on Wikipedia Category-Article Structure

The preliminary step before generation of these features involves generation of candidate phrases as outlined in section 4.1 of Chapter 4 of this thesis. This is followed by the utilisation of relatedness score features that were explained in section 4.2 of Chapter 4 of this thesis. As we mentioned in those sections, the choice of category taxonomy is dependent upon the application scenario and is generally representative of the entity/concept under investigation, herein we describe the strategy for finding category taxonomies as follows:

- We fetch all the parent categories[7] and all sub-categories[8] to a depth of two of an entity's Wikipedia article.
- Using the training data we select the top category taxonomies as follows. First, we combine the training tweets of a single domain/entity[9] into one document,

[7]These are basically the categories of an entity's Wikipedia article i.e., categories at the depth zero from the Wikipedia article of an entity.

[8]These are basically entity-related categories at depth count of one and two.

[9]As explained in section 5.3.1 on the task's explicit challenges, a domain represents a particular business or organizational type e.g. automotives, banking, university etc.

and then we perform the process of variable-length phrase chunking (as explained in section 4.1 of Chapter 4) to extract candidate phrases. Using the Wikipedia articles in which the match occurs for candidate phrases, we extract the categories associated with these articles[10]. Using these categories (i.e., all categories derived from candidate phrases) as category taxonomies, we calculate relatedness scores for each phrase in a given tweet using equations 4.5 and 4.9 of Chapter 4 of this thesis. Finally, the category taxonomies which yield the highest scores for all tweets within the training set across a given domain are chosen as category taxonomies for that particular domain.

Listing 5.1 shows the pseudo-code for the above-explained category taxonomy selection process. Following are our main intuitions behind the choice of methods for the selection of Wikipedia category taxonomies:

- The parent category connected to an entity and its sub-categories to a depth count of two contain a significant amount of useful information pertaining to the entity of interest. Recall from Figure 4.2 and 4.3 how the parent category and related sub-categories contain significant terms such as "Steve Jobs" and "Rainbow Flag" respectively.

- Wikipedia categories with the top relatedness scores give a good representation of an entity-based frequent discussion in tweets and hence, tend to provide an effective set of features for the "filtering" task as shown in the section on experimental evaluations.

Using the category taxonomies chosen from above two steps we generate the feature set based on Wikipedia category-article structure. For each category taxonomy, we generate a score corresponding to $Depth_{significance}$((i.e., equation 4.6), $Cat_{significance}$ (i.e., equation 4.7), $Phrase_{significance}$ (i.e., equation 4.8), and *Relatedness* (i.e., equation 4.9) as the feature set.

Listing 5.1: Psuedo-Code for selecting category taxonomies

```
1 def main():
2  get_selectedTaxonomies(entity_Wiki_article) # main
       functionality
3
4 def generate_taxonomy(rootcat, d):
5  return (sub_cat(cat, 2), rootcat) # returns sub categories
       along with root
```

[10]Recall from section 2.2.6 of Chapter 2 that each article belongs to one or more categories.

```
6
7  def get_lst_taxonomies(relscores):
8   taxonomies = []
9   for sc, taxonomy in relscores:
10   taxonomies.append(taxonomy)
11  return taxonomies
12
13 def get_selectedTaxonomies(wiki_article):
14  selected_taxonomies = []
15  for pcat in categories(wiki_article):
16   selected_taxonomies.append(generate_taxonomy(pcat))
17
18  D = Merge all tweets belonging to single domain/entity #
       one big document of tweets
19  phrases = extractVariablePhrases(D) // Psuedo-code
       presented in Listing 4.1
20  cat_lst = set()
21  for p in phrases:
22   cat_lst.union_update(getCategories(p)) // since each
       phrase is a wiki article
23  relscores = []
24  for cat in cat_lst:
25    cat_t = generate_taxonomy(cat, 2)
26    score = relatedness(D, cat_t)
27    relscores.append([score, cat_t])
28
29  ordered_relscores = relscores order by score (descending
       order)
30  additional_taxonomies = get_lst_taxonomies(
       ordered_relscores[:100]) # select top-k taxonomies
31  selected_taxonomies.extend(additional_taxonomies)
32  return selected_taxonomies
```

5.5.1.2 Feature Set Based on Topic Modelling

A well-known topic modelling technique known as Latent Dirichlet Allocation (LDA for short) [24] is used for this set of features. LDA is an unsupervised machine learning technique aimed at identification of latent topics in large document collections. It is built on the "bag of words" approach with each document being treated as a vector of word counts and finally as an outcome of LDA, each document is represented as a probability distribution over some topics, while each topic is represented as a probability distribution over a number of words.

We trained LDA with 300 topics on each domain (music, automobile, etc) containing several entities, and the score (i.e., probability distribution) in each topic is

Category	Description
Tweet content features	Tweet character and word length
	Number of phrase markers in the tweet
	Does the tweet contain the entity name?
	Cosine similarity between the entity name and tweet
	Cosine similarity between the entity description (provided by task organizers) and tweet
	Does the tweet content contain url base address (provided by task organizers) of the entity?
	Number of hashtags in the tweet
	Total characters in all hashtags of the tweet
	Number of urls in the tweet
	Was the tweet translated into English?
	Number of stopwords in the tweet content
User information features	Friend_{count}
	Follower_{count}
	Whether or not the user holds a verified Twitter account
	Number of tweets by user
	Number of lists in which the user is present
	Number of tweets marked as favourite by the user
	Friend_{count}/(Follower_{count}+smoothing) + Status_{count}/ (Follower_{count}+smoothing)
	Cosine similarity between username and entity name
	Cosine similarity between username and domain category of entity (provided by task organizers)
	Cosine similarity between username and the entity description (provided by task organizers)
Mention features	Number of mentions in a tweet
	Total characters in all mentions
	Cosine similarity between mention and entity name
	Cosine similarity between mention and domain category of entity (provided by task organizers)

Table 5.2: Detailed Description of Twitter-Specific Feature Set

then utilised as a feature, and hence all topics make a feature set. The rationale for this is that the Wikipedia article titles cannot match all the terms and therefore, with the help of LDA we can include the influence of the remaining terms.

5.5.1.3 Twitter-Specific Feature Set

In this section we present the set of features that are specific to the nature of Twitter. We categorize these features into three categories: *tweet content features*, *user information features*, and *mention features*.

Tweet content features: These are features derived from the content of tweets.

User information features: These are features derived from the profile information of the Twitter user who is the producer of the tweet.

Mention features: These are features derived from the profile information of the users that are mentioned in a tweet.

Table 5.2 shows the detailed description of these features. Note that the profile information features for the users who produce a tweet or are mentioned in a tweet are utilised only in cases when the profile information is available (i.e., in cases where the user profile is public and has not been deleted or blocked from Twitter). The use of Twitter-specific features helps enriching the machine learning model which in turn improves the classification accuracy. This is on account of specific attributes of Twitter whereby organizations and individuals use it differently. Moreover, each entity differs from the other in terms of its presence on Twitter; as an example certain Spanish banks from within the dataset have an overly active Twitter presence due to their sponsorship of football clubs.

Note that Table 5.2 to the best of our knowledge shows an exhaustive set of Twitter-specific features and the selection of these features was motivated by standard works on tweet classification from within the literature [20, 52, 158, 165, 176, 197, 223]. Specifically, classification of tweets for purposes of marketing [158, 197, 223] rely on all the above classes of features namely tweet content features, user information features, and mention features. Feature selection shows the importance of every feature, and hence, we utilise all of them for the purpose of our experimental evaluations.

5.5.2 Reputation Dimensions Classification Task

Recall from Section 5.2.2 that the reputation dimensions classification task requires multi-class classification of tweets into pre-defined classes that reflect which aspect of an entity's reputation is under discussion. Again, Table 5.1 shows the standard dimensions used. In the subsections that follow we present an explanation of the features used for the reputation dimensions classification task as discussed in section 5.4.2.

5.5.2.1 Feature Set Based on Wikipedia Category-Article Structure

Similar to the filtering task, the preliminary step involves generation of candidate phrases (again, section 4.1 of Chapter 4 of this thesis) followed by the utilisation of the relatedness score features (again, section 4.2 of Chapter 4 of this thesis). Since the choice of category taxonomy is dependent upon the application scenario and is generally representative of the entity/concept under investigation, herein we describe the strategy for finding category taxonomies as follows.

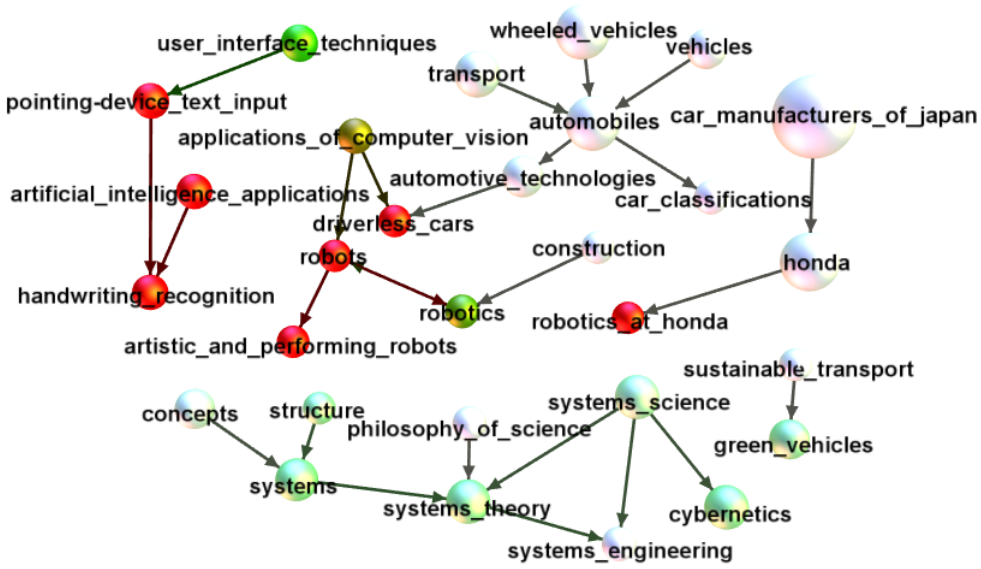

Figure 5.3: Wikipedia Categories for Reputation Dimension "Innovation" (from Training Data) for Automotive Domain

Using the training data we select the top category taxonomies by first combining the training tweets of a single reputation dimension into one document, and then we perform the process of variable-length (as explained in section 4.1) to extract candidate phrases. Each matched Wikipedia article corresponding to a candidate phrase[11] belongs to one or more Wikipedia categories. We maintain a voting count corresponding to each Wikipedia category through which the probability of a Wikipedia category belonging to a particular reputation dimension is calculated, and finally the *top-k* Wikipedia categories with highest probabilities are used as category taxonomies. To aid the reader in visualizing the obtained categories, we plot the obtained categories using Gephi[12] whereby probabilities are plotted to select the Wikipedia categories most closely related to a given reputation dimension.

Figure 5.3 illustrates the graph of Wikipedia categories corresponding to the reputation dimension of "Innovation" for the automotive domain, and Figure 5.4 illustrates the graph of Wikipedia categories corresponding to the reputation dimension of "Innovation" for the banking domain. The red-colored nodes in these Figures represent the Wikipedia categories that occur in a particular dimension with a probability of 1.0, the white-colored nodes represent a probability of 0.0, and the various

[11]Recall from section 4.1 that the final step in extraction of candidate phrases corresponds to matching with Wikipedia article titles.

[12]`http://gephi.github.io`

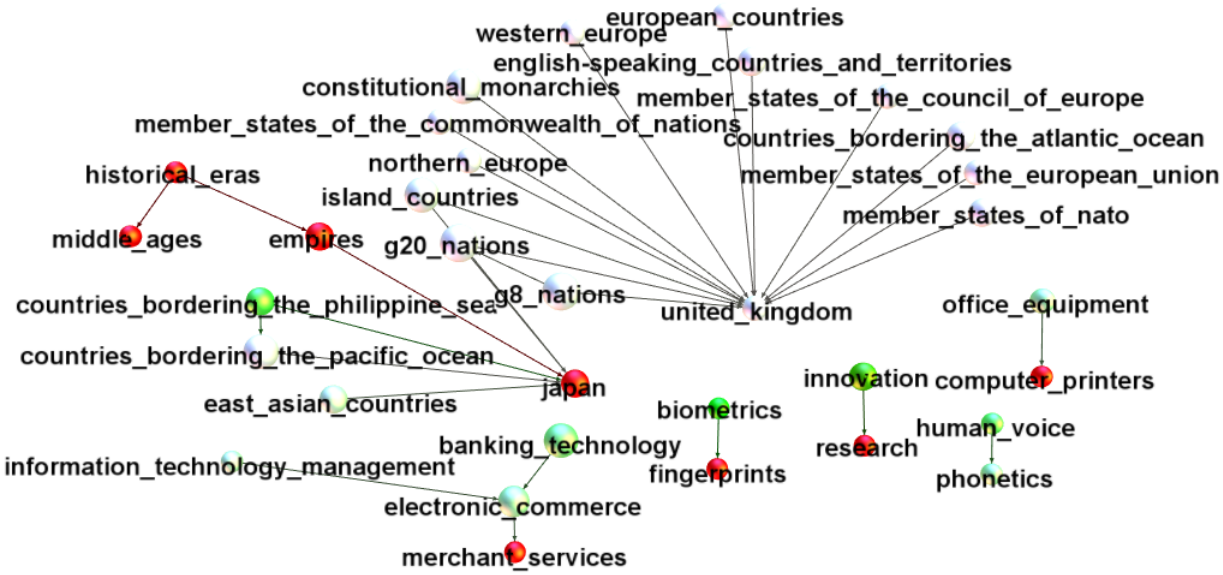

Figure 5.4: Wikipedia Categories for Reputation Dimension "Innovation" (from Training Data) for Banking Domain

green-colored nodes represent probabilities around 0.5. In addition, the size of node indicates the number of times a category was observed in the dataset.

Using the category taxonomies representing the highest probabilities we generate the feature set based on Wikipedia category-article structure. For each category taxonomy we generate a score corresponding to *Relatedness* (i.e., equation 4.9) as the feature.

5.5.2.2 Tweet-Specific Feature Set

We used four tweet-specific features that relate to how a tweet is written. They are: (1) presence of hashtag (#tag); (2) presence of user mention (some_user); (3) presence of url in a tweet; (4) language of the tweet (i.e., English or Spanish).

5.5.2.3 Language-Specific Feature Set

We used three language-specific features that relate to various aspects of reputation dimension for a brand/entity. They are: (1) occurrence of a percentage symbol in a tweet; (2) occurrence of currency symbol in a tweet; (3) proportion of common-noun POS tags, proper-noun POS tags, adjective POS tags, and verb POS tags in a tweet.

5.5.2.4 Word-Occurrence Feature Set

We used two word-occurrence features of which the first checks for the presence of other entity names of same domain; note that products and services dimension contains a lot of tweets whereby other entities are mentioned in the tweet. The second feature first counts the number of times a word occurs in a given dimension for different entities (i.e., checks for word occurrence in 20 entities of automotive domain, and 11 entities of banking domain) and if the number of occurrences is above an empirically-set threshold we add that particular word to our dictionary of dimension terms. The number of dimension terms present are then used as features.

5.6 Experimental Evaluations

This section describes the experimental procedure that we undertake to demonstrate the effectiveness of the proposed methods. First, we present details of experimental data and environment and finally, we present the experimental results.

5.6.1 Dataset and Environment

We use the dataset provided by CLEF 2013 RepLab filtering task [8][13] which basically comprises a collection of tweets. We do not utilise the RepLab 2012 dataset on account of its representativeness whereby the *unknown-entity* scenario is addressed i.e. the entity of interest is represented as canonical name and a representative URL (e.g., the entitys homepage) but no entity-specific training data is available. Therefore, supervised models have to learn from data associated to other similar entities, and this scenario rarely arises in a real-world online reputation management setting [191].

The Wikipedia data is accessed using the WikiMadeEasy API[14] as this is an operational requirement of the proposed methodology.

5.6.1.1 Twitter Dataset

We use the dataset provided by CLEF 2013 RepLab task organizers which is a multilingual collection of tweets (i.e., 20.3% Spanish tweets and 79.7% English tweets). The corpus contains tweets referring to a set of 61 entities from four domains; automotive, banking, university, and music. The filtering task utilised tweets from all four domains

[13]Note that CLEF 2014 RepLab Reputation Dimensions' Classification task also utilised the same dataset.

[14]http://bit.ly/1eMADG9, it is a custom made Wikipedia API.

	All	Automotives	Banking	University	Music
Entities	61	20	11	10	20
Training No. Tweets	45,679	15,123	7,774	6,960	15,822
Test No. Tweets	96,848	31,785	16,621	14,944	33,498
No. Tweets EN	113,544	38,614	16,305	20,342	38,283
No. Tweets ES	28,983	8,294	8,090	1,562	11,037

Table 5.3: RepLab 2013 Dataset Details

whereas the reputation dimensions' classification task utilised tweets from automotive and banking domain.

The tweets were gathered by organizers of the task by issuing the entity's name as the query. For each entity roughly 2300 tweets were collected with the first 750 constituting the training set, and the rest serving as the test set. Table 5.3 shows the statistics of the dataset.

5.6.1.2 Wikipedia

The data for Wikipedia category-article structure is obtained through a custom Wikipedia API that has pre-indexed Wikipedia data and hence, it is computationally fast[15]. The API has been developed using the DBPedia [22] dumps and it is a programmer-friendly API enabling developers and researchers to mine the huge amount of knowledge encoded within the Wikipedia structure. A more detailed description of the API which we name as WikiMadeEasy API appears in Appendix A of this thesis.

Our previous work for addressing the CLEF RepLab 2013 filtering task made use of several features instead of direct entity linking, and both of them outperformed Peetzs approach at RepLab 2013 [171]. An overview of previous work is presented in Appendix B of this thesis. Here, we show that the entity filtering task can be effectively addressed by approaches relying on Wikipedia; in fact the new enhanced approach to the filtering task shows a performance comparable to the one exposed by the best systems at RepLab 2013 as we will show in the evaluations reported in this chapter.

[15]http://bit.ly/1eMADG9, we aim to release the API as an open source Wikipedia tool to facilitate other researchers.

5.6.2 Experimental Setup

We first describe the experimental setup for the filtering task followed by a description of experimental setup for reputation dimensions' classification task.

5.6.2.1 Filtering Task

In order to test the effectiveness of the various feature sets described in Section 5.5.1, we design the experiments in a way that involves testing various combinations of the feature set. As we described in Section 5.5.1 the feature sets fall into three categories as follows: those based on Wikipedia category-article structure (from here on denoted as $Wiki_{specific}$), those based on topic modelling (from now on denoted as $Topic_{specific}$) and those that are Twitter-specific (from now on denoted as $Twitter_{specific}$). We utilise the three sets of features in various combinations performing three sets of experiments as follows:

1. In the first set of experiments we use each of the feature sets $Wiki_{specific}$, $Topic_{specific}$ and $Twitter_{specific}$ individually in order to assess the contribution of each class of features in isolation.

2. In the second set of experiments we use each possible pair of features together resulting in the following combinations

   - $Wiki_{specific}$ and $Topic_{specific}$
   - $Wiki_{specific}$ and $Twitter_{specific}$
   - $Topic_{specific}$ and $Twitter_{specific}$

3. In the third set of experiments we use the three feature sets $Wiki_{specific}$, $Topic_{specific}$ and $Twitter_{specific}$ together.

The above-mentioned feature set combinations are first used in conjunction with a random forest classifier by training separately over the four domains and the given entities. The motivation for distinguishing between training over all tweets within a domain and within an entity separately is to capture the difference between various domains and within various entities in an attempt to discover the most useful training method. There are few shared characteristics and few differentiating characteristics between various entities in a domain; as an example the entities within the university domain distributed as part of the CLEF RepLab 2013 task possess some common characteristics (e.g., all of them contain different faculties of knowledge such as Faculty

of Science, Faculty of Humanities etc.) and some differences arise (e.g., in terms of locations of universities along with their rankings).

5.6.2.2 Reputation Dimensions' Classification Task

Using the feature sets described in 5.5.2, we train a random forest classifier over the training data and then use it to predict labels for the test data. We perform three machine learning runs as follows:

1. For the first run, we use only Wikipedia-based features of section 5.5.2 whilst training a random forest classifier per-domain i.e. combining all tweets related to a particular domain into one training and one test set

2. For the second run, we use only the additional features of section 5.5.2 whilst training a random forest classifier per-domain i.e. combining all tweets related to a particular domain into one training and one test set

3. For the third run, we use all features i.e. both Wikipedia-based features and additional features of section 5.5.2 whilst training a random forest classifier per-domain i.e. combining all tweets related to a particular domain into one training and one test set

5.6.3 Experimental Results for Filtering Task

For the filtering task, we utilise the evaluation measures proposed by the organizers of the task which we earlier introduced in equations 2.9, 2.10 and 2.11 of section 2.1.4 (Chapter 2). Furthermore, we also report the measures of Precision (equation 2.1 in Chapter 2), Recall (equation 2.2 in Chapter 2) and F-Measure (equation 2.3 in Chapter 2) for relevant entities and irrelevant entities separately denoted by Precision_R, Recall_R, F-Measure_R, Precision_I, Recall_I, and F-Measure_I respectively.

Tables 5.4-5.9 present the results of our experiments with each of the individual feature sets with the difference being that Tables 5.4-5.6 present the results with training undertaken in a per-entity manner while Tables 5.7-5.9 present the results training undertaken per-domain. The results clearly demonstrate the superior performance of $Wiki_{specific}$ features corresponding to both per-entity and per-domain training thereby showing the power of the Wikipedia category-article structure for the filtering task at hand. The $Twitter_{specific}$ features show second best performance which confirms the fact that twitter-specific features are important over twitter for sharing information, while $Topic_{specific}$ shows the least performance.

Setting	**Precision**$_R$	**Recall**$_R$	**F-Measure**$_R$	**Precision**$_I$	**Recall**$_I$	**F-Measure**$_I$	**Sensitivity**	**Reliability**	**F-Measure**
Automotives	0.9207	0.8878	0.8987	0.7883	0.4953	0.5200	0.7268	0.3970	0.4496
Banking	0.9400	0.8740	0.9006	0.8901	0.6521	0.6316	0.8390	0.5339	0.5483
University	0.8929	0.7595	0.7970	0.8308	0.6670	0.6754	0.4382	0.7410	0.51348
Music	0.9710	0.9768	0.9733	0.8341	0.4202	0.4340	0.4050	0.8124	0.4201
Average	0.9361	0.8935	0.9068	0.8286	0.5271	0.5374	0.4311	0.7774	0.4683

Table 5.4: Evaluation Results on Test Set for $Wiki_{specific}$ Features and per-Entity Training

Setting	**Precision**$_R$	**Recall**$_R$	**F-Measure**$_R$	**Precision**$_I$	**Recall**$_I$	**F-Measure**$_I$	**Sensitivity**	**Reliability**	**F-Measure**
Automotives	0.8605	0.8688	0.8593	0.7429	0.3066	0.3118	0.2032	0.6438	0.2278
Banking	0.8117	0.8111	0.81046	0.8023	0.5356	0.5324	0.3859	0.6546	0.3845
University	0.7293	0.5567	0.5546	0.6732	0.5730	0.5553	0.1804	0.4851	0.2123
Music	0.9589	0.9734	0.9653	0.7822	0.2320	0.2249	0.2168	0.7590	0.2124
Average	0.8625	0.8415	0.8353	0.7551	0.3671	0.3630	0.2369	0.6575	0.2485

Table 5.5: Evaluation Results on Test Set for $Topic_{specific}$ Features and per-Entity Training

Setting	**Precision**$_R$	**Recall**$_R$	**F-Measure**$_R$	**Precision**$_I$	**Recall**$_I$	**F-Measure**$_I$	**Sensitivity**	**Reliability**	**F-Measure**
Automotives	0.8903	0.8881	0.8769	0.8958	0.4352	0.4764	0.3421	0.8072	0.3940
Banking	0.8976	0.8954	0.8929	0.9432	0.5702	0.5730	0.4795	0.8558	0.4860
University	0.8337	0.5803	0.6071	0.7684	0.6087	0.5857	0.2088	0.6430	0.2761
Music	0.9651	0.9748	0.9687	0.9228	0.3237	0.3394	0.3068	0.8936	0.3269
Average	0.9069	0.8674	0.8657	0.8923	0.4514	0.4668	0.3334	0.8174	0.3693

Table 5.6: Evaluation Results on Test Set for $Twitter_{specific}$ Features and per-Entity Training

Setting	**Precision**$_R$	**Recall**$_R$	**F-Measure**$_R$	**Precision**$_I$	**Recall**$_I$	**F-Measure**$_I$	**Sensitivity**	**Reliability**	**F-Measure**
Automotives	0.9148	0.8783	0.8923	0.6682	0.6088	0.5837	0.5060	0.6038	0.5078
Banking	0.9141	0.8513	0.8752	0.4765	0.6640	0.4694	0.5257	0.4033	0.3702
University	0.8300	0.7593	0.7743	0.7462	0.7096	0.7031	0.4914	0.6047	0.5279
Music	0.9650	0.9770	0.9670	0.3614	0.4388	0.3150	0.4261	0.3363	0.3027
Average	0.9172	0.8863	0.8953	0.5459	0.5795	0.4946	0.4810	0.4800	0.4190

Table 5.7: Evaluation Results on Test Set for $Wiki_{specific}$ Features and per-Domain Training

Setting	$\textbf{Precision}_R$	$\textbf{Recall}_R$	$\textbf{F-Measure}_R$	$\textbf{Precision}_I$	$\textbf{Recall}_I$	$\textbf{F-Measure}_I$	Sensitivity	Reliability	F-Measure
Automotives	0.8045	0.9262	0.8516	0.4374	0.2576	0.2543	0.2335	0.3064	0.2095
Banking	0.7138	0.9198	0.75986	0.3655	0.2909	0.1497	0.2700	0.1357	0.1005
University	0.5846	0.6490	0.5696	0.5500	0.5630	0.5125	0.3639	0.2290	0.2708
Music	0.9431	0.9962	0.9662	0.4173	0.1490	0.0586	0.1482	0.3807	0.05718
Average	0.7976	0.9026	0.8264	0.4363	0.2780	0.2136	0.2335	0.2873	0.1500

Table 5.8: Evaluation Results on Test Set for $Topic_{specific}$ Features and per-Domain Training

Setting	$\textbf{Precision}_R$	$\textbf{Recall}_R$	$\textbf{F-Measure}_R$	$\textbf{Precision}_I$	$\textbf{Recall}_I$	$\textbf{F-Measure}_I$	Sensitivity	Reliability	F-Measure
Automotives	0.8362	0.9775	0.8926	0.7359	0.4249	0.4744	0.4138	0.5916	0.4358
Banking	0.8621	0.8969	0.8556	0.5563	0.5764	0.4849	0.4832	0.4317	0.4058
University	0.7103	0.5675	0.5938	0.5981	0.7567	0.6391	0.3966	0.3710	0.3450
Music	0.9470	0.9904	0.9653	0.8148	0.2523	0.2884	0.2480	0.7674	0.2852
Average	0.8566	0.9000	0.8608	0.7068	0.4500	0.4423	0.3691	0.5843	0.3661

Table 5.9: Evaluation Results on Test Set for $Twitter_{specific}$ Features and per-Domain Training

Tables 5.10-5.15 present the results of our experiments with two of the feature sets from the total of three feature sets. Tables 5.10 and 5.13 show the results for the combination of the $Wiki_{specific}$ and the $Topic_{specific}$ features corresponding to per-entity and per-domain training respectively. Tables 5.11 and 5.14 show the results for the combination of the $Wiki_{specific}$ and the $Twitter_{specific}$ features corresponding to per-entity and per-domain training respectively. Tables 5.12 and 5.15 show the results for the combination of $Topic_{specific}$ and $Twitter_{specific}$ features corresponding to per-entity and per-domain training respectively.

Setting	$\textbf{Precision}_R$	$\textbf{Recall}_R$	$\textbf{F-Measure}_R$	$\textbf{Precision}_I$	$\textbf{Recall}_I$	$\textbf{F-Measure}_I$	Sensitivity	Reliability	F-Measure
Automotives	0.9207	0.8878	0.8987	0.7883	0.4953	0.5200	0.7268	0.3970	0.4496
Banking	0.9400	0.8740	0.9006	0.8901	0.6521	0.6316	0.8390	0.5339	0.5483
University	0.8929	0.7595	0.7970	0.8308	0.6670	0.6754	0.4382	0.7410	0.51348
Music	0.9710	0.9768	0.9733	0.8341	0.4202	0.4340	0.4050	0.8124	0.4201
Average	0.9361	0.8935	0.9068	0.8286	0.5271	0.5374	0.4311	0.7774	0.4683

Table 5.10: Evaluation Results on Test Set for $Wiki_{specific}$ and $Topic_{specific}$ Combined Features with per-Entity Training

Setting	Precision_R	Recall_R	F-Measure_R	Precision_I	Recall_I	F-Measure_I	Sensitivity	Reliability	F-Measure
Automotives	0.9391	0.8827	0.9014	0.8697	0.5322	0.5556	0.4234	0.8195	0.4867
Banking	0.9423	0.8851	0.9075	0.8955	0.6516	0.6344	0.5436	0.8453	0.5563
University	0.9006	0.7560	0.7949	0.8535	0.6769	0.6879	0.4428	0.7690	0.5244
Music	0.9743	0.9800	0.9769	0.8951	0.4210	0.4374	0.4071	0.8749	0.4243
Average	0.945	0.8943	0.9098	0.8800	0.5410	0.5527	0.4429	0.8340	0.4849

Table 5.11: Evaluation Results on Test Set for $Wiki_{specific}$ and $Twitter_{specific}$ Combined Features with per-Entity Training

Setting	Precision_R	Recall_R	F-Measure_R	Precision_I	Recall_I	F-Measure_I	Sensitivity	Reliability	F-Measure
Automotives	0.8903	0.8882	0.8769	0.8958	0.4352	0.4764	0.3421	0.8072	0.3940
Banking	0.8976	0.8954	0.8929	0.9432	0.5702	0.5730	0.4795	0.8558	0.4860
University	0.8337	0.5803	0.6071	0.7684	0.6087	0.5857	0.2088	0.6430	0.2761
Music	0.9651	0.9748	0.9687	0.9228	0.3237	0.3394	0.3068	0.8936	0.3269
Average	0.9069	0.8674	0.8657	0.8923	0.4514	0.4668	0.3334	0.8174	0.3693

Table 5.12: Evaluation Results on Test Set for $Topic_{specific}$ and $Twitter_{specific}$ Combined Features with per-Entity Training

Setting	Precision_R	Recall_R	F-Measure_R	Precision_I	Recall_I	F-Measure_I	Sensitivity	Reliability	F-Measure
Automotives	0.9148	0.8783	0.8923	0.6682	0.6088	0.5837	0.5060	0.6038	0.5078
Banking	0.9141	0.8513	0.8752	0.4765	0.6640	0.4694	0.5257	0.4033	0.3702
University	0.8300	0.7593	0.7743	0.7462	0.7096	0.7031	0.4914	0.6047	0.5279
Music	0.9650	0.9770	0.9670	0.3614	0.4388	0.3150	0.4261	0.3363	0.3027
Average	0.9172	0.8863	0.8953	0.5459	0.5795	0.4946	0.4810	0.4800	0.4190

Table 5.13: Evaluation Results on Test Set for $Wiki_{specific}$ and $Topic_{specific}$ Combined Features with per-Domain Training

Setting	Precision_R	Recall_R	F-Measure_R	Precision_I	Recall_I	F-Measure_I	Sensitivity	Reliability	F-Measure
Automotives	0.9367	0.8763	0.8968	0.7089	0.6233	0.6125	0.5104	0.6590	0.5405
Banking	0.9458	0.8524	0.8900	0.4838	0.6907	0.4844	0.5498	0.4379	0.3985
University	0.8362	0.7672	0.7806	0.8200	0.7158	0.7404	0.4979	0.6789	0.5652
Music	0.9706	0.9829	0.9763	0.4227	0.4507	0.3641	0.4406	0.4008	0.3515
Average	0.9330	0.8891	0.9026	0.5927	0.5940	0.5289	0.4926	0.5377	0.4570

Table 5.14: Evaluation Results on Test Set for $Wiki_{specific}$ and $Twitter_{specific}$ Combined Features with per-Domain Training

Setting	Precision_R	Recall_R	F-Measure_R	Precision_I	Recall_I	F-Measure_I	Sensitivity	Reliability	F-Measure
Automotives	0.8362	0.9775	0.8926	0.7359	0.4249	0.4744	0.4138	0.5916	0.4358
Banking	0.8621	0.8969	0.8556	0.5563	0.5764	0.4849	0.4832	0.4317	0.4058
University	0.7103	0.5675	0.5938	0.5981	0.7567	0.6391	0.3966	0.3710	0.3450
Music	0.9470	0.9904	0.9653	0.8148	0.2523	0.2884	0.2480	0.7674	0.2852
Average	0.8566	0.9000	0.8608	0.7068	0.4500	0.4423	0.3691	0.5843	0.3661

Table 5.15: Evaluation Results on Test Set for $Topic_{specific}$ and $Twitter_{specific}$ Combined Features with per-Domain Training

Setting	Precision_R	Recall_R	F-Measure_R	Precision_I	Recall_I	F-Measure_I	Sensitivity	Reliability	F-Measure
Automotives	0.9391	0.8857	0.9033	0.8723	0.5341	0.5601	0.4283	0.8219	0.4920
Banking	0.9434	0.8884	0.9097	0.8969	0.6514	0.6351	0.5465	0.8475	0.5588
University	0.9006	0.7560	0.7949	0.8535	0.6769	0.6879	0.4428	0.7690	0.5244
Music	0.9745	0.9795	0.9767	0.8946	0.4213	0.4370	0.4070	0.8747	0.4237
Average	0.9452	0.8957	0.9108	0.8810	0.5417	0.5542	0.4450	0.8351	0.4870

Table 5.16: Evaluation Results on Test Set for $Wiki_{specific}$, $Topic_{specific}$, and $Twitter_{specific}$ Features with per-Entity Training and RF Classifier

As can be seen from Tables 5.11 and 5.14 the combination of features $Wiki_{specific}$ and $Twitter_{specific}$ show the best performance with per-entity training outperforming the per-domain setting. This fits well into what was observed in the individual feature set (i.e., Tables 5.4-5.9) where $Wiki_{specific}$ and $Twitter_{specific}$ performed best and second best respectively. Note that the feature set $Topic_{specific}$ does not boost the performance at all (with the results for the combination of $Wiki_{specific}$ and $Topic_{specific}$ along with $Twitter_{specific}$ and $Topic_{specific}$ being the same).

Finally, Tables 5.16-5.19 show the results with all three feature sets combined and with multiple machine learning algorithms in order to confirm the general validity of the proposed method. We experiment with a random forest classifier (RF), naive bayes classifier (NB), gradient boost regression trees classifier (GBRT) and extremely randomized trees classifier (ERT). It can be seen that RT and ERT perform comparable, followed by GBRT and NB. This confirms that the nature of the problem which actually depends upon the feature set (e.g., apple can be a fruit if it appears with words like mango, oranges) which NB over simplifies by making an assumption of independence.

Table 5.20 also reports the statistical significance of the results for all domains (averaged) over the baseline with the various machine learning algorithms. Note

Setting	Precision_R	Recall_R	F-Measure_R	Precision_I	Recall_I	F-Measure_I	Sensitivity	Reliability	F-Measure
Automotives	0.9026	0.7749	0.8211	0.5574	0.5353	0.4492	0.3540	0.5025	0.3463
Banking	0.8600	0.8304	0.8399	0.6683	0.6177	0.5917	0.4733	0.5530	0.4627
University	0.5944	0.6037	0.5820	0.5786	0.5402	0.5227	0.2723	0.2774	0.2457
Music	0.9614	0.9235	0.9387	0.4988	0.3133	0.2586	0.2668	0.4850	0.2356
Average	0.8637	0.8056	0.8238	0.5617	0.4782	0.4245	0.3335	0.4690	0.3145

Table 5.17: Evaluation Results on Test Set for $Wiki_{specific}$, $Topic_{specific}$, and $Twitter_{specific}$ Features with per-Entity Training and NB Classifier

Setting	Precision_R	Recall_R	F-Measure_R	Precision_I	Recall_I	F-Measure_I	Sensitivity	Reliability	F-Measure
Automotives	0.9032	0.8934	0.8963	0.6365	0.4786	0.4924	0.3938	0.5666	0.4209
Banking	0.8953	0.9059	0.8995	0.5075	0.6177	0.4227	0.5408	0.4185	0.3407
University	0.8345	0.7487	0.7731	0.7598	0.6859	0.6670	0.4659	0.6259	0.4865
Music	0.9704	0.9720	0.9710	0.5429	0.3775	0.2953	0.3606	0.5221	0.2811
Average	0.9125	0.8977	0.9011	0.6028	0.5045	0.4443	0.4213	0.5350	0.3714

Table 5.18: Evaluation Results on Test Set for $Wiki_{specific}$, $Topic_{specific}$, and $Twitter_{specific}$ Features with per-Entity Training and GBRT Classifier

Setting	Precision_R	Recall_R	F-Measure_R	Precision_I	Recall_I	F-Measure_I	Sensitivity	Reliability	F-Measure
Automotives	0.9330	0.8897	0.9046	0.8703	0.5366	0.5660	0.4367	0.8154	0.4983
Banking	0.9240	0.8931	0.9066	0.7992	0.6415	0.6280	0.5450	0.7343	0.5489
University	0.8974	0.7556	0.7953	0.8501	0.6687	0.6800	0.4351	0.7638	0.5173
Music	0.9747	0.9787	0.9764	0.7948	0.4219	0.4367	0.4069	0.7760	0.4235
Average	0.9392	0.8975	0.9106	0.8294	0.5395	0.5535	0.4462	0.7794	0.4860

Table 5.19: Evaluation Results on Test Set for $Wiki_{specific}$, $Topic_{specific}$, and $Twitter_{specific}$ Features with per-Entity Training and ERT Classifier

Classifier	Precision_R	Recall_R	F-Measure_R	Precision_I	Recall_I	F-Measure_I	Sensitivity	Reliability	F-Measure
RF	0.9452***	0.8957	0.9108**	0.8810***	0.5417**	0.5542*	0.4450*	0.8351***	0.4870**
NB	0.8637	0.8056*	0.8238**	0.5617*	0.4782**	0.4245*	0.3335*	0.4690**	0.3145***
GBRT	0.9125*	0.8977**	0.9011**	0.6028*	0.5045	0.4443*	0.4213**	0.5350*	0.3714***
ERT	0.9392***	0.8975	0.9106*	0.8294***	0.5395*	0.5535*	0.4462**	0.7794**	0.4860***

Note *p<.05, **p<.01, ***p<.001

Table 5.20: Average Evaluation Results on Test Set for $Wiki_{specific}$, $Topic_{specific}$, and $Twitter_{specific}$ Features with per-Entity Training using different classifiers

that we do not utilise support vector machines and artificial neural networks as our aim was to demonstrate the generality of the proposed classification technique and hence, we utilise the simplest classification algorithms for our task with the default parameters within the python-scikit library[16].

Table 5.21 presents a summary of the evaluation measures presented by the task organizers i.e. Reliability and Sensitivity [11] in comparison with the baseline system and best performing systems of the CLEF RepLab 2013 task. From this table it can seen that our system performs comparably to the best reported system. The high reliability[17] values for our approach again indicates the strength of Wikipedia in precisely identifying concepts relevant to a given entity. However, sensitivity[18] suffers on account of some tweets having too less of a context to able to have matches in Wikipedia article titles: as an example, our approach will fail to identify a tweet such as "I love Apple" as being relevant to entity "Apple Inc."

5.6.4 Experimental Results for Reputation Dimensions' Classification Task

Table 5.22 presents a snapshot of the official results for the filtering task of RepLab 2014, where CIRGIRDISCO is the name of our team. As can be seen from Table 5.22, out of a total of 8 participating teams in RepLab2014 reputation dimension classification task 4 teams outperform our best run. Our system shows good results for the evaluation measure of accuracy; however, the evaluation measures of precision and recall show an average performance and one reason for this is due to our training

[16]The only exception is random forest classifier for which we utilised 500 estimators since we have a decent machine.

[17]Recall from section 2.1.4 of Chapter 2 that reliability is basically product of precision of both relevant and irrelevant entities.

[18]Recall from section 2.1.4 of Chapter 2 that reliability is basically product of recall of both relevant and irrelevant entities.

Setting	Sensitivity	Reliability	F-Measure
Our Approach	**0.4450**	**0.8351**	**0.4870**
Baseline	0.4902	0.3200	0.3255
POPSTAR [184]	0.7288	0.4507	0.4885
SZTE [86]	0.5990	0.4444	0.4385
Previous Approach [171]	0.4164	0.6687	0.4485

Table 5.21: Comparison of Experimental Results for Systems in CLEF RepLab 2013 Task

Team	Accuracy	F-measure
uogTr_RD_4	0.7318	0.4735
DAE_RD_1	0.7231	0.3906
Lys_RD_1	0.7167	0.4774
SIBTEX_RD_1	0.7073	0.4057
CIRGIRDISCO_RD_3	0.7071	0.3012
CIRGIRDISCO_RD_2	0.6924	0.2386
Baseline	0.6222	0.4072
CIRGIRDISCO_RD_1	0.6073	0.3195

Table 5.22: Results of Reputation Dimensions' Classification Task of RepLab 2014

Entity (Domain)	Total Training Tweets	Irrelevant Tweets	Relevant Tweets
Adele (Music)	694	20	674
Jennifer Lopez (Music)	862	4	858
Led Zeppelin (Music)	908	0	908
Maroon 5 (Music)	738	0	738
Bankia (Banking)	760	19	741
Barclays (Banking)	747	1	746
HSBC (Banking)	797	6	791

Table 5.23: Proportion of Relevant and Irrelevant Tweets for Some Entities in Training Data

Sample	Innovation	Citizenship	Leadership	Workplace	Governance	Undefined	Performance	Products and Services
Training Data	313	2209	297	468	1303	2228	947	7898
Test Data	306	5027	744	1124	3395	4349	1598	15903

Table 5.24: Proportion of Relevant and Irrelevant Tweets for Some Entities in Training Data

and testing methods being applied over eight classes because we included the class "Undefined" in our training and testing supervised learning method whereas the RepLab 2014 organizers excluded this class. However, it was not clear in the task guidelines.

5.6.5 Discussion and Conclusion

This section presents an analysis of the strengths of the proposed methodology along with underlying issues with the given dataset that add to the complexity of the tasks at hand. We also present two limitations of our proposed methodology.

5.6.5.1 Analysis of our Proposed Methodology

In summary, classifying tweets into relevant or irrelevant for an entity or along various reputation dimensions is a challenging task with most of the challenges stemming from the nature of how text is written by Twitter users. In this section, we perform an analysis of our proposed methodology in an attempt to perform a detailed study of the effectiveness of the proposed features:

Domain	**F-Measure$_R$**	**F-Measure$_I$**	**F-Measure$_{RS}$**
Automotives	0.1115	0.3180	0.2746
Banking	0.1230	0.4105	0.3767
University	0.1689	0.2971	0.2169
Music	0.0391	0.4058	0.3974

Table 5.25: Standard Deviation of F-Measure$_R$, F-Measure$_I$, and F-Measure$_{RS}$ for Various Domains in Dataset

Lack of context in tweets: As mentioned in Section 5.3 the lack of context in tweets poses a serious challenge for the online reputation management tasks. This means that most approaches that utilise training over bag-of-words and term co-occurrence features fail to correctly classify relevant examples. As an example, consider the following tweets:

"Honda VTEC is very fantastic!!!"

"AFF Suzuki Cup. A terrific first half. Malaysia 0-0 Thailand = blood and thunder, guts galore and some quality football. Evenly balanced"

The first tweet is relevant for the entity "Honda" in the "automotives" domain which the baseline system incorrectly classified as irrelevant for the entity[19]. The second tweet pertaining to the entity "Suzuki" in the "automotives" domain is concerned with the "Citizenship" dimension

Here, *VTEC* is a system developed by the automotives company "Honda" to improve efficiency of a four-stroke internal combustion engine. Furthermore, *AFF Suzuki Cup* is a sports tournament organized and sponsored by "Honda" to foster community engagement and hence, the tweet concerns the reputation dimension of "Citizenship" (refer to Table 5.1). Simple textual features fail to classify this tweet as relevant while our approach is able to detect the relationship between "Honda" and "VTEC" along with the relationship between "AFF Suzuki Cup" and the reputation dimension of "Citizenship" through its use of the Wikipedia category-article structure. More specifically, the category taxonomies derived through the entity's Wikipedia article and through the tweet phrases matching Wikipedia articles' titles provide sufficient

[19]This example has been taken from the test dataset distributed by CLEF RepLab 2013 organizers.

context to enrich the tweets' representation and hence, make a correct prediction for the tweets.

As a further example, the following tweet within the "university" domain for the entity "Princeton University" is fairly complex and textual features alone fail to classify it as relevant. Our technique however is able to extract it as a relavant tweet for the entity "Princeton University", and this is on account of the Wikipedia article corresponding to "Princeton offense" which is linked to the Wikipedia category "Princeton University" within the Wikipedia category-article structure.

"Fuk mike brown n that weak ass Princeton offense wtf does Princeton offer besides smart ass mf they sports ain't talkin bout shit! #adios"

User-generated content: The informality of text in Twitter implies a range of ways in which users can compose tweets relating to different entities. The fact that user-generated content on Twitter is written in a non-standard manner adds to the complexity of the tasks.

Discrepancies between training and test data: The dataset for the CLEF RepLab 2013 Filtering Task is not balanced in terms of relevant/irrelevant tweets with the proportion not following a normal distribution. As a result, training is not performed in an optimal manner for some of the entities. Many entities within some of the domains (especially) have unbalanced training data as shown in Table 5.23. Similarly, the dataset for the CLEF RepLab 2014 Reputation Dimensions' Classification task is not balanced as shown in Table 5.24. Data balancing may be performed to even out the effects of such a discrepancy but that may not be reflective of a real-world online reputation management scenario where tweets are not balanced with either a great deal of noise for some entities or lots of relevant tweets for other entities[20].

Table 5.25 shows the standard deviation of F-Measure$_R$, F-Measure$_I$, and F-Measure$_{RS}$ of our system for the various domains in the dataset. As can be seen the evaluation measures Reliability and Sensitivity used by the organizers are too sensitive and do not capture the imbalance between the proportion of related/unrelated tweets. However, the standard deviation of F-Measure$_R$ is less and this demonstrates

[20]There are occasions when a certain entity becomes a trending topic on account of some event occurring becoming a topic of discussion in news media; such an event occurred during FIFA World Cup 2014 after the Dutch airline KLM posted a "racist" tweet in response to Mexico's defeat against Netherlands.

the robust nature of our proposed features. A somewhat high standard deviation of F-Measure$_I$, and F-Measure$_{RS}$ especially for the banking and music domain is due to most of the entities within these domains having very few irrelevant examples in the training data. Such imbalance hinders the performance of our proposed technique; however, despite this hindrance our system performs as good as the best system that participated in CLEF RepLab 2013 Filtering Task and it may perform even better if the training data is in a balanced proportion.

5.6.5.2 Limitations of our Proposed Methodology

We identified the following limitations of our proposed methodology:

- Some entities are not covered in Wikipedia and this is specifically the case for long-tail entities for which popularity emerges in a short period of time.
- The Wikipedia category-article structure contains some amount of noise in case of certain entities. As an example in case of domain of "musicians", certain entities are linked to Wikipedia articles about their parents, siblings etc. who do not have a direct relationship with the main business concerning the entity; this adds some imprecision within the filtering for these entities.

We discuss potential ways to overcome these limitations during discussion of future work in Chapter 8.

5.6.5.3 Conclusion

The experimental evaluations establish Wikipedia's strength as a significant encyclopaedic resource for the challenging tasks arising in the context of online reputation management. The relatedness score defined using Wikipedia category-article structure introduces a powerful semantic notion of linking n-grams in a tweet with the information relevant to an entity and/or reputation dimension under discussion as shown by the performance of the proposed approach.

5.7 Summary of the Chapter

In this chapter we presented two application scenarios arising in the context of "online reputation management" which basically emanates from the domain of "public relations" and is concerned with monitoring the reputation of entities online. Entities within reputation management represent brands, celebrities, businesses etc. The

application scenarios we deal with have emerged on account of Twitter emerging as a new online forum for expression of opinions with respect to an entity. The scenarios are 1) filtering tweets and identifying whether or not a given tweet is related to a certain entity, and 2) identifying the reputation dimension with which a certain tweet deals with. We deal with these challenging tasks through a set of features; among these features our Wikipedia-based semantic relatedness features described in Chapter 4 constitute the most significant ones. We also describe the task-specific techniques to identify the category taxonomies which are then utilised for the extraction of relatedness scores with respect to categories representing an entity. For the filtering task, parent categories and sub-categories to a depth count of two are utilised in addition to categories corresponding to those matched phrases within tweets that are able to generate highest relatedness scores. For the reputation dimensions classification task, a probabilistic approach is utilised on the basis of a voting count corresponding to the number of times a Wikipedia category occurs for a given reputation dimension. Finally, experimental evaluations demonstrate the richness of the proposed Wikipedia-based features. For the filtering task in particular the choice of machine learning algorithm does not influence the outcome by a large degree with other sets of features (namely topical and Twitter-specific) showing poor performance. Similarly, for the reputation dimensions classification task Wikipedia-based features outperform the feature set based on language, tweet, and word-occurrence.

Chapter 6

A Perspective-Aware Approach to Search: Visualizing Perspectives in News Search Results

A number of documents covering a similar range of topics may differ from each other in terms of different perspectives and subjective views exhibited in the documents [157]. This situation is even more prevalent in the case of controversial topics [56]. As an example, consider the case of existing political debates surrounding free speech, same-sex marriage, vaccinations etc — these debates have seen polarized views being expressed. The complex interplay between various topical narratives on documents found in different collections makes the information seeking process more complex. We may have documents that are topically similar but very dissimilar in opinion and sentiment expressed.

One interesting contribution of this thesis is an attempt to address the complexities in such scenarios via an innovative search engine called "perspective-aware search"; it attempts to identify implicit and explicit topical assertions in text in line with the third research question raised in Section 1.3 (Chapter 1). The front-end of the search engine enables a user to complement a query with what we call a "perspective" while the back-end utilises Wikipedia category-article structure to infer topical drifts with respect to the given query and perspective.

This chapter begins with an introduction to polarized discourse in Web search in an attempt to motivate the need for a novel perspective-aware search interface for analysis of search results. This is followed by a description of the system architecture whereby we explain the Wikipedia-based retrieval model of perspective-aware search. We then present some scenarios from within the news domain where such kind of search can yield useful insights. This is followed by a discussion in which we position

the notion of "perspective" as a means to study inherent subjectivity within the documents retrieved by a search engine; we also explain how "perspective-aware search" demonstrates the strength of our Wikipedia-based semantic relatedness framework described in Chapter 4. Finally, a user-study that is organized on principles of "interactive evaluation" concludes the chapter thereby serving as a proof into the usefulness of the proposed search interface.

6.1 Polarized Discourse in Web Search Results

The Web today is full of subjectively "biased" information on various topics and often the users are not even aware of the subjective views to which they are exposed in response to certain queries [56]. In an attempt to study the effects that polarized, political Web content has on Web search engines, some researchers performed an analysis of search queries and corresponding results [27, 220]. Their findings revealed that search engines nowadays expose their users to a narrow range of view-points.

There have been notable efforts in the information retrieval research community to provide users with an insight into the relationship between the query and the result set [93]. Capturing this information during the retrieval process provides the user with much valuable information (e.g. whether a term is overly specific, or whether a term is ambiguous etc.). Various attempts have been undertaken to tackle this problem, ranging from the definition of snippets [199] to the definition of approaches to cluster search results (e.g. Clusty...) [92], to the presentation of diversified search results in the first position of the ranked list offered to the users [189]. Recently there has been a resurgence of interest in defining visualization techniques of search results that offer an effective and more informative alternative to the usual and less informative ranked list. Pioneer visualization systems are represented by Tilebar [95], and Infocrystal [193], and more recently by the interface of the search engine Kartoo [16]. All these attempts have aimed to provide the users with more information than that provided by the traditional ranked list. This additional information can help the users in their search task (e.g. allowing them to navigate the collection more easily or providing evidence to allow the users to reformulate their query more efficiently).

Despite the above-mentioned efforts together with efforts at search result diversification that aim to minimize the effects of controversial Web content [224], there still remains a need for a system that helps pursue a qualitative and quantitative analysis of the amount of bias and controversy within the search results. In the following

subsection, we present an overview of a perspective-aware search engine that aims to support such analyses.

6.1.1 "Perspectives" for Monitoring Subjectively Biased Viewpoints in Search Results

Current information retrieval systems do not support means to investigate "potential bias" towards a certain perspective introduced during the search process. The "potential bias" may be introduced due to issues with the search engine itself or with the underlying collection. According to the Oxford Dictionary, the definition of perspective is as follows: "Perspective is a particular attitude towards or way of regarding something." In line with this we argue for incorporating the essential cognitive element of *"perspective"* within the search engine interface thereby introducing "perspective-aware" search in this thesis.

The proposed system allows the user to specify an additional input to the system along with standard type-keywords-in-entry-form interface for the entry of a "perspective" through a perspective phrase (see Figure 6.1). Note that this is not equivalent to appending the query with the perspective phrase because this modified query may not necessarily be a part of the user's search intent. However, there may be a bias in the result set towards a certain "perspective"; and hence, we propose perspective-aware search as a means to investigate and analyse a leaning towards an agenda. We explain through the following motivating example: Consider a case in which a user wishes to find information about a certain event (say, a bomb blast in a certain region). The search results returned may be polarized instead of focusing on factual aspects i.e., relating to a certain race, ethnicity, or political movement which caused violence. This can prompt a user to explicitly evaluate drift from objective factual reporting to subjective reporting within the top results. In doing such evaluation, the user is able to assess the prevalent controversies in returned results while discovering inherent subjective biases of the various document collections[1].

Our system utilises knowledge from Wikipedia to make conceptual sense of the perspective phrase. This knowledge does not modify the query (as would an additional query term) but is instead used to highlight the presence of a perspective in the result set.

[1]The documents collections of news web sites have this problem in particular on account of the political leaning they represent [83].

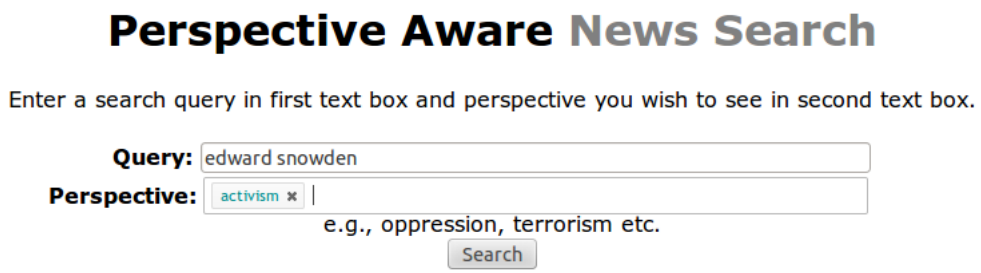

Figure 6.1: Perspective-Aware Search Entry Form

6.2 System Description

Figure 6.2 shows the architecture of our system. Note that the perspective-aware system within this architecture includes a perspective scoring system that uses the Wikipedia category-article structure to score the amount of content present inside a document with respect to the input perspective. The underlying perspective computation algorithm makes use of the semantic relatedness framework introduced in Chapter 4 of this thesis. We explain the perspective computation algorithm in Section 6.2.1.

As shown in Figure 6.2, the user enters the query together with the perspective phrase and the query is fed to the underlying information retrieval system which generates a ranked list of documents. The document extractor then forwards the content of the documents to the tokenizer and the extracted tokens along with the input perspective are fed into our perspective scoring system which uses our custom-built *WikiMadeEasy* API[2]. The perspective scoring system scores each token with respect to the perspective entered by the user and, the score of each token is aggregated to produce a perspective score for a document in the ranked list returned by the information retrieval system(s). Finally, the ranked list returned by the information retrieval system(s) and the perspective scores of tokens & documents returned by the scoring system is returned as output to generate the HTML result page.

6.2.1 Perspective Computation Algorithm

First, the candidate phrases are extracted from within the search results retrieved in response to a query (refer to section 4.1 of Chapter 4 of this thesis). This is followed by utilisation of the perspective phrase's Wikipedia article to extract category

[2]A more detailed description of the API which we name as WikiMadeEasy API appears in Appendix A of this thesis.

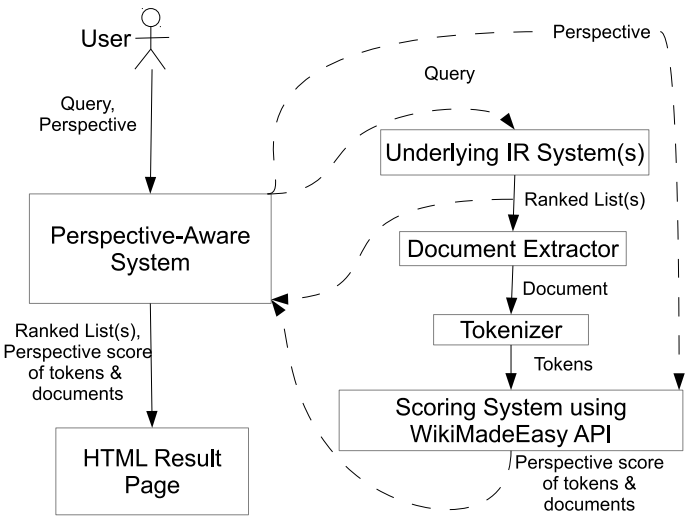

Figure 6.2: Perspective-Aware Search Overall Architecture

taxonomies for the relatedness score generation step (refer to section 4.2 of Chapter 4 of this thesis). Specifically, the categories and sub-categories to a depth count of two of the matched Wikipedia article corresponding to perspective phrase are utilised; note that these constitute RC explained in section 4.2.1 of Chapter 4 of this thesis. Following from the explanation in Section 4.2.1, we retrieve the set of all articles within the Wikipedia category set RC (we refer this set as $Articles_{RC}$), and all categories associated with these articles i.e. WC are retrieved. Recall from section 4.2.1 that the extracted tokens from a textual document (retrieved search result in this case) which are contained in $Articles_{RC}$ are called matched phrases (here, specifically they are phrases defining the perspective input by the user). We use these matched phrases to calculate the perspective (i.e., relatedness) score via the following relatedness measures in section 4.2.2 of Chapter 4 of this thesis:

- $Depth_{significance}$ (i.e., equation 4.6)
- $Cat_{significance}$ (i.e., equation 4.7)
- $Phrase_{significance}$ (i.e., equation 4.8)
- *Relatedness* (i.e., equation 4.9)

Finally, to facilitate user convenience, we have provided a *perspective autocompletion* feature within the system which simplifies the perspective input process for the user. To provide *perspective autocompletion* facility, we utilise all Wikipedia category names as possible perspectives and during the input process we suggest a short

list of category names which are textually similar to what the user is typing in the perspective input box.

6.3 Demonstration for News Domain

The perspective-aware search prototype we describe in this thesis is tailored for news search and fetches news search results from the US version of three popular search engines (i.e., Yahoo!, Google, and Bing). The video demonstration can be accessed at http://youtu.be/mPO763z6H4Y. The system provides additional information within the snippet of each news search result, where the perspective score for each result together with its relative perspective rank within the particular popular search engine is displayed[3]. Fig. 6.3 shows a snapshot of each search result and the additional information returned by the perspective scoring strategy is explained as follows:

- The bold list of keywords depicts the keywords related to the perspective provided during the issuance of the query.
- Score shows the score calculated by the scoring strategy, higher the score[4], higher is the amount of perspective found in the returned result.
- Rank shows the relative rank in terms of descending order for each list of top results returned by a popular search engine. Fig. 6.4 shows a list of search results, where different relative ranks can be seen corresponding to each result.

Furthermore, the search results of each search engine are displayed in a side-by-side manner as shown in Fig. 6.4. The system also employs visualization techniques (i.e., bar charts and line charts) to display the comparative perspective scores. Fig. 6.5 shows the visualizations incorporated in the system as they appear on the search results page giving the user further insights into the result sets of major search engines together with the web sites from which the result sets are retrieved. These charts attempt to address the following questions that a user may have:

- What are the differences in the search results retrieved by different search engines in terms of inherent perspective in the results? The two graphs at the bottom of Fig. 6.5 show this, where the graph on the left side shows the amount of perspective in the top ten results individually while the graph on the right

[3]It is the ordering by perspective scores for the top ten results.

[4]The minimum value which can be assumed by a score is 0, whereas there is no fixed upper bound.

Figure 6.3: Perspective Information Added to Snippet

side shows the cumulative perspective score. It is evident that Bing shows a higher perspective than Google and Yahoo!

- What is the amount of perspective displayed/contained by web sites in the top results of each search engine? The top graph of Fig. 6.5 shows this. For example, in the the Fig. 6.5 it can be seen that the "The Guardian"[5] newspaper shows the maximum amount of perspective "Activism" corresponding to query "Edward Snowden".

- What is the difference in the amount of perspective across the same web sites (news sources) covered by different search engines? Furthermore, which web sites (news sources) are covered more by a particular search engine? This is displayed on clicking "Show More" on the search results page.

An interface such as the proposed one can be particularly useful in exploratory tasks such as those commonly encountered in the news domain by journalists, media studies researchers or by end-users.

6.4 Discussion

Despite the fact that some notable efforts within the information retrieval research community have attempted to present a shift from the classical ranked list of search results to visualizations aimed at capturing various aspects of user intent [93]. The prominent of these include efforts along the following directions [214]:

- Systems aiming to provide insights into query and result set

- Systems aiming to provide insights into query and document collection

[5]http://www.theguardian.com

- Systems aiming to provide insights into result set and document collection

This additional information relating to query, result set and document collection can help the users in their search task (e.g. allowing them to navigate the collection more easily or providing evidence to allow the users to reformulate their query more efficiently). However, such efforts are unable to provide insights into the inherent subjectivity and controversy within the various topical dimensions of a result set or document collection. We aim to fill this gap through the notion of "perspective." Other related works investigate "search engine bias" and "search engine sentiment" which despite being somewhat related differ in terms of the underlying research goal. Works investigating bias in search propose a notion different to perspective-aware search in that their focus is towards analysis of retrievability which is a measure of the degree of ease with which certain Web pages are retrieved and predominantly studies how the search engine favors certain popular urls over others [150, 207]. On the other hand, works investigating sentiment in search analyse positive and negative sentiment over a topic through the use of external Web content (such as tweets, blog posts, opinion forums etc.) thereby complementing the search results with popular, public opinion on a topic [54] while our goal is towards the quantification of subjective bias exhibited by content creators for controversial topics. To further illustrate, consider a query on "Edward Snowden" where sentiment with respect to this query may be positive or negative while relating him with perspective "activism" is as per the choice of the content writer.

It is well-known that certain query topics involve a variety of opinions, judgements, and polarization; few examples include query topics related to theory of evolution, same-sex marriage, vaccinations, gun control, feminism etc. To the best of our knowledge, current systems lack in their ability to provide both qualitative and quantitative insights into the skewed retrieval process. As an example, consider a user wanting to know more about "same-sex marriage"; for the sake of neutrality[6] we also assume that this user has no opinion on the subject and wishes to pursue a neutral research on the topic. Given the recent activism on the subject of "LGBT rights", it is natural that most retrieved documents will likely be from authors who support such rights but such retrieval harms the search intent in this particular example. Complementing the search process with a "perspective" helps alleviate this problem by aiding the user in performing an explicit analysis of the documents that contain a skewed opinion and subjective view with respect to the topic at hand. Moreover, the quantitative

[6]This assumption aids the reader in understanding how perspectives come into play.

visualizations of Figure 6.4 provide a quantitative summary further helping the user in understanding the amount of controversy within the topic.

Query Topic	**Description of Controversial Nature**	**Perspective**
Abortion	This represents debate between "pro-life" and "pro-choice" activists on the issue of voluntary pregnancy termination	Murder
Edward Snowden	Edward Snowden is a former CIA professional who leaked classified information from U.S. National Security Agency (NSA); controversy surrounds him due to some considering him a hero and others considering him a traitor	Activism
Iran	Iran has always been at the forefront of nuclear race which is why holds a controversial place in the Western world	Nuclear Technology
Islam	Recently mainstream media associates acts of violence around the world with the religion "Islam" which is why it occupies a controversial status	Terrorism
Same-sex Marriage	Same-sex marriage continues to remain controversial on account of it going against traditional concept of "family" with most of the opposition coming from religious circles	Family

Table 6.1: Pre-selected Controversial Query Topics with Descriptions and Associated Perspectives

Finally, "perspective-aware search" serves as a proof of concept for demonstrating the strength of our relatedness framework built upon the Wikipedia category-article

structure. Our system uses an external and collectively created knowledge resource namely Wikipedia, which is less likely to be biased in a given direction. The category-article associations within Wikipedia which constitute the fundamental building block of our semantic relatedness framework (refer to Chapter 4) help obtain extra terms to represent the perspective of interest to the user. This knowledge (perspective term and related terms) does not change the query (as would an additional query term), but instead used to highlight the presence of a perspective in the result set, thereby helping the user in performing a search task with clarity and objectivity.

6.5 User Study for Perspective-Aware Search Evaluation

This section explores the use of our proposed perspective-aware search engine for analysis of search behavior when the information need involves a significant amount of controversy. We study the usefulness of the proposed interface through an online user study whereby various correlations between perspective "biases" and users' political orientations are analysed. The aim of this investigation is to provide evidence into the meaningfulness of the concept of "perspective" and how it aids the user in identification of subjective views contained in news articles returned by various search engines. Note that the evaluations presented here differ from traditional Cranfield-style evaluation paradigm that is commonly used in information retrieval [185], and instead bases itself on principles of "interactive information retrieval" evaluation [112].

The following subsections present details of our online study where we first present details of the methodology employed (i.e., participants' recruitment and variables measured) followed by a presentation of correlation analysis.

6.5.1 Data Collection

The data was collected by means of an online study wherein users were recruited via crowdsourcing as well as by mailing lists and were requested to use our perspective-aware search engine.

The recruited users were inquired about their political leaning (i.e., left-wing, right-wing or neutral). Moreover, they were asked to perform a search using our interface for three pre-selected topics from the list of topics shown in Table 6.1[7]; note that users in our study were asked to select query topics which they considered

[7]Table 6.1 shows the query topic along with presenting an explanation of why the topic is controversial and the "perspective" term that highlights a significant facet of the query topic.

controversial. Finally, we asked them to mark as relevant or irrelevant each perspective term corresponding to each search result returned by the three engines, namely, Google, Yahoo!, and Bing. The users were also asked to order by "perspective bias" the search results returned by Google, Yahoo!, and Bing[8]. A total of thirteen users were recruited of which five identified with a political orientation of left-wing, three identified with a political of right-wing, and the remaining five identified themselves as neutral.

Political Orientation	Google	Yahoo!	Bing
Left-Wing	42%	36%	47%
Right-Wing	44%	51%	38%
Neutral	72%	69%	83%

Table 6.2: Percentage of Perspective Terms Overlap across Various Political Orientations for Different Search Engines

Political Orientation	Google	Yahoo!	Bing
Left-Wing	4	5	3
Right-Wing	4	9	7
Neutral	2	1	1

Table 6.3: Perspective Ranking Difference across Various Political Orientations for Different Search Engines

6.5.2 Analysis of User-Study Results

Table 6.2 shows the percentage of terms marked as relevant by users of our study thereby representing the percentage of similarity between the terms considered relevant by our recruited users and our perspective computation algorithm of Section 6.2.1. Secondly, Table 6.3 shows the average Spearman footrule values for the rankings produced by users of our study and our algorithm presented in Section 6.2.1.

Table 6.2 clearly shows the usefulness of perspective terms for neutral users, and hence, we can argue for the usefulness of our perspective-aware search interface in highlighting subjective opinions within returned documents. Moreover, Table 6.3 shows lower average Spearman footrule values across the category of neutral users thereby providing evidence for the need of a search interface that aids the non-partisan user in analysis of subjective viewpoints. Such an interface can aid the user in formulating educated opinions rather than adoption of "bandwagon" opinions on issues of considerable significance while also involving considerable controversy.

6.6 Summary of the Chapter

In this chapter, we considered a novel application scenario which we called "perspective-aware search". The application serves as a proof-by-example for further analysis of the strength of the proposed Wikipedia-based semantic relatedness measures. More

[8]To assist the users, we explained to our users the concept of "perspective bias" clearly with the help of examples.

specifically, it advances research in the context of detecting controversial topics while also providing qualitative and quantitative visualizations. We began by positioning the idea of "perspective-aware search" in the context of the polarized Web followed by a demonstration of how it works. This was followed by an explanation of the system architecture and the fundamentals of the perspective computation algorithm. Finally, we presented few examples from within the news domain where subjectivity and polarization is abundant. We then presented a discussion on the strengths of the proposed search functionality along with its ability to effectively use Wikipedia category-article structure, and an online user-study demonstrating the usefulness of the proposed search functionality finally concludes the chapter.

No.	Google	Yahoo!	Bing
1	Britain targets Guardian newspaper over intelligence leaks related to Edward ... - Washington Post **public good, parliamentary committee, trade association, security;** Score: 0.95; Rank: 4 http://www.washingtonpost.com/world/europe/britai... - Cached Britain targets Guardian newspaper over intelligence leaks related to Edward Snowden Bethany Clarke/Getty Images - The Guardian Newspaper offices... about, said Jo Glanville, director of English PEN, a London-based freedom of expression group.	Microsoft, Yahoo Upgrades Shows Snowden Won, Obama Failed **electronic frontier foundation, whistleblower, security, nonprofit;** Score: 2.15; Rank: 2 http://www.bloomberg.com/news/2013-11-29/microsof... - Cached Microsoft, Yahoo Upgrades Shows Snowden Won, Obama FailedProtesters with posters of whistleblower Edward Snowden in front of the Reichstag in Berlin... general counsel for the Redmond, Washington-based company. Photographer: Kiyoshi Ota/Bloomberg	Edward Snowden could have doomsday cache of classified material: officials **nonprofit group, unlock, security agency, social media;** Score: 0.44; Rank: 9 http://www.nydailynews.com/news/national/edward-s... - Cached British and U.S. intelligence officials say they are worried about a "doomsday" cache of highly classified, heavily encrypted material they believe... possession. Snowden himself has been quoted as saying he took no such materials with him to Russia.
2	NSA terror over Edward Snowden's 'doomsday' cache of secrets - Daily Mail **nonprofit group, unlock, social media, security agency;** Score: 0.42; Rank: 5 http://www.dailymail.co.uk/news/article-2514095/N... - Cached NSA terror over Edward Snowden's 'doomsday' cache of secretsNSA terror over 'doomsday' cache of secrets stashed in online cloud by Edward... possession. Snowden himself has been quoted as saying he took no such materials with him to Russia.	Snowden doomsday data threat to spies **nonprofit group, unlock, social media, security agency;** Score: 0.40; Rank: 7 http://nypost.com/2013/11/25/snowden-doomsday-dat... - Cached Snowden asks world's help against US charges British and U.S. intelligence officials say they are worried about a doomsday cache of highly... for both NSA and the U.S. Office of the Director of National Intelligence declined to comment	Britain targets Guardian newspaper over intelligence leaks related to Edward Snowden **public good, parliamentary committee, trade association, security;** Score: 0.95; Rank: 5 http://www.washingtonpost.com/world/europe/britai... - Cached Britain targets Guardian newspaper over intelligence leaks related to Edward Snowden Bethany Clarke/Getty Images - The Guardian Newspaper offices... about, said Jo Glanville, director of English PEN, a London-based freedom of expression group.
3	Person of the year: Edward Snowden or the Pope - The Week Magazine **rights movement, privacy rights, crisis, surveillance;** Score: 0.30; Rank: 7 http://theweek.com/article/index/253449/person-of... - Cached 5 best books about the JFK assassination Even if you find people rankings tendentious, Time's Person of the Year provides a good opportunity to... have made universal compassion and good works popular again in Rome, and may energize the laity.	Spies worry over "doomsday" cache stashed by ex-NSA contractor Snowden **protest, nonprofit group, unlock, social media;** Score: 1.40; Rank: 3 http://finance.yahoo.com/news/spies-worry-over-do... - Cached A woman holds a portrait of former U.S. spy agency contractor Edward Snowden in front of her face as she stands in front of the U.S. embassy during... as saying he took no such materials with him to Russia. (Editing by Warren Strobel and Tim Dobbyn)	F-Secure: Snowden Files Prove America, China Surveillance Equals **whistleblower, privacy, the security, security;** Score: 0.91; Rank: 6 http://www.forbes.com/sites/tamlinmagee/2013/11/3... - Cached The NSA spy scandal, revealed by Edward Snowden, threatens to draw divisions between Europe and the United States. Finnish security company... have a different opinion than the politicians, Kangas said. I think they are as worried.

Figure 6.4: Perspective-Aware Search Results Presentation Corresponding to Query “Edward Snowden” and Perspective ”Activism”

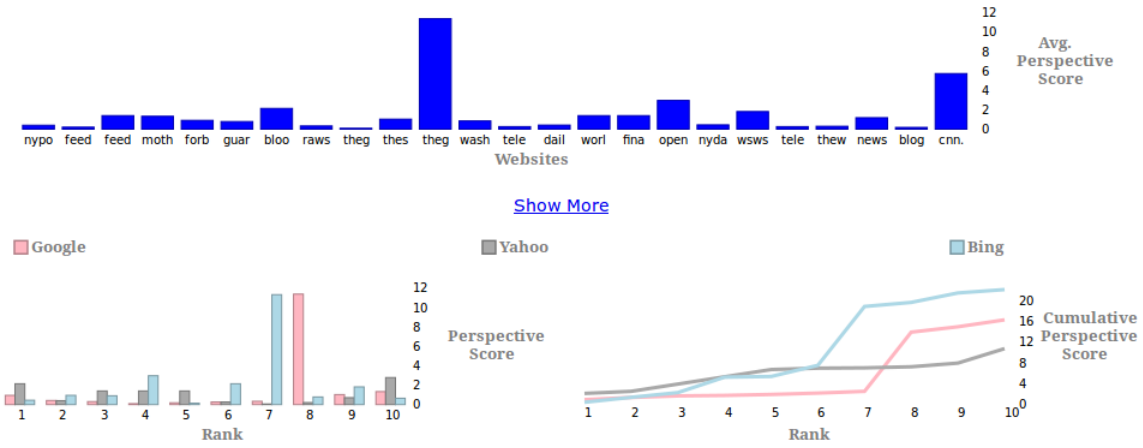

Figure 6.5: Perspective-Aware Search Graphical Comparison of Results Corresponding to Query “Edward Snowden” and Perspective ”Activism”

Chapter 7

Knowledge Extraction via Identification of Domain-Specific Keywords

This chapter presents an application scenario which can be a considered as a subtask within knowledge extraction (refer to Chapter 3 on related work). More specifically, we consider the utilisation of Wikipedia for the identification of keywords representative of a collection of textual documents (recall first research question from Section 1.3 of Chapter 1). We begin this chapter with an introduction to the problem of domain-specific keyword extraction[1]. This is followed by an explanation of the challenges involved in the domain-specific keyword extraction task. We then explain the proposed methodology in detail and finally conclude with presentation of experimental evaluations and their results. Note that unlike techniques described in the previous application scenarios (i.e., Chapter 5 and Chapter 6) our domain-specific keyword extraction technique does not rely on semantic relatedness measures introduced in Chapter 4; however, it is built upon exploitation of the Wikipedia category-article structure thereby being related to the fundamental research question being addressed in this thesis[2].

[1]To the best of our knowledge, this presents the first work that has attempted to deal with domain-specific keyword extraction for an entire document collection.

[2]More specifically, it is built on a community detection framework over the Wikipedia category-article structure as explained in Section 7.3.

7.1 Introduction to Domain-Specific Keyword Extraction

Large corpora of text contain significant information ranging from topics of a general nature to that of a specific nature; among the tasks that address the issue of identifying meaningful information from a huge repository is the task of keyword extraction which has been increasingly explored in recent years [69, 218, 104, 138, 88]. Keyword extraction is particularly important for various information access tasks such as exploratory search, query expansion, and document clustering to name a few.

Over the past few years several approaches have been proposed to extract keywords from text; current keyword extraction approaches work at the granularity level of single documents as they aim to identify keywords that characterise the content of a single document, and not that of a whole document collection [218, 205, 145, 210, 88]. We consider the task of extracting keywords at the granularity level of an entire corpus in an attempt to make inferences regarding the text collection.

The task of keyword extraction that represents the entire text collection may be defined as follows: given a text collection focused on multiple related knowledge (or topical) domains, the aim is to extract keywords that characterise knowledge domains represented in the text collection. As an example of this problem, let us consider a collection of Web sites of post-graduate schools; such a collection could be represented by specific keywords that characterize the academic research domains undertaken in those schools. We utilise related and interconnected category-article structure of Wikipedia for accurate identification of the range of topics on which the document collection is focused. Furthermore, we address the above task on a collection of short-text. The reason for this choice is twofold: 1) on one hand understanding the meaning of short-text is important in the age of micro-blogging, and 2) on the other hand processing short-text with respect to long-text over the WWW may be computationally less expensive through the application of appropriate techniques. Defining a technique that performs effectively and efficiently with short-text (for which defining a context is more difficult) is a challenging research issue that we address in this chapter.

In particular as short-text we consider the titles of Web pages, and we present a novel domain-specific keyword extraction method, which relies on both the notion of n-gram overlap between the titles of Wikipedia articles and the redirects of the Wikipedia articles and those of the short-text collection (titles of Web pages), and on a community detection algorithm that makes use of the Wikipedia category-article

structure in order to boost the performance of extraction of domain-specific keywords. The output of the proposed method is a set of meaningful keywords (n-grams) that define the topical domains of the considered collection. The proposed technique is composed of several steps aimed at refinement of the process of keyword selection.

7.2 Challenging Nature of Task

Extracting keywords that represent topical domains on which a short-text collection is focused is a challenging research issue due to the following reasons:

- Short-text cannot be assumed to contain well-written sentences like those in long-text; therefore extracting candidate keywords from them can be difficult in the sense that POS tagging may not work well as it would with long-text.
- Short-text contains very little context unlike long-text documents, therefore finding domain-specific keywords representing the focus of the entire collection is more challenging compared to that of a collection of long-text. For instance, in long-text we can exploit some features based on both the structure (formatting) and the length of documents (tf.idf), as shown in Section 3.5.2.

Due to the two above reasons, we propose to use Wikipedia to overcome the previously outlined problems. First, in order to discover candidate keywords, matching n-grams extracted from short-text with those of Wikipedia articles will eliminate non-readable terms or phrases such as "abcy2; the mere use of n-grams extracted from short-text would trivially increase the number of candidate keywords. Second, by matching n-grams with Wikipedia we are able to find a list of readable and sensible phrases (such as "tourist", "contact us", "information retrieval", "data mining", "database"), and each of the matched n-grams (i.e., Wikipedia articles) is linked with a number of Wikipedia categories which can then be exploited to define inter-relationships between the matched n-grams, hence exploiting this information. Note that the use of Wikipedia aids contextualization for short-text and this ties in with the second research question in Section 1.3 (Chapter 1).

7.3 The Proposed Methodology

In this section we present the method we have used to extract domain-specific keywords from a document collection. Figure 7.1 illustrates the complete approach.

First, we extract humanly readable phrases by calculating the intersection between the possible n-grams extracted from the input text and the Wikipedia article titles. Referring back to Section 1.5 and Figure 1.2 from Chapter 1, we repeat for the reader that the phrase extraction step is similar to the one introduced in Section 4.1 of Chapter 4. However, once extracted the candidate phrases are utilised in a community detection algorithm for definition of communities to which they belong. These communities are then ranked according to the richness of each community to represent the topical focus of the document collection. Finally, using these communities, we score the domain-specific keywords.

Note that the framework presented for domain-specific keyword extraction is slightly different from the one presented earlier in Chapter 4 whereby relatedness scores were calculated to represent the degree of association between an entity of interest and the candidate phrases. A pre-requisite for the semantic relatedness framework however was specification of an entity of interest against which to calculate relatedness scores; for the domain-specific keyword extraction task such entities do not exist, and the task is to determine significant keywords from within free textual data. Given the lack of pre-defined entities against which to determine semantic relatedness, we propose a slightly different approach as explained in following subsections.

7.3.1 Candidate Phrase Extraction

The aim of the step is to extract candidate phrases which are humanly understandable (standard English phrases) from the input text. Once the candidate phrases are discovered we can assign them scores according to their importance. For instance, "national talent hunt" and "business administration" are humanly understandable phrases but "c d iompra ochta" is not understandable, and it is intuitively sensible to filter noisy phrases. Therefore, in order to achieve this objective we perform intersection between possible n-grams (2-5 grams) extracted from the input text and the title of Wikipedia articles together with the redirects of the Wikipedia articles, where the titles of Wikipedia articles are well written by humans in order to describe a concept. Note that this particular step is similar to the one proposed in Section 4.1 of Chapter 4; the next two phases however differ on account of the task's nature whereby pre-defined entities do not exist and hence, semantic relatedness between two concepts cannot be captured.

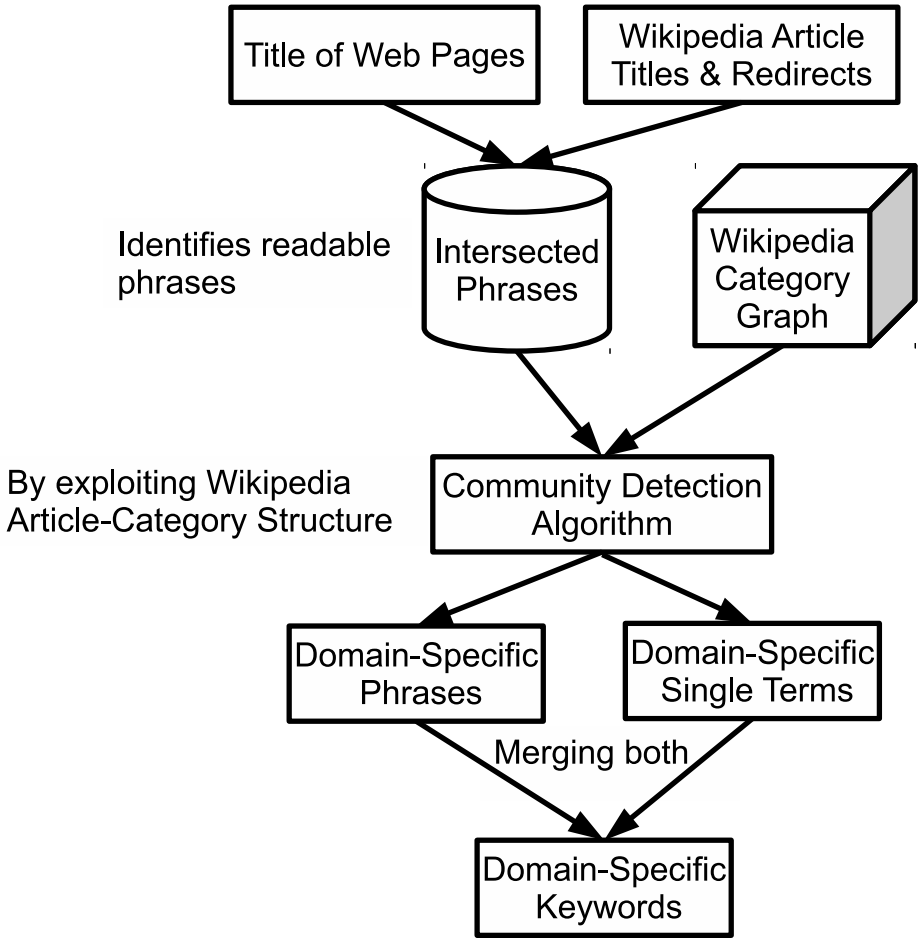

Figure 7.1: Methodology

7.3.2 Community Detection using Wikipedia Category Graph Structure

The aim of this step is to find communities among the Wikipedia categories by exploiting the Wikipedia category-article structure, and more specifically through utilisation of the associativity between Wikipedia categories and articles. Once communities are discovered, we utilise these communities to score candidate keywords according to the strength of communities to which each keyword belongs. This implies ranking keywords according to their association with the focus of the document collection as explained later in the Section 7.3.3.

7.3.2.1 The graph

First, we construct the semantic graph exploiting the candidate phrases (i.e., titles of Wikipedia articles) from the previous step, but instead of directly using the title of Wikipedia articles (i.e., candidate phrases), we use Wikipedia categories associated with the articles. We select Wikipedia categories on account of the existence of an organized structure within the Wikipedia category graph (see Figure 2.4). Therefore we choose Wikipedia categories as vertices and edges between these vertices follow the Wikipedia category structure (i.e., edges in the Wikipedia category graph, again refer to Figure 2.4). The weight on each edge is defined as the sum of the number of articles (i.e., matched candidate phrases) belonging to each Wikipedia category connected by the edge.

7.3.2.2 The communities

In this step, we apply the community detection algorithm to discover communities among Wikipedia categories. It is necessary to emphasize here that community detection is a well-known technique from within the domain of social network analysis that detects closely-knit groups, and for our purpose we use the undirected multilevel infomap algorithm. It is the successor of the algorithm which was found to be the best for the community detection problem [68, 182]; our choice is motivated by the high efficiency of the infomap algorithm on large graphs unlike community detection algorithms based on edge-betweenness [78], eigenvectors [154], and walktrap [167][3]. This algorithm yields an assignment of each category to exactly one commu-

[3]Note that we do not propose a community detection algorithm of our own and instead utilise the best known one from within social network analysis literature[6].

nity. Some communities may contain many categories while other communities may contain simply one category.

Rank of Community	Categories within Community
First	Political Philosophy, Political Theories, Political Ideologies, Political Science, Liberalism, ...
Second	Interdisciplinary Fields, Humanities, Applied and Interdisciplinary Physics, Academic Disciplines, Media Studies, ...
Third	Genetics, Population Genetics, DNA, Human Genetics, Genetic Engineering, ...

Table 7.1: Top-3 Communities after Application of Community Detection Algorithm

Finally, we rank communities in a way that the top-ranked community would represent the main domain of the document collection. Table 7.1 shows the top-3 communities with five categories representing a single domain[4] i.e. in case of top community all categories represent the domain of political science, in case of the second community all categories represent interdisciplinary fields, and finally in case of third community all categories represent genetics. Intuitively, our ranking of communities implies a richness of the top-ranked community demonstrated by the existence within it of several unique keywords related to each other which in turn are representative of the main domain (or focus) of the document collection. In order to rank rich communities higher, we rank (score) them on the basis of the number of *unique* articles they contain i.e., Wikipedia articles (or candidate phrases) which are only linked to Wikipedia categories exclusive to that community. Therefore, the top community would contain several unique articles which are exclusively defined by the categories of that community only. For example, an article on chemistry may be unique to the community that contains the categories related to chemical sciences and it would not be mentioned in other communities such as the community that contains categories related to political sciences. If a community contains several unique articles then it

[4]Note that the communities contain more categories but we only show five.

becomes a strong representative of the popular domain of interest compared to the community that contains lesser or zero unique articles.

7.3.3 Ranking Domain-Specific Keywords

The aim of this step is to rank candidate keywords according to their ability to represent the domain of interest of the collection. First, we will present the strategy to rank domain-specific phrases and then we will present the strategy to rank domain-specific single terms.

7.3.3.1 Ranking domain-specific phrases

Equation 7.1 scores the associativity of a candidate phrase (denoted as p) with the communities discovered in Section 7.3.2; fundamentally it involves determination of the representativeness of phrase p with respect to the main domain (or focus) of the document collection. This representativeness is calculated by means of measuring the richness of the communities that contain the phrase p whereby richness of a community is defined through the number of *unique* articles it contains. A phrase p can either be a unique article to a community (as explained in Section 7.3.2.2), and hence, belong to exactly one community or a phrase can belong to two or more communities i.e., there exists at least one or more category in each community that links to the phrase (i.e., matched Wikipedia article). Therefore we choose the average of community rank scores[5] for scoring the associativity of phrase with communities. Moreover, if a phrase belongs to a unique community then the phrase is specific to that community compared to a phrase which belongs to several communities. Likewise, if a phrase belongs to a top community then it is more likely to represent the focus of the document collection compared to phrases belonging to low ranked communities. This intuition is captured in the calculation of the associativity of a phrase p with the community that contains it as follows; we denote this associativity as $Avg_{CommRankSc}$:

$$Avg_{CommRankSc}(p) = \frac{\sum_{Community \in p} totalUniqueArticles(Community)}{|Community \in p|} \tag{7.1}$$

We then score each candidate phrase on the basis of their associativity with communities i.e. $Avg_{CommRankSc}$ and the word frequency of the phrase within the entire

[5]Community rank score is basically defined by the number of unique Wikipedia articles within that community.

document collection. The word frequency is a fairly common measure used in statistical natural language processing that highlights the significance of a term/phrase by means of how frequently its used.

$$Score(p) = log(freq(p) + 1) \times Avg_{CommRankSc}(p) \tag{7.2}$$

Equation 7.2 is used to score the importance of a phrase; the greater the score the higher the importance of the phrase.

7.3.3.2 Extraction and ranking of domain-specific single terms

In this step, we rank single terms extracted from the data that match Wikipedia category names using the following three criteria:

- The strength of the term's association with communities discovered from Section 7.3.2,
- Word frequency of the term within the entire document collection,
- Dominant word patterns where word patterns denote specific form of word prefixes and suffixes within a domain.

Of these, the first is identical to the previously explained associativity measure of a phrase p with the community in Equation 7.1. Similarly, the word frequency of the term is identical to the common frequency measure used within statistical natural language processing. The dominant word patterns are used to detect specific terms with special prefixes and/or suffixes that are mostly representative of a document collection; as an example within the domain of academic documents suffixes such as -'ics', 'ogy' are fairly common.

In order to discover the dominant word patterns, we generate an exhaustive list of candidate terms denoted as *Terms* by splitting all category names which are composed of two or more words[6] into single terms. We limit extraction to only those category names which appear in the community that has at least one unique article (see Section 7.3.2). We then adopt a simple heuristic from within a commonly used technique in 'Natural Language Processing' tasks [77, 125] whereby words with higher frequency of certain prefixes and suffixes get a higher score. Equations 7.3 and 7.4 below show the word pattern normalized and non-normalized scores respectively. First, Equation 7.3 determines an associativity score for terms with a commonly occurring prefix

[6]e.g., 'Cell_biology' is composed of two terms, 'cell' and 'biology'.

and/or suffix i.e., we take the cumulative effect of a term's word pattern across all discovered communities. This helps in determining those prefixes/suffixes that are highly reflective of the domain (or focus) of the entire document collection and words having common prefix and suffix (among the list of candidate single terms *Terms*) gets higher score compared to the words with unique prefix and suffix. Equation 7.4 then normalizes the obtained score by means of reducing the overall effect of prefixes/suffixes in the collection.

$$\begin{aligned} WordPattern(t_1) = & \sum_{\substack{t_2 \in Terms \\ |prefix(t_2)=prefix(t_1)}} Avg_{CommRankSc}(t_2) \\ & + \sum_{\substack{t_2 \in Terms \\ |suffix(t_2)=suffix(t_1)}} Avg_{CommRankSc}(t_2) \end{aligned} \tag{7.3}$$

$$WordPattern_{norm}(t) = \frac{WordPattern(t)}{Max(WordPattern(t_i))} \tag{7.4}$$

Finally, Equation 7.5 captures the importance of a single term by multiplying frequency of terms (which is skewed by the normalized word pattern score) with their (strength of) association with communities discovered from Section 7.3.2. Basically Equation 7.5 scores single terms by their importance within the entire document collection, and hence, the greater the score the higher the importance of the term.

$$Score(t) = log(freq(t) \times WordPattern_{norm}(t) + 1) \times Avg_{CommRankSc}(t) \tag{7.5}$$

7.4 Experiments and Results

In this section we present the employed dataset, the evaluation measures, and the performed experiments. We also present a discussion on the obtained results.

7.4.1 Dataset and Evaluation Measures

We performed experimental evaluations over a set of academic Web sites with an aim to extract keywords capturing the topics related to research and teaching activities performed by the departments in the considered universities. We constructed a collection of academic Web sites by crawling the English Web pages of the Web sites of eight post-graduate schools from five different countries as shown in Table 7.2. For each Web site, we crawled up to the depth of five from the root page in order to cover

School	Convention	Website (Location)
IBA Karachi Campus	IBA-KHI	www.iba.edu.pk (PK)
FAST NU Karachi Campus	FAST-NU-KHI	www.khi.nu.edu.pk (PK)
LUMS	LUMS	www.lums.edu.pk (PK)
MMU Cyberjaya Campus	MMU	www.mmu.edu.my (MY)
Milano-Bicocca	Milano	www.unimib.it/go/page/English[a] (IT)
NUI Galway Campus	NUIG	www.nuigalway.ie (IE)
Cambridge	Cambridge	www.cam.ac.uk (UK)
Oxford	Oxford	www.ox.ac.uk (UK)

[a]The URL has now changed for English Website

Table 7.2: Dataset of school Web sites

at least 80%-95% of the important Web pages, according to the estimate by Yates and Castillo [14]. In addition, to avoid crawler traps, i.e. infinite dynamic Web pages such as calendars, we adopted the policy to crawl at most the first 500 instances of each dynamic Web page.

Tables 7.3 and 7.4 show some statistics of the dataset. Table 7.3 shows the total number of crawled documents (Web pages), the total number of titles extracted (some Web pages do not have a title), the total number of unique titles, and the average length (in words) of titles (including and excluding stopwords) for each school's Web site and for an aggregation of all schools. Table 7.4 shows the total number of non-unique and unique words excluding stopwords, and a few examples of titles in the data set.

We performed the evaluations using the metrics of Precision at k (P@k, Equation 2.4), Average Precision (AP, Equation 2.5), Mean Average Precision (MAP, Equation 2.6), Reciprocal Rank (RR, Equation 2.7), and Mean Reciprocal Rank (MRR, Equation 2.8).

We now explain the amount of relevance judgements (i.e., manual annotations) that would be needed for the dataset. As Table 7.3 shows, the dataset contains 34,674 unique titles with an average title length of 6.3 words per title. The assessment of relevance judgements for the entire data set would be huge on account of variable length n-grams extracted from each unique title (expression). Precisely, the number of required relevance judgements for the dataset would be greater than the total

School	Total No. of Webpages	Total No. of Titles	No. of Unique Titles Discovered	Avg. Length of Titles including Stopwords (in words)	Avg. Length of Titles excluding Stopwords (in words)
IBA-KHI	411	411	169	4.4	3.3
Fast-NU-KHI	355	355	63	3.5	2.9
LUMS	2,783	2768	2,77	5.7	4.6
MMU	5,341	5,341	1,849	6.0	5.0
Milano	443	443	214	7.2	5.5
NUIG	29,248	29,182	7,552	6.0	4.9
Cambridge	26,765	26,749	12,859	6.5	5.4
Oxford	26,866	26,787	11,855	6.6	5.1
Total	**92,212**	**92,036**	**34,674**	**6.3**	**5.1**

Table 7.3: Statistics of the Dataset

Total No. of Non-Unique Words excluding Stopwords	466,449
Total No. of Unique Words excluding Stopwords	23,266
Example of Titles of Web pages	'document moved', 'index', 'the resource cannot be found.', 'login', 'frequently asked questions', 'political ecology: a critical introduction (blackwell critical introductions to' geography)'

Table 7.4: More Statistics of Dataset

number of unique titles times the number of possible keyword extractions from the titles[7]. Due to the difficulty in complete annotation of such an enormous dataset, the relevance judgements were obtained for the top-20 results.

7.4.2 Evaluations

We conduct two types of experiments; the first experiment type evaluates the quality of extraction of our method at each step (as individual components) and the second experiment type evaluates the quality of single terms extraction. For both types of experiments, 13 human annotators[8] made the relevance judgements for the top-20 results by associating a label *relevant* or *irrelevant* with each extracted keyword; this amounts to a total of 1320 relevance judgements across phrases and 676 relevance judgements across single terms. For each keyword, the 13 judgements are aggregated to produce a single label: a keyword is labelled as *relevant* (or *irrelevant*) with the majority vote.

Before conducting experiments, we produced variants of the proposed methodology for the identification of domain-specific keywords, explained below:

n-grams: the basic algorithm (or a baseline) that extracts all possible n-grams and orders them by (descending) frequency of each n-gram.

inter: the algorithm that extracts readable phrases as discussed in Section 7.3.1 and orders them by their (descending) frequency.

comp_phrases: the algorithm based on extracted phrases ordered by scoring strategy as discussed in Section 7.3.3.1.

complete: the algorithm based on extracted phrases ordered by scoring strategy as discussed in Section 7.3.3.1 and extracted single terms ordered by the scoring strategy from Section 7.3.3.2.

By evaluating these variants, we aim to gain an insight into the contribution of the individual components to the overall system performance.

7.4.2.1 Experiment Type 1

In this experiment, we evaluate the capability of the approach to generate high quality domain-specific keywords. First we compare individual components of the proposed methodology and then we compare the proposed methodology with current state-of-the-art approaches. We asked annotators to label a keyword as relevant when it

[7]Number of possible keyword extractions equal C(n+1,2), where n is the word length of the extracted keyword.

[8]Except one, all the rest have completed (at least) their post-graduate studies.

correctly represents a complete name of a topical domain or sub-domain (academic topical area of interest). For instance,'Information Retrieval', 'Marine Biology', and 'Science' are relevant examples but 'Marine' is an irrelevant keyword because it does not represent the name of a topical domain or sub-domain. Inter-annotator agreement was calculated using Fleiss's Kappa [66], which showed high agreements (value 0.83).

Tables 7.5–7.7 show the quality of identifying domain-specific keywords by individual components of the proposed methodology for the top-20 keywords. These tables show that the ***complete*** algorithm outperforms the rest. Table 7.5 shows that ***complete*** extracts more relevant keywords than the rest as evident in the aggregated mean (of P@20), table 7.6 shows that ***complete*** has higher tendency of extracting first relevant keyword than all of the others as evident in the MRR, and table 7.7 shows that on the average ***complete*** extracts more relevant keywords earlier in the ordered list than the rest as evident in the MAP.

Tables 7.8–7.10 show the comparison of the proposed algorithm with tf-idf, TextRank, ExpandRank[9], and TagMe[10] for the top-20 extracted keywords by each. From these tables it is clear that our algorithm (both ***complete*** and ***com_phrases***) outperforms the rest of the algorithms in terms of aggregated mean (of P@20), MRR, and MAP. Note that **TR_1**, **TR_2**, and **TR_7** are the best cases of TextRank with parameter window size 1, 2, and 7 respectively, within the tested range of 1-10. Similarly, **ER_1_1** and **ER_3_1** are the best cases of ExpandRank with parameters window size 1, neighbourhood size 1 and window size 3, neighbourhood size 1 respectively, for the tested range of window size 1-10 and neighbourhood size 1-8.

7.4.2.2 Experiment Type 2

In this experiment, we evaluate the capability to generate high quality domain-specific key terms (i.e., single terms only) which can be useful for generating single term tag cloud. We asked annotators to label a key term as relevant when it correctly represents a complete or partial name of a topical domain or sub-domain (academic topical area of interest). For instance, 'Science' and 'Biology' are relevant examples, and so is 'Marine' when it represents a partial representation of 'Marine Biology'. Similar to the previous experiment, we calculated Fleiss's Kappa and found the value of 0.78, showing a high agreement among annotators.

[9]We used the implementation by [87].

[10]We used the public API http://tagme.di.unipi.it/

School	**complete**	**com_phrases**	**inter**	**n-grams**
Cambridge	0.75	0.65	0.15	0.00
Fast-NU-KHI	0.13	0.00	0.00	0.00
IBA-KHI	0.35	0.35	0.30	0.05
LUMS	0.50	0.50	0.20	0.10
MMU	0.55	0.50	0.40	0.00
Milano	0.80	0.75	0.40	0.00
NUIG	0.75	0.70	0.40	0.00
Oxford	0.75	0.60	0.30	0.00
Aggregated Mean	0.57	0.51	0.27	0.02

Table 7.5: P@20 for the identification of domain-specific keywords

School	**complete**	**com_phrases**	**inter**	**n-grams**
Cambridge	1.00	1.00	0.08	0.00
Fast-NU-KHI	0.50	0.00	0.00	0.00
IBA-KHI	0.50	0.50	0.33	0.14
LUMS	0.50	0.50	1.00	0.20
MMU	1.00	1.00	0.17	0.00
Milano	1.00	1.00	1.00	0.00
NUIG	1.00	1.00	0.25	0.00
Oxford	1.00	1.00	0.20	0.00
MRR	0.81	0.75	0.38	0.04

Table 7.6: RR and MRR for the identification of domain-specific keywords

School	complete	com_phrases	inter	n-grams
Cambridge	0.81	0.74	0.13	0.00
Fast-NU-KHI	0.50	0.00	0.00	0.00
IBA	0.43	0.43	0.33	0.14
LUMS	0.59	0.50	0.60	0.18
MMU	0.58	0.70	0.33	0.00
Milano	0.86	0.70	0.72	0.00
NUIG	0.77	0.73	0.36	0.00
Oxford	0.88	0.80	0.31	0.00
MAP	0.68	0.57	0.35	0.04

Table 7.7: AP and MAP for the identification of domain-specific keywords

School	complete	com_phrases	tf-idf	TR_1	TR_2	TR_7	ER_1_1	ER_3_1	TagMe
Cambridge	0.75	0.65	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Fast-NU-KHI	0.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
IBA-KHI	0.35	0.35	0.00	0.10	0.10	0.10	0.15	0.15	0.05
LUMS	0.50	0.50	0.05	0.05	0.05	0.20	0.15	0.25	0.20
MMU	0.55	0.50	0.00	0.05	0.00	0.00	0.00	0.00	0.00
Milano	0.80	0.75	0.25	0.30	0.25	0.30	0.80	0.75	0.20
NUIG	0.75	0.70	0.00	0.25	0.00	0.00	0.00	0.00	0.00
Oxford	0.75	0.60	0.00	0.00	0.05	0.00	0.00	0.00	0.00
Aggregated Mean	0.57	0.51	0.04	0.09	0.06	0.08	0.14	0.14	0.06

Table 7.8: P@20: comparison between different algorithms for the identification of domain-specific keywords

School	**complete**	**com_phrases**	**tf-idf**	**TR_1**	**TR_2**	**TR_7**	**ER_1_1**	**ER_3_1**	**TagMe**
Cambridge	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Fast-NU-KHI	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
IBA-KHI	0.50	0.50	0.00	0.50	0.20	0.17	0.25	0.11	0.07
LUMS	0.50	0.50	0.06	0.07	0.08	0.50	0.33	1.00	0.20
MMU	1.00	1.00	0.00	0.09	0.00	0.00	0.00	0.00	0.00
Milano	1.00	1.00	0.17	0.20	0.20	0.20	1.00	0.20	0.25
NUIG	1.00	1.00	0.00	0.14	0.00	0.00	0.00	0.00	0.00
Oxford	1.00	1.00	0.00	0.00	0.06	0.00	0.00	0.00	0.00
MRR	0.81	0.75	0.03	0.13	0.07	0.11	0.20	0.16	0.06

Table 7.9: RR and MRR: comparison between different algorithms for the identification of domain-specific keywords

School	**complete**	**com_phrases**	**tf-idf**	**TR_1**	**TR_2**	**TR_7**	**ER_1_1**	**ER_3_1**	**TagMe**
Cambridge	0.81	0.74	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Fast-NU-KHI	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
IBA-KHI	0.43	0.43	0.00	0.31	0.19	0.15	0.23	0.16	0.07
LUMS	0.59	0.50	0.06	0.07	0.08	0.36	0.29	0.59	0.37
MMU	0.58	0.70	0.00	0.09	0.00	0.00	0.00	0.00	0.00
Milano	0.86	0.70	0.21	0.25	0.25	0.28	0.75	0.56	0.26
NUIG	0.77	0.73	0.00	0.26	0.00	0.00	0.00	0.00	0.00
Oxford	0.88	0.80	0.00	0.00	0.06	0.00	0.00	0.00	0.00
MAP	0.68	0.57	0.03	0.12	0.07	0.10	0.16	0.16	0.09

Table 7.10: AP and MAP: comparison between different algorithms for the identification of domain-specific keywords

School	complete	com_phrases	inter	BM25	tf-idf	tf-norm	n-grams	TR_1	TR_2	TR_7	ER_1_1	ER_3_1	TagMe
Cambridge	0.70	0.70	0.35	0.00	0.05	0.00	0.05	0.05	0.05	0.00	0.05	0.05	0.00
Fast-NU-KHI	0.13	0.00	0.10	0.10	0.10	0.10	0.10	0.00	0.00	0.00	0.00	0.00	0.10
IBA-KHI	0.45	0.30	0.25	0.15	0.15	0.15	0.10	0.15	0.10	0.10	0.10	0.10	0.10
LUMS	0.50	0.40	0.15	0.25	0.25	0.25	0.25	0.10	0.10	0.20	0.15	0.20	0.25
MMU	0.60	0.60	0.40	0.00	0.00	0.00	0.05	0.30	0.15	0.10	0.15	0.10	0.00
Milano	0.90	0.65	0.55	0.40	0.40	0.45	0.50	0.50	0.50	0.50	0.45	0.45	0.40
NUIG	0.55	0.55	0.45	0.05	0.05	0.05	0.10	0.40	0.45	0.40	0.30	0.15	0.00
Oxford	0.65	0.65	0.25	0.20	0.20	0.15	0.25	0.05	0.10	0.10	0.00	0.00	0.00
Aggregated Mean	0.56	0.48	0.31	0.14	0.15	0.14	0.18	0.19	0.18	0.18	0.15	0.13	0.11

Table 7.11: P@20 for the identification of domain-specific single key terms

For this experiment, the extracted keywords were reduced to a list of single terms while preserving the score of each term in a keyword and increasing the score whenever there are more matches for that term in a different keyword. For example, consider having only two n-grams in the index; 'a b' with score 'n' and 'b c' with score 'k', so now the extracted single terms would have scores 'a': 'n', 'b':'n+k', and 'c':'k'. Furthermore, we lemmatize all the obtained terms in order to use a conceptual representation of each term (e.g., sciences becomes science) while scoring them[11].

In this experiment, we compare the individual components of our overall system against BM25[179], tf-idf and tf-norm (term frequency normalized), TextRank, ExpandRank, and TagMe. Tables 7.11–7.13 show that our system significantly outperforms other algorithms.

To provide an illustration of typical results, Table 7.14 shows the data from the Milano Web site. In this table, we show top-20 domain-specific keywords (for experiment 1) and domain-specific single key terms (for experiment 2) detected by ***complete***.

7.4.3 Failure Analysis

So far, we have presented average results of the various components of our approach. Here, we present a failure analysis.

Given the low performance of the evaluation measures for some universities, we performed an analysis of the dataset. Our analysis led to the observation that the datasets with Precision at 20 of less than 0.6 contain noisy terms not related to academic research. This is on account of the dataset of corresponding universities from within our academic web sites' collection being from countries that are not

[11]We count the variations just once as a key term in order to avoid redundant terms.

School	complete	com_phrases	inter	BM25	tf-idf	tf-norm	n-grams	TR_1	TR_2	TR_7	ER_1_1	ER_3_1	TagMe
Cambridge	1.00	1.00	0.17	0.00	0.06	0.00	0.08	0.08	0.06	0.00	0.13	0.10	0.00
Fast-NU-KHI	0.33	0.00	0.25	0.06	0.06	0.06	0.20	0.00	0.00	0.00	0.00	0.00	0.07
IBA-KHI	1.00	1.00	0.33	0.09	0.09	0.10	0.20	0.33	0.50	0.50	0.50	0.50	0.08
LUMS	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.08	0.08	0.50	0.25	0.50	1.00
MMU	1.00	1.00	0.25	0.00	0.00	0.00	0.08	0.14	0.13	0.13	0.17	0.17	0.00
Milano	1.00	1.00	1.00	0.25	0.25	0.25	0.25	1.00	0.33	1.00	1.00	1.00	0.25
NUIG	1.00	1.00	0.14	0.25	0.33	0.13	0.17	0.33	0.17	0.17	0.50	0.17	0.00
Oxford	1.00	1.00	0.33	0.20	0.20	0.20	0.13	0.11	0.14	0.13	0.00	0.00	0.00
MRR	0.92	0.88	0.43	0.23	0.25	0.22	0.26	0.26	0.18	0.30	0.32	0.30	0.18

Table 7.12: RR and MRR for the identification of domain-specific single key terms

School	complete	com_phrases	inter	BM25	tf-idf	tf-norm	n-grams	TR_1	TR_2	TR_7	ER_1_1	ER_3_1	TagMe
Cambridge	0.75	0.75	0.28	0.00	0.06	0.00	0.08	0.08	0.06	0.00	0.13	0.10	0.00
Fast-NU-KHI	0.33	0.00	0.33	0.09	0.08	0.09	0.27	0.00	0.00	0.00	0.00	0.00	0.10
IBA-KHI	0.49	0.51	0.28	0.14	0.13	0.14	0.21	0.22	0.35	0.39	0.45	0.45	0.10
LUMS	0.61	0.50	0.87	0.69	0.66	0.62	0.71	0.11	0.12	0.38	0.23	0.49	0.55
MMU	0.94	0.89	0.38	0.00	0.00	0.00	0.08	0.23	0.16	0.13	0.19	0.16	0.00
Milano	0.89	0.83	0.69	0.39	0.37	0.42	0.42	0.55	0.44	0.50	0.67	0.46	0.36
NUIG	0.77	0.75	0.35	0.25	0.33	0.13	0.14	0.42	0.38	0.34	0.40	0.20	0.00
Oxford	0.74	0.73	0.36	0.20	0.19	0.17	0.22	0.11	0.13	0.15	0.00	0.00	0.00
MAP	0.69	0.62	0.44	0.22	0.23	0.20	0.27	0.22	0.20	0.24	0.26	0.23	0.14

Table 7.13: AP and MAP for the identification of domain-specific single key terms

Type	Extracted Data
Keywords (for Exp. 1)	biotechnology, sociology, developmental psychology, economics, technology, computer science, statistics, international community, global markets, residence permit, international relations, financial institutions, political economics, human resource development, economic systems, legal services, business law, physics, political sciences, business administration
Single Key Terms (for Exp. 2)	science, economics, service, psychology, business, international, community, technology, political, medicine, social, human, physics, biotechnology, economic, sociology, developmental, computer, law, communication

Table 7.14: Extracted data from Milano Web site

well-known for academic research and, in cases when they are engaged in academic research it is not promoted on the university's web site. These universities are IBA-KHI, FAST-NU-KHI, and LUMS from Pakistan; and MMU from Malaysia.

Moreover, a potential limitation of the community detection framework built on top of the Wikipedia category graph arises from the noise within Wikipedia category structure. The derived communities despite containing Wikipedia categories that are related to each other do not always contain Wikipedia categories representing scientific fields. As an example, a subcategory of "Data Management" is "Computer file systems" which is linked to Wikipedia article "file directory." In case of the keyword extraction problem addressed in this chapter, academic web titles contain the phrase "file directory" which is basically irrelevant as an academic area of research. Other examples are phrases such as "long term" and "full time" detected from within Wikipedia subcategories of "Economics terminology" and "Employment classifications" respectively which in turn are derived from Wikipedia categories "Economics" and "Economic classification systems" respectively.

7.5 Summary of the Chapter

This chapter presented an approach for the identification of domain-specific keywords using the Wikipedia category-article structure. We explained the uniqueness of the task and how it differs from standard keyword extraction in that it performs keyword extraction at the granularity level of the entire corpus instead of single documents; note that this fundamentally addresses the first research question raised in Section 1.3 of Chapter 1 of this thesis. Furthermore, we also proposed to perform the task on short-text on account of computational complexities over long-text. However, short-text suffers from the problem of lack of context thereby making the process of textual inferences inherently difficult. This however is solved through utilisation of context driven from the Wikipedia category-article structure thereby leading to addressing the second research question raised in Section 1.3 of Chapter 1 of this thesis.

We began the chapter with a formal introduction to the task of domain-specific keyword extraction, followed by an explanation of the challenging nature of the task. We then described the overall architecture of the domain-keyword extraction framework, followed by a detailed explanation of our strategy to extract keywords representative of a given document collection. We explained the process of communities' extraction over the Wikipedia category graph followed by a description of the algorithm that ranks the keywords on basis of the strengths of associations between Wikipedia categories and articles. Finally, we presented details of experimental evaluations over a custom dataset gathered from Web page titles of school web sites in order to demonstrate the strength of our approach.

Chapter 8

Conclusion

In this chapter, we present the summary of the contributions made in this thesis and position it with respect to advancements in the field of text mining by summarizing the significance of research outcomes of this thesis. We revisit the research questions presented in Chapter 1 and review our findings with respect to these research questions. Then, we discuss some limitations within our research contributions made in this thesis and present some future directions with respect to them.

8.1 Summary of Contributions

We first present a summary of our contributions in this thesis followed by a comparison of the use of our framework in the different contributions of this thesis. Text mining has attracted significant attention from the research community on account of the recent plethora of web-enabled applications that generate huge amounts of textual data. There is a need for advances in algorithmic design which can learn interesting patterns from textual data. One recent advancement that researchers have explored is the use of Wikipedia for text mining applications [64, 71, 141, 217]. This thesis is also a step in that direction; however, to the best of our knowledge we have pioneered efforts that show the effectiveness of Wikipedia category-article structure to address a number of natural language processing tasks i.e., classifying a piece of short-text relevant to a particular entity or not, classifying a piece of short-text along pre-specified topical dimensions, a news search engine interface that scores the presence of inherent perspectives among the search results, and a keyword extraction method for short-text corpus.

Following presents a focussed summary of the contributions of this thesis:

8.1.1 Classification Task

We addressed two tasks of classification for a dataset of tweets i.e., filtering task and reputation dimensions classification task. In both of the tasks, we proposed a strategy that exploits Wikipedia category taxonomies from within the Wikipedia category-article structure to define a notion of semantic relatedness between the category taxonomies matched in a tweet with either an entity or topical reputation dimension. To match a tweet with a category taxonomy, we extracted the phrases that matched with the Wikipedia article titles/redirect within the tweet, and then we extracted associated Wikipedia categories with the matched phrases to define the notion of relatedness. The relatedness comprised three essential constituents listed as follows:

- The depth within the category taxonomy where the match between a phrase and Wikipedia category occurs. The intuition behind this is that the deeper the category, the lesser its significance for the task being explored.
- The intersection between the relevant Wikipedia categories (i.e., those related directly to the entity of interest and containing Wikipedia article titles corresponding to matched phrase) and the additional Wikipedia categories (i.e., those further away from the entity of interest and containing Wikipedia article titles corresponding to matched phrase).
- Word length and frequency of occurrence of matched phrase.

To the best of our knowledge, this is one of the first research efforts that has attempted to formulate the definition of semantic relatedness through the use of the association between Wikipedia categories and articles. Our semantic relatedness framework is able to achieve a considerably high performance when utilised in a classification framework for short-text; this was demonstrated via comparison between Wikipedia-based semantic relatedness features and topical features whereby Wikipedia-based features provide two-fold performance improvement in classification accuracy (refer to Table 5.20 in Chapter 5). Our proposed semantic relatedness framework which forms the core of our classification metholodology for tweets in an "online reputation management" scenario is able to outperform state-of-the-art Wikipedia-based approaches along with approaches that utilise textual features (refer to Tables 5.21 and 5.22 in Chapter 5).

8.1.2 News Search Interface

We utilised the notion of relatedness between Wikipedia category taxonomies and textual content to infer the relationship between the phrases that appear in the text and a particular topic of interest. This topic of interest is what we introduced as "perspective" which basically served as a means to capture subjective and controversial views within textual content. Through utilisation of various related concepts within Wikipedia categories and articles, the potential bias behind a textual piece is explored in both a qualitative and quantitative fashion. The relationship to be explored is specified at query time by the user through the novel search engine interface. "Perspective-aware search" served as a proof of concept for demonstrating the strength of our relatedness framework built upon Wikipedia category-article structure. We verified the usefulness of our novel search engine interface by means of an online user-study whereby users belonging to different political orientations are asked whether or not they agree with the subjective biases discovered by our underlying perspective computation algorithm. We discover the existence of high agreement scores clearly showing the usefulness of "perspective-aware search" in aiding the non-partisan user when it comes to analysis of subjective viewpoints (refer to Tables 6.2 and 6.2 in Chapter 6).

8.1.3 Keyword Extraction

In the absence of context in short text such as in the titles of Web pages, it is difficult to define the relationship among extracted terms and phrases. In order to overcome this problem we developed a strategy utilising Wikipedia category-article structure for discovery of the communities of related Wikipedia categories. These communities are then exploited to extract meaningful keywords which are related with each other and represent the domain of the corpus of the short text. First, we extracted the phrases that match a Wikipedia article title/redirect and then we extracted the Wikipedia categories associated with those titles/redirects. Once these categories were extracted we arranged them in a graph by utilising Wikipedia category graph structure and defined the edge weight as proportional to the matching that occurred with Wikipedia articles/redirects in those categories. Finally, we applied community detection on this graph for extraction of domain-specific keywords. The extracted phrases are then ranked with the help of the detected communities, and phrases belonging to rich communities are ranked higher. Extensive experimental evaluations demonstrated the strength of our keyword extraction approach built using community detection

(refer to Tables 7.8 - 7.10), and for the academic collections which do not perform so well our technique still outperforms state-of-the-art keyword extraction techniques.

8.2 Significance of Research Outcome

The fundamental goal of text mining as a research area has been improvement of the quality and effectiveness of the inferences derived from textual data, and for this purpose several approaches have been proposed for different text mining applications. Of these approaches only few have explored the role of knowledge bases and their potential in solving the various problems that arise in the domain of text mining. Furthermore, the largest human-curated, online encyclopaedia has not been utilised to its potential and we aimed to achieve this via the above contributions.

A novel aspect of the contributions in Section 8.1 is utilisation of Wikipedia categories and Wikipedia articles together as a source of information. Wikipedia categories are organized in a taxonomical manner serving as semantic tags for Wikipedia articles and this provides a strong abstraction and expressive mode of knowledge representation. We used this mode (i.e., Wikipedia's category-article structure) in the domains of text classification, analysis (via a notion of "perspective" in news search), and keyword extraction.

For text classification and subjectivity analysis, we have proposed a semantic relatedness framework which first extracted phrases matching Wikipedia article titles/redirects, and then utilised these phrases in matched Wikipedia categories corresponding to the entity of interest in order to determine the relatedness between phrases and the entity of interest. The relatedness measure made use of 1) the Wikipedia category depth at which a phrase matches a Wikipedia article associated with Wikipedia category taxonomies related to the entity of interest, 2) the intersection of Wikipedia categories between the Wikipedia categories related to Wikipedia article matching the candidate phrase and the Wikipedia category taxonomies related to the entity of interest, ad 3) significance of candidate phrase itself. These relatedness measures when used as features in text classification and subjectivity analysis yielded accurate results as demonstrated through experimental and user-study evaluations in Chapters 5 and 6 respectively. For the domain-specific keyword extraction task, we extracted phrases in a similar manner as described for the above task. The matched Wikipedia categories corresponding to the matched phrases are utilised in a community detection framework which helped discover communities comprising Wikipedia categories. The phrases are then scored with the help of rich communities where richness of a

Table 8.1: Summary of our framework for text classification/subjectivity analysis and keyword extraction

Text Classification and Subjectivity Analysis	**Keyword Extraction**
Extracts candidate phrases based on matches with Wikipedia article titles/redirects.	Extracts candidate phrases based on matches with Wikipedia article titles/redirects.
Requires a pre-defined entity of interest to defined Wikipedia category taxonomies across which semantic relatedness is computed	Does not require a predefined entity.
Operates on the associations between Wikipedia categories and Wikipedia articles.	Operates on the associations between Wikipedia categories and Wikipedia articles.
Utilises relatedness measures defined over Wikipedia category depths and Wikipedia category intersections.	Utilises communities defined over Wikipedia category graph.
Takes phrase significance into account.	Takes phrase significance into account.

community is basically derived by means of the unique Wikipedia articles contained in it. In other words, closely-knit communities with phrases representing the focus of the document collection are ranked higher and thereby extracted as keyphrases. Table 8.1 summarizes these application scenarios and the use of Wikipedia category-article structure within them.

The research outcomes show significant promise and we posit that this thesis could serve as a starting point into further exploration efforts for relationship into Wikipedia categories and their associated articles. Furthermore, there is rich knowledge encoded within the associations between Wikipedia categories and Wikipedia articles which can be effectively utilised to further advance research within text mining.

8.3 Answers to Research Questions

We examine how our work answers the research questions stated in Chapter 1.

RQ1. ***How can Wikipedia be used for the identification of effective keywords that summarize the text collection?***

This question has been answered by means of the keyword extraction framework that we explained in Chapter 7 whereby we proposed the utilisation of Wikipedia categories in a community detection framework; and the Wikipedia categories within these communities were further utilised for ranking the Wikipedia articles they were associated with. This ranking in turn reflected the significance of the keywords that had a match with the Wikipedia article titles/redirects which finally lead to the extraction of the significant keywords. Furthermore, the uniqueness of our approach lies in its ability to effectively operate at the granularity level of the entire corpus thereby providing an effective summary for the entire text collection and producing an output that comprises domain-specific keywords.

RQ2. ***How can Wikipedia be used for enhanced context representation within an informal text piece?***

This question has been answered by application of our Wikipedia-based methods first on a corpus of tweets in Chapter 5 and then on a corpus of Web page titles in Chapter 7. The primary limitation of short-text is its lack of context in order to make useful inferences. Years and years of research efforts within the text mining domain have focused on long-text with most of the algorithms operating on assumptions that stand true for this form of text. These are text mining approaches that utilise term co-occurence features, features extracted from various portions of a long-text document, features that capture frequency of occurrence of terms, etc. These assumptions however are inapplicable over short-text where the surrounding context is too limited thereby rendering traditional text mining approaches useless. Wikipedia provides an excellent resource in such cases by enriching the textual content with useful information that helps provide context and in our case through the relationships between Wikipedia categories and articles.

RQ3. ***How can we identify various topical assertions (both implicit and explicit) in a piece of text?***

The answer to this question lies in the innovative notion of "perspective" that

we introduced in Chapter 6. Subjective biases of certain authors are implicit in a textual piece whereas some authors are explicit about their views. Our Wikipedia-based semantic relatedness framework introduced in Chapter 4 is able to capture these biases in both cases. Furthermore, the structure and associations between Wikipedia categories and articles enables identification of topical drifts which also aids the reader in identification of a specific agenda[1]. Moreover, the task of filtering tweets relevant to an entity and its various reputation dimensions in Chapter 5 also contains implicit and subtle topics that we are able to identify via Wikipedia category and article associations.

Finally, the answers to above-listed research questions form the basis of the answer to the fundamental question in Section 1.3 of Chapter 1 whereby we sought to explore the structure and relationship between Wikipedia categories and articles, and our exploration yielded fruitful insights with successful outcomes in various text mining applications listed in Section 8.1.

8.4 Limitations

As with any human-curated effort Wikipedia despite its wide-scale coverage of knowledge has some limitations which affect the outcomes of this thesis. Below we list two significant limitations of the contributions of this thesis:

- Our phrase chunking strategy introduced in Chapter 4 may have tendency to miss out significant phrases on account of Wikipedia missing out some information on long-tail entities.
- The human-curated Wikipedia category-article structure contains some noisy relationships which affects the accuracy of relatedness measures of Chapters 5 and 6, and the richness of discovered communities in Chapter 7.

8.5 Future Directions

There are different research directions generated by the work in this thesis. We list some of these as follows:

[1]This is particularly true in the news domain.

- ***Utilising partial matching strategy for phrase chunking:*** As mentioned in Chapter 4 some significant phrases tend to be missed out on account of our exact matching strategy over Wikipedia article titles/redirects. Partial matching if employed can help increase coverage of identified phrases thereby leading to richness of our proposed framework.

- ***Combination of semantic relatedness measures driven from Wikipedia category-article structure with traditional text similarity measures:*** An interesting research direction worth exploring is utilisation of semantic relatedness in combination with traditional text similarity measures such as cosine similarity, jaccard similarity etc. to make stronger inferences from within textual data. This can help alleviate the limitation arising due to noise in Wikipedia category-article structure thereby assisting in addressing some limitations of noisy and limited datasets as for example in Chapter 7 whereby some academic collections had noisy titles.

- ***Exploring variants of semantic relatedness measures:*** We aim to make use of semantic relatedness measures of Chapter 4 in a more sophisticated manner such as utilising it in a probabilistic framework. Moreover, we aim to combine our relatedness measures with the traditional notions of path length from with text mining literature in an attempt to remove the effect of noise found in the Wikipedia category-article structure.

- ***Combination of community detection framework with relatedness measures:*** In the current form, the community detection framework introduced in Chapter 7 does not make use of semantic relatedness measures of Chapter 4. As future work, a combination of the two techniques can lead towards reduction of noise within the Wikipedia structure and as result achieve further improvement in various text mining applications.

Appendix A

WikiMadeEasy: A Programmer-Friendly API for Mining Wikipedia Data

A.1 Introduction

Wikipedia has emerged as an extremely useful knowledge resource that has found numerous applications in the areas of natural language processing, knowledge management, data mining and other research areas [138]. The strength of Wikipedia stems from the fact that it is human-curated and hence, it alleviates the need for customised manual annotations that are an essential component of numerous knowledge management and extraction applications. The immense popularity of Wikipedia as an online encyclopedia of concepts and semantic relations has led to the development of derived thesauri or ontologies such as DBPedia [22] and YAGO [195]. These however are limited in that they do not provide sufficient flexibility to programmers to incorporate efficient mining methods using the underlying Wikipedia data. Having access to raw data in Wikipedia (e.g., Wikipedia article titles, Wikipedia article and category hierarchy etc.) is crucial for modern text-mining applications [141, 147] that utilise machine learning methods over a number of raw textual features derived from Wikipedia articles and categories.

We present WikiMadeEasy, a programmer-friendly API enabling developers and researchers to mine the huge amount of knowledge encoded within the Wikipedia structure. The WikiMadeEasy API follows a client-server architecture so as to facilitate greater flexibility for the programmer through the: 1) ability to use the information returned from the Wikipedia knowledge-base as per need, and 2) ability to use any programming language of choice within the client once the server returns the

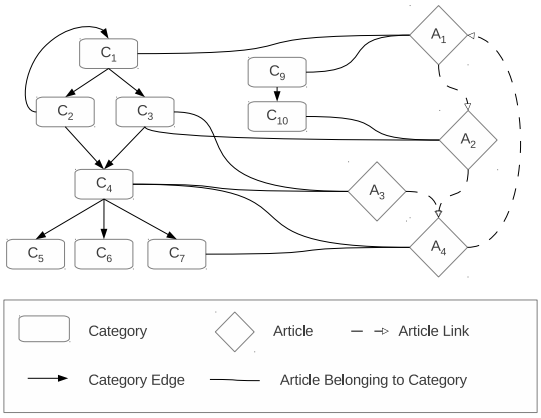

Figure A.1: Wikipedia Category Graph Structure along with Wikipedia Articles

information required by the programmer. WikiMadeEasy provides the core functionalities on top of Wikipedia data through the availability of easy-to-use client wrapper methods. Furthermore, unlike other similar tools, WikiMadeEasy provides the programmer with the ability to mine the rich graph structure of Wikipedia categories efficiently which in turn enables the development of novel natural language processing applications and three of these were covered in Chapters 5, 6, and 7 of this thesis.

A.2 Background: Wikipedia as a Knowledge Base

Wikipedia is a huge, rapidly evolving knowledge resource of interlinked textual concepts organized semantically into categories and articles. More specifically, Wikipedia is organized into categories in a taxonomical structure (see Figure A.1). Each Wikipedia category can have an arbitrary number of subcategories as well as being mentioned inside an arbitrary number of supercategories (e.g., category C_4 in Figure A.1 is a subcategory of C_2 and C_3, and a supercategory of C_5, C_6 and C_7.) Furthermore, in Wikipedia each article can belong to an arbitrary number of categories, where each category is a kind of semantic tag for that article [227]. As an example, in Figure 2 article A_1 belongs to categories C_1 and C_3, article A_2 belongs to categories C_3 and C_4, while article A_3 belongs to categories C_4 and C_7.

WikiMadeEasy enables developers and researchers to easily extract the interlinked data within the Wikipedia category and article graph via the methods explained in the next section. It differs significantly from other similar tools such as Wikipedia Miner [148] in that it does not require several machine hours of preprocessing for

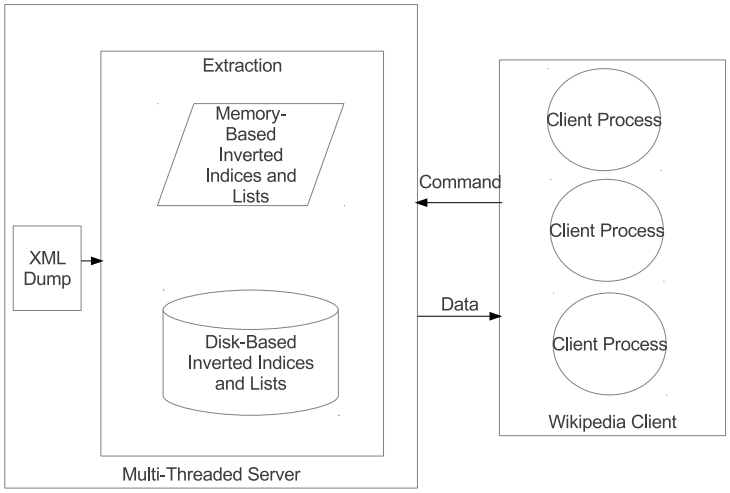

Figure A.2: WikiMadeEasy Architecture

usage of all its features[1].

A.3 Architecture and Functionality

Figure A.2 illustrates the overall architecture of WikiMadeEasy. It comprises a client and a server with the server running multiple threads so as to allow multiple clients to request data from it. Furthermore, separation of server and client allows programming flexibility thus enabling to use any programming language of choice within the client. The server includes an extraction module which extracts memory-based and disk-based inverted indexes and lists from Wikipedia dumps[2]. The WikiMadeEasy API provides flexibility to the programmer to opt for memory-based and/or disk-based structures according to available resources. The client is able to package requests into easy-to-use commands sent via sockets to the server and the server responds with requested data over the same socket connection.

The design of the interface to the Wikipedia server is centered around the object *wiki_client* shown in *line 1* of *Listing A.1*. The method *process* is invoked for *wiki_client* object each time a specific functionality is desired, and this desired functionality is passed as a parameter to the *wiki_client.process* method. The available

[1]Wikipedia Miner extracted data from full English Wikipedia in a little over 2.5 hours with cluster of 30 machines, each with two 2.66 GHz processors and 4 GB of RAM.

[2]We utilise the dumps made available by DBPedia [22].

functionalities within WikiMadeEasy are shown in lines *lines 2-13* of *Listing A.1.* The details of each line of code is explained as follows:

- Line 2: Returns true if the provided string (here, business) is the exact name of a Wikipedia article
- Line 3: Returns false if the provided string (here, abdus salam) is the name of a person mentioned on Wikipedia
- Line 4: Returns the list of categories in which the Wikipedia article (here, data mining) is mentioned.
- Line 5: Returns the list of all articles mentioned inside the given category (here, business).
- Line 6: Return the list of all categories that match the partially given string (here, pakistan).
- Line 7: Return the list of all articles that match the partially given string (here, computer science).
- Line 8: Returns outgoing links from the given Wikipedia article (here, pagerank) to other Wikipedia articles.
- Line 9: Returns ingoing links to the given Wikipedia article (here, google) from other Wikipedia articles.
- Line 10: Returns text within extended abstract of the given Wikipedia article (here, pakistan).
- Line 11: Returns the list of sub categories for the given Wikipedia category (here, science).
- Line 12: Returns the list of super categories for the given Wikipedia category (here, science).
- Line 13: Generates a graph between the given categories (here, information science and sociology) to given hop count (here, 2) depicting relations between their super categories. Figure 1 shows the generated graph corresponding to WikiMadeEasy code of *line 13*.

Listing A.1: Python Code Snippet for Usage of WikiMadeEasy

```
1 wiki_client = Wiki_client_service()
2 print wiki_client.process(['isTitle', 'business'])
3 print wiki_client.process(['isPerson', 'abdus salam'])
4 print wiki_client.process(['mentionInCategories', 'data
    mining'])
5 print wiki_client.process(['containsArticles', 'business'])
6 print wiki_client.process(['matchesCategories', 'pakistan'])
7 print wiki_client.process(['matchesArticles', 'computer
    science'])
```

```
8  print wiki_client.process(['getWikiOutlinks', 'pagerank'])
9  print wiki_client.process(['getWikiInlinks', 'google'])
10 print wiki_client.process(['getExtendedAbstract', 'pakistan
       '])
11 print wiki_client.process(['getSubCategory', 'science'])
12 print wiki_client.process(['getSuperCategory', 'science'])
13 graph_dict = wiki_client.process(['
       getSubtoSuperCategoryGraph', 'information science', '
       sociology', 2])
```

Furthermore, these commands are also supported by list-based operations i.e. one socket call can handle a command which is a list of commands thereby minimizing the network overload.

Appendix B

Use of Wikipedia Articles' Hyperlink for Filtering Task

B.1 Introduction

Here, we explain our previous approach for addressing the filtering task in the context of CLEF RepLab 2013 filtering task. The earlier technique like previous approaches in the literature utilises the Wikipedia disambiguation pages for an entity to determine the amount of disambiguation within a particular tweet while at the same time proposing a technique on top of Wikipedia hyperlink[1] structure to determine context of a tweet.

B.2 The Approach

In this section we present our strategy to exploit the Wikipedia articles' hyperlink structure; first we discuss phrase extraction which is followed by a discussion on how we actually exploit the Wikipedia articles' hyperlink structure.

B.2.1 Phrase Extraction from Tweets

There are two kinds of phrases that we extract in this step. First is the entity phrase which represents the entity while the rest of the phrases are context phrases. As an example, consider the tweets in Table B.1. For the first tweet, we extract all possible n-grams within the chunks "I prefer Samsung over HTC", "Apple", "Nokia", and "because it is economical and good". In this tweet, "Samsung" constitutes an entity phrase whereas other possible n-grams are considered as context phrases.

[1]Inlinks and outlinks within the Wikipedia articles.

Context phrase extraction is performed by the generation of possible n-grams within phrase chunks of a tweet. We do not perform n-gram generation for the complete tweet but instead treat a tweet as a composition of phrase chunks with boundaries such as commas, semi-colons, sentence terminators etc. along with other tweet-specific markers such as @, RT etc. We then reduce candidate phrases extracted from a tweet to those that have a match in Wikipedia article titles[2]. The reduced set of phrases extracted from a tweet are referred to as *ContextPhrases*. In Table B.1, considering the second tweet, we extract all possible n-grams within the chunks "Dear Ryanair", "I hate travelling with you", and "You suck." Note that we utilise the n-grams within tweets' phrase chunks for efficiency purposes in order to speed up the feature extraction process of the next step.

Entity	Tweet
Samsung	I prefer Samsung over HTC, Apple, Nokia because it is economical and good
Ryanair	Dear Ryanair, I hate travelling with you. You suck!!!

Table B.1: Example Tweets to Illustrate Phrase Extraction

B.2.2 Feature Extraction Using Wikipedia Articles' Hyperlinks

As described in the previous section, we extract an entity phrase and context phrases for each tweet which we now utilise in this section to generate features using the links between Wikipedia articles.

At the first level, we use the parent Wikipedia article for the entity under investigation[3] and we extract a set of parent categories that contain the entity name. For example, corresponding to entity "Toyota", the categories "Companies listed on the New York Stock Exchange", "Marine engine manufacturers", "Military equipment of Japan","Companies based in Nagoya" and "Toyota" occur as parent categories of which only "Toyota" is selected. We then extract sub-categories from the selected categories up to a depth count of two[4]; finally all articles belonging to these sub-

[2]Note that this step is similar to the one explained in Section 4.1 of Chapter 4 of this thesis.

[3]The parent Wikipedia article for each entity is given as part of the dataset for this task.

[4]This was chosen following empirical analysis; a depth of two was found sufficient to gather a representative set of categories while preventing too much drift.

categories are marked as being related to the entity under investigation and we refer these articles as $Articles_{related}$.

We then construct an information table of Wikipedia-based features using entity phrase, context phrases, and $Articles_{related}$ as follows:

- In order to perform entity disambiguation for the entity phrase, we extract from Wikipedia the disambiguation pages (senses) for an entity phrase and context phrases. Using these potential senses of the entity phrase (denoted as e_{s_i}) and each context phrase (denoted as c_{s_i}) we then define three collections or bags;
  - Wikipedia articles linking to e_{s_i} or any c_{s_i} referred to as *Inlinks*
  - Wikipedia articles linking from e_{s_i} or any c_{s_i} referred to as *Outlinks*
  - Wikipedia articles linking to/from e_{s_i} or any c_{s_i} referred to as *Inlinks+ Outlinks*
- Using information of *Inlinks*, *Outlinks* and *Inlinks+Outlinks*, we derive the features shown in Table B.2.

Feature	Description
$Intersection_{duplication}$	No. of intersections between *Inlinks* for e_{s_i} and each c_{s_i} without removing duplicated articles
$NormalizedIntersection_{duplication}$	No. of intersections between *Inlinks* for e_{s_i} and each c_{s_i} without removing duplicated articles and normalized by total number of articles in the sets
$Intersection_{noduplication}$	No. of intersections between *Inlinks* for e_{s_i} and each c_{s_i} after removing duplicated articles
$NormalizedIntersection_{noduplication}$	No. of intersections between *Inlinks* for e_{s_i} and each c_{s_i} without removing duplicated articles and normalized by total number of articles in the sets
$Ratio_{inlink:outlink}$	Ratio between articles in *Inlinks* to articles in *Outlinks*

Table B.2: Feature set for entity name disambiguation in tweets on top of Wikipedia Article Link Structure
*Note that we calculate similarly for *Outlinks* and *Inlinks+Outlinks* for the first four features.

- We illustrate the use of this entire feature set with the help of the example illustrated in Table B.3 depicting a tweet with three context phrases $c1$, $c2$, and $c3$. Here, the entity phrase e has three Wikipedia senses e_{s_1}, e_{s_2}, and e_{s_3}. There are

two senses corresponding to $c1$ (i.e, $c1_{s_1}$ and $c1_{s_2}$), three senses corresponding to $c2$ (i.e., $c2_{s_1}$, $c2_{s_2}$, and $c2_{s_3}$), and finally $c3$ which is unambiguous (i.e., has only one sense $c3_{s_1}$). We assume Table B.3 to represent $Intersection_{noduplication}$ i.e., the third feature from Table B.2 for *Inlinks*. Corresponding to each context phrase, the sense that maximizes $Intersection_{noduplication}\{Inlinks\}$ is chosen implying selection of $c1_{s_2}$ and $c2_{s_3}$ across e_{s_1}, $c1_{s_1}$ and $c2_{s_2}$ across e_{s_2} and finally, $c1_{s_1}$ and $c2_{s_3}$ across e_{s_3}; note that no reduction takes place for $c3$ on account of it having a single sense only. We show the reduction step in Table B.4. The reduction is followed by averaging the numerical values of features (i.e., $Intersection_{noduplication}\{Inlinks\}$ in the considered example) for selected context phrase sense across each entity phrase sense implying a value of 294 across e_{s_1}, 318.33 across e_{s_2}, and 323.67 across e_{s_3}. As a final step, we select the entity phrase sense with the highest context phrase score and in the considered example e_{s_3} (with value 323.67) is selected and the value of this score is added as a feature for the entity name disambiguation task.

Furthermore, if the selected entity corresponding to the highest score value belongs to one of the articles that are related to the entity (i.e., articles in $Articles_{related}$ explained previously in this section), we add a Boolean feature marked True, and False otherwise. Hence, for each feature listed in Table B.2 there are two associated features with one being a continuous variable (score) and the other being a discrete variable (Boolean value representing entity sense mapping). Note that this reduction of features is performed corresponding to each feature in Table B.2 and for the purpose of the example above we only use $Intersection_{noduplication}\{Inlinks\}$; similarly it is done for all three *Inlinks*, *Outlinks*, and *Inlinks+Outlinks*. We also do such feature set construction separately for stemmed and non-stemmed versions of the tweets.

Entity Senses	Context Phrase Senses					
	$c1$		$c2$			$c3$
	$c1_{s_1}$	$c1_{s_2}$	$c2_{s_1}$	$c2_{s_2}$	$c2_{s_3}$	$c3_{s_1}$
e_{s_1}	150	230	400	415	532	120
e_{s_2}	180	147	350	375	280	400
e_{s_3}	234	115	83	127	237	500

Table B.3: Information Table Corresponding to $Intersection_{noduplication}\{Inlinks\}$

Entity Senses	Context Phrase Senses		
	$c1$	$c2$	$c3$
e_{s_1}	$c1_{s_2}$:230	$c2_{s_3}$:532	120
e_{s_2}	$c1_{s_1}$:180	$c2_{s_2}$:375	400
e_{s_3}	$c1_{s_1}$:234	$c2_{s_3}$:237	500

Table B.4: Reduction of Information Table Corresponding to $Intersection_{noduplication}\{Inlinks\}$

Appendix C

Publications

Following are the list of papers which form the main body of the thesis.

- Qureshi, M. A., ORiordan, C., & Pasi, G. (2014). Exploiting Wikipedia for Entity Name Disambiguation in Tweets. In Natural Language Processing and Information Systems (pp. 184-195). Springer International Publishing.
- Qureshi, M. A., O'Riordan, C., & Pasi, G. (2014). A perspective-aware approach to search: visualizing perspectives in news search results. In Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval (pp. 1261-1262). ACM.
- Qureshi, M. A., ORiordan, C., & Pasi, G. (2014). Exploiting Wikipedia to Identify Domain-Specific Key Terms/Phrases from a Short-Text Collection. In Italian Information Retrieval Workshop (pp. 63-74).
- Qureshi, M. A., Younus, A., ORiordan, C., & Pasi, G (2014). CIRGIRDISCO at RepLab2014 Reputation Dimension Task: Using Wikipedia Graph Structure for Classifying the Reputation Dimension of a Tweet. CLEF (Online Working Notes/Labs/Workshop).
- Qureshi, M. A., Younus, A., Abril, D., ORiordan, C., & Pasi, G. (2013). CIRG_IRGDISCO at RepLab2013 Filtering Task: Use of Wikipedia's Graph Structure for Entity Name Disambiguation in Tweets. In CLEF (Online Working Notes/Labs/Workshop).
- Qureshi, M. A., Younus, A., O'Riordan, C., Pasi, G., & Touheed, N. (2013). A System for Perspective Aware Search. European Workshop on Human-Computer Interaction and Information Retrieval (EuroHCIR2013) co-located with SIGIR 2013 (pp. 55-58).

- Qureshi, M. A., O'Riordan, C., & Pasi, G. (2012, October). Short-text domain specific key terms/phrases extraction using an n-gram model with wikipedia. In Proceedings of the 21st ACM international conference on Information and knowledge management (pp. 2515-2518). ACM.

Other papers published during thesis — related topics in *Information Retrieval* or *Social Media Analytics* but not central to this thesis.

- Qureshi, M.A., Younus, A., Griffith, J., O'Riordan, C., & Pasi, G. (2015). Company Name Disambiguation in Tweets: a Two-Step Filtering Approach. In AIRS 2015 (Approved for publishing)

- Qureshi, M.A., Younus, A., Griffith, J., O'Riordan, C., Pasi, G., & Meguebli, Y. (2015). NewsOpinionSummarizer: A Visualization and Predictive System for Opinion Pieces in Online News. In Web Intelligence Conference. (Approved for publishing)

- Younus, A., Qureshi, M.A., Griffith, J., O'Riordan, C., & Pasi, G. (2015). A Study into the Correlation between Narcissism and Facebook Communication Patterns. In Web Intelligence Conference. (Approved for publishing)

- Qureshi, M. A., Younus, A., Yousuf, M., Moiz, A., Saeed, M., Touheed, N., O'Riordan, C., & Pasi, G. (2014). YummyKarachi: Using Real-Time Tweets for Restaurant Recommendations in an Unsafe Location. In UMAP Posters, Demonstrations and Late-breaking Results (pp. 49-52).

- Younus, A., Qureshi, M. A., Saeed, M., Touheed, N., O'Riordan, C., & Pasi, G. (2014). Election trolling: analyzing sentiment in tweets during pakistan elections 2013. In Proceedings of the companion publication of the 23rd international conference on World wide web companion (pp. 411-412). International World Wide Web Conferences Steering Committee.

- Qureshi, M. A., Younus, A., ORiordan, C., & Pasi, G. (2013). CIRGIRDISCO at TREC 2013 Contextual Suggestion Track: Using the Wikipedia Graph Structure for Item-to-Item Recommendation. TREC 2013 Contextual Suggestion Track.

- Qureshi, M. A., ORiordan, C., & Pasi, G. (2013). CIRGIRDISCO at TREC 2013 Microblog Track. TREC 2013 Microblog Track.

- Younus, A., Qureshi, M. A., ORiordan, C., & Pasi, G. (2013). Personalization for difficult queries. Workshop on Modeling User Behavior for Information Retrieval Evaluation, 15-16.

- Qureshi, M. A., ORiordan, C., & Pasi, G. (2013). Clustering with Error-Estimation for Monitoring Reputation of Companies on Twitter. In Information Retrieval Technology (pp. 170-180). Springer Berlin Heidelberg.

- Qureshi, M. A., O'Riordan, C., & Pasi, G. (2012). Concept Term Expansion Approach for Monitoring Reputation of Companies on Twitter. In CLEF (Online Working Notes/Labs/Workshop).

- Younus, A., Qureshi, M. A., Kingrani, S. K., Saeed, M., Touheed, N., O'Riordan, C., & Gabriella, P. (2012). Investigating bias in traditional media through social media. In Proceedings of the 21st international conference companion on World Wide Web (pp. 643-644). ACM.

- Qureshi, M. A., Younus, A., Soon, L. K., Saeed, M., Touheed, N., ORiordan, C., & Gabriella, P. (2012). Traces of Social Media Activism from Malaysia and Pakistan. Web Science track co-located with 21st International World Wide Web Conference (Additional Companion Archived on Conference Website).

Bibliography

[1] Sisay Fissaha Adafre and Maarten De Rijke. Finding similar sentences across multiple languages in wikipedia. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 62–69, 2006.

[2] Charu C Aggarwal and ChengXiang Zhai. *Mining text data*. Springer Science & Business Media, 2012.

[3] Eugene Agichtein, Eric Brill, and Susan Dumais. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06. ACM.

[4] Eneko Agirre and Aitor Soroa. Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 33–41, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[5] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 5–14. ACM, 2009.

[6] Rodrigo Aldecoa and Ignacio Marín. Exploring the limits of community detection strategies in complex networks. *Scientific reports*, 3, 2013.

[7] Enrique Alfonseca and Suresh Manandhar. An unsupervised method for general named entity recognition and automated concept discovery. In *Proceedings of the 1st international conference on general WordNet, Mysore, India*, pages 34–43, 2002.

[8] Enrique Amigo, Jorge Carrillo de Albornoz, Irina Chugur, Adolfo Corujo, Julio Gonzalo, Tamara Martin, Edgar Meij, Maarten de Rijke, and Damiano Spina.

Overview of replab 2013: Evaluating online reputation monitoring systems. In *CLEF 2013 Labs and Workshop Notebook Papers*, Springer LNCS, 2013.

[9] Enrique Amigó, Jorge Carrillo-de Albornoz, Irina Chugur, Adolfo Corujo, Julio Gonzalo, Edgar Meij, Maarten de Rijke, and Damiano Spina. Overview of replab 2014: author profiling and reputation dimensions for online reputation management. In *Information Access Evaluation. Multilinguality, Multimodality, and Interaction*, pages 307–322. Springer, 2014.

[10] Enrique Amigó, Adolfo Corujo, Julio Gonzalo, Edgar Meij, and Maarten de Rijke. Overview of replab 2012: Evaluating online reputation management systems. In *CLEF 2012 Labs and Workshop Notebook Papers*, 2012.

[11] Enrique Amigó, Julio Gonzalo, and Felisa Verdejo. A General Evaluation Measure for Document Organization Tasks. In *Proceedings SIGIR 2013*, pages 643–652, July.

[12] Jisun An, Meeyoung Cha, P Krishna Gummadi, and Jon Crowcroft. Media landscape in twitter: A world of new conventions and political diversity. In *ICWSM*, 2011.

[13] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. *Dbpedia: A nucleus for a web of open data*. Springer, 2007.

[14] Ricardo Baeza-Yates and Carlos Castillo. Crawling the infinite web: five levels are enough. In *In Proceedings of the third Workshop on Web Graphs (WAW)*, pages 156–167. Springer, 2004.

[15] Breck Baldwin and Thomas S Morton. Dynamic coreference-based summarization. In *EMNLP*, pages 1–6, 1998.

[16] L Baleydier. The kartoo visual metasearch engine. 2008.

[17] Ken Barker and Nadia Cornacchia. Using noun phrase heads to extract document keyphrases. In *Advances in Artificial Intelligence*, pages 40–52. Springer, 2000.

[18] Regina Barzilay and Kathleen R McKeown. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328, 2005.

[19] S. Bergamaschi, F. Guerra, and B. Leiba. Guest editors' introduction: Information overload. *Internet Computing, IEEE*, 14(6):10–13, Nov 2010.

[20] Shane Bergsma, Mark Dredze, Benjamin Van Durme, Theresa Wilson, and David Yarowsky. Broadly improving user classification via communication-based name and location clustering on twitter. In *HLT-NAACL*, pages 1010–1019, 2013.

[21] Daniel M Bikel, Richard Schwartz, and Ralph M Weischedel. An algorithm that learns what's in a name. *Machine learning*, 34(1-3):211–231, 1999.

[22] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. Dbpedia - a crystallization point for the web of data. *Web Semant.*, 7(3):154–165, September 2009.

[23] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. Dbpedia-a crystallization point for the web of data. *Web Semantics: science, services and agents on the world wide web*, 7(3):154–165, 2009.

[24] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[25] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.

[26] Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. Measuring semantic similarity between words using web search engines. *www*, 7:757–766, 2007.

[27] Erik Borra and Ingmar Weber. Political insights: Exploring partisanship in web search queries. *First Monday*, 17(7), 2012.

[28] Andrew Borthwick. *A maximum entropy approach to named entity recognition.* PhD thesis, New York University, 1999.

[29] Adrien Bougouin, Florian Boudin, and Béatrice Daille. Topicrank: Graph-based topic ranking for keyphrase extraction. In *International Joint Conference on Natural Language Processing (IJCNLP)*, pages 543–551, 2013.

[30] Ronald Brandow, Karl Mitze, and Lisa F Rau. Automatic condensation of electronic publications by sentence selection. *Information Processing & Management*, 31(5):675–685, 1995.

[31] Sergey Brin. Extracting patterns and relations from the world wide web. In *The World Wide Web and Databases*, pages 172–183. Springer, 1999.

[32] Marc Bron, Jasmijn Van Gorp, Frank Nack, Maarten de Rijke, Andrei Vishneuski, and Sonja de Leeuw. A subjunctive exploratory search interface to support media studies researchers. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 425–434. ACM, 2012.

[33] Ceren Budak, Sharad Goel, and Justin M Rao. Fair and balanced? quantifying media bias through crowdsourced content analysis. *Quantifying Media Bias Through Crowdsourced Content Analysis (November 17, 2014)*, 2014.

[34] Alexander Budanitsky and Graeme Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47, 2006.

[35] Razvan C Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. In *EACL*, volume 6, pages 9–16, 2006.

[36] Silvia Calegari and Gabriella Pasi. Personal ontologies: Generation of user profiles based on the yago ontology. *Information processing & management*, 49(3):640–658, 2013.

[37] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 335–336, New York, NY, USA, 1998. ACM.

[38] Yllias Chali and Maheedhar Kolla. University of lethridge summarizer at duc04. In *Proceedings of the Document Understanding Conference (DUC 2004), Boston, USA*, 2004.

[39] Mo Chen, Jian-Tao Sun, Hua-Jun Zeng, and Kwok-Yan Lam. A practical system of keyphrase extraction for web pages. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 277–278. ACM, 2005.

[40] Xiao Cheng and Dan Roth. Relational inference for wikification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1787–1796, 2013.

[41] Paul Alexandru Chirita, Wolfgang Nejdl, Raluca Paiu, and Christian Kohlschütter. Using odp metadata to personalize search. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 178–185, 2005.

[42] Charles LA Clarke, Maheedhar Kolla, Gordon V Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 659–666. ACM, 2008.

[43] Kevin A Clauson, Hyla H Polen, Maged N Kamel Boulos, and Joan H Dzenowagis. Scope, completeness, and accuracy of drug information in wikipedia. *Annals of Pharmacotherapy*, 42(12):1814–1821, 2008.

[44] Michael Collins and Yoram Singer. Unsupervised models for named entity classification. In *Proceedings of the joint SIGDAT conference on empirical methods in natural language processing and very large corpora*, pages 100–110. Citeseer, 1999.

[45] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.

[46] Irene Cramer. How well do semantic relatedness measures perform?: a meta-study. In *Proceedings of the 2008 Conference on Semantics in Text Processing*, pages 59–70. Association for Computational Linguistics, 2008.

[47] Alessandro Cucchiarelli and Paola Velardi. Unsupervised named entity recognition using syntactic and semantic contextual evidence. *Computational Linguistics*, 27(1):123–131, 2001.

[48] Silviu Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*, volume 7, pages 708–716, 2007.

[49] James R Curran and Stephen Clark. Language independent ner using a maximum entropy tagger. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 164–167. Association for Computational Linguistics, 2003.

[50] Ido Dagan, Lillian Lee, and Fernando CN Pereira. Similarity-based models of word cooccurrence probabilities. *Machine Learning*, 34(1-3):43–69, 1999.

[51] Alexandre Davis, Adriano Veloso, Altigran S da Silva, Wagner Meira Jr, and Alberto HF Laender. Named entity disambiguation in streaming data. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 815–824. Association for Computational Linguistics, 2012.

[52] Munmun De Choudhury, Nicholas Diakopoulos, and Mor Naaman. Unfolding the event landscape on twitter: classification and exploration of user categories. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 241–244. ACM, 2012.

[53] Chrysanthos Dellarocas, Neveen Farag Awad, and Xiaoquan (Michael) Zhang. Exploring the value of online reviews to organizations: Implications for revenue forecasting and planning. In *MANAGEMENT SCIENCE*, pages 1407–1424, 2003.

[54] Gianluca Demartini. Ares: A retrieval engine based on sentiments. In *Advances in Information Retrieval*, pages 772–775. Springer, 2011.

[55] Mira Dontcheva, Steven M Drucker, Geraldine Wade, David Salesin, and Michael F Cohen. Summarizing personal web browsing sessions. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 115–124. ACM, 2006.

[56] Shiri Dori-Hacohen and James Allan. Automated controversy detection on the web. In *Advances in Information Retrieval*, pages 423–434. Springer, 2015.

[57] Marina Drosou and Evaggelia Pitoura. Search result diversification. *SIGMOD Rec.*, 39(1):41–47, September 2010.

[58] Philip Edmonds and Scott Cotton. senseval-2: Overview. In *The Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 1–5. Association for Computational Linguistics, 2001.

[59] Harold P Edmundson. New methods in automatic extracting. *Journal of the ACM (JACM)*, 16(2):264–285, 1969.

[60] Ofer Egozi, Shaul Markovitch, and Evgeniy Gabrilovich. Concept-based information retrieval using explicit semantic analysis. *ACM Transactions on Information Systems (TOIS)*, 29(2):8, 2011.

[61] Günes Erkan and Dragomir R Radev. Lexrank: graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, pages 457–479, 2004.

[62] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165(1):91–134, 2005.

[63] Ronen Feldman and Ido Dagan. Knowledge discovery in textual databases (kdt). In *KDD*, volume 95, pages 112–117, 1995.

[64] Paolo Ferragina and Ugo Scaiella. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). CIKM '10, pages 1625–1628, New York, NY, USA, 2010. ACM.

[65] David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79, 2010.

[66] J.L. Fleiss et al. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.

[67] Charles Fombrun and Mark Shanley. What's in a name? reputation building and corporate strategy. *Academy of management Journal*, 33(2):233–258, 1990.

[68] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010.

[69] Eibe Frank, Gordon W Paynter, Ian H Witten, Carl Gutwin, and Craig G Nevill-Manning. Domain-specific keyphrase extraction. In *Proceedings of 16th International Joint Conference on Artificial Intelligence*, page 668673. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[70] Evgeniy Gabrilovich and Shaul Markovitch. Feature generation for text categorization using world knowledge. In *IJCAI*, volume 5, pages 1048–1053, 2005.

[71] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, volume 7, pages 1606–1611, 2007.

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

[73] 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.

[74] Anna Lisa Gentile, Ziqi Zhang, Lei Xia, and José Iria. Semantic relatedness approach for named entity disambiguation. In *Digital libraries*, pages 137–148. Springer, 2010.

[75] Jim Giles. Internet encyclopaedias go head to head. *Nature*, 438(7070):900–901, 2005.

[76] Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 42–47. Association for Computational Linguistics, 2011.

[77] Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. Part-of-speech tagging for twitter: annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*, HLT '11, pages 42–47, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[78] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[79] Natalie Glance, Matthew Hurst, Kamal Nigam, Matthew Siegler, Robert Stockton, and Takashi Tomokiyo. Deriving marketing intelligence from online discussion. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, pages 419–428, 2005.

[80] Michael Granitzer, Wolfgang Kienreich, Vedran Sabol, Keith Andrews, and Werner Klieber. Evaluating a system for interactive exploration of large, hierarchically structured document repositories. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, pages 127–134. IEEE, 2004.

[81] Maria Grineva, Maxim Grinev, and Dmitry Lizorkin. Extracting key terms from noisy and multitheme documents. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 661–670, New York, NY, USA, 2009. ACM.

[82] Ralph Grishman and Beth Sundheim. Message understanding conference-6: A brief history. In *COLING*, volume 96, pages 466–471, 1996.

[83] Tim Groseclose and Jeffrey Milyo. A measure of media bias. *The Quarterly Journal of Economics*, pages 1191–1237, 2005.

[84] Yuhang Guo, Wanxiang Che, Ting Liu, and Sheng Li. A graph-based method for entity linking. In *IJCNLP*, pages 1010–1018, 2011.

[85] M.B. Habib and M. van Keulen. A generic open world named entity disambiguation approach for tweets. In *5th International Conference on Knowledge Discovery and Information Retrieval, KDIR 2013*. SciTePress, September 2013.

[86] Viktor Hangya and Richárd Farkas. Filtering and polarity detection for reputation management on tweets. In *CLEF 2013 Eval. Labs and Workshop Online Working Notes*, 2013.

[87] Kazi Saidul Hasan and Vincent Ng. Conundrums in unsupervised keyphrase extraction: making sense of the state-of-the-art. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 365–373, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[88] Kazi Saidul Hasan and Vincent Ng. Automatic keyphrase extraction: A survey of the state of the art. *Proceedings of the Association for Computational Linguistics (ACL), Baltimore, Maryland: Association for Computational Linguistics*, 2014.

[89] Samer Hassan and Rada Mihalcea. Semantic relatedness using salient semantic analysis. In *AAAI*, 2011.

[90] Taher H Haveliwala. Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web*, pages 517–526. ACM, 2002.

[91] Marti Hearst. *Search user interfaces*. Cambridge University Press, 2009.

[92] Marti A Hearst. Clustering versus faceted categories for information exploration. *Communications of the ACM*, 49(4):59–61, 2006.

[93] Marti A. Hearst. *Search User Interfaces*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[94] Marti A. Hearst. 'natural' search user interfaces. *Commun. ACM*, 54(11):60–67, November 2011.

[95] Marti A Hearst and Jan O Pedersen. Visualizing information retrieval results: a demonstration of the tilebar interface. In *Conference Companion on Human Factors in Computing Systems*, pages 394–395, 1996.

[96] Brent Hecht and Martin Raubal. Geosr: Geographically explore semantic relations in world knowledge. In *The European Information Society*, pages 95–113. Springer, 2008.

[97] Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. Kore: Keyphrase overlap relatedness for entity disambiguation. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, 2012.

[98] Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 3161–3165. AAAI Press, 2013.

[99] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

[100] Jian Hu, Lujun Fang, Yang Cao, Hua-Jun Zeng, Hua Li, Qiang Yang, and Zheng Chen. Enhancing text clustering by leveraging wikipedia semantics. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 179–186. ACM, 2008.

[101] Jian Hu, Gang Wang, Fred Lochovsky, Jian-tao Sun, and Zheng Chen. Understanding user's query intent with wikipedia. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, 2009.

[102] Xia Hu, Lei Tang, and Huan Liu. Enhancing accessibility of microblogging messages using semantic knowledge. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 2465–2468, 2011.

[103] Xiaohua Hu, Xiaodan Zhang, Caimei Lu, E. K. Park, and Xiaohua Zhou. Exploiting wikipedia as external knowledge for document clustering. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 389–396, 2009.

[104] Anette Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 216–223. Association for Computational Linguistics, 2003.

[105] Anette Hulth. Enhancing linguistically oriented automatic keyword extraction. In *Proceedings of HLT-NAACL 2004: Short Papers*, pages 17–20. Association for Computational Linguistics, 2004.

[106] James G Hutton, Michael B Goodman, Jill B Alexander, and Christina M Genest. Reputation management: the new face of corporate public relations? *Public Relations Review*, 27(3):247–261, 2001.

[107] Wouter IJntema, Frank Goossen, Flavius Frasincar, and Frederik Hogenboom. Ontology-based news recommendation. In *Proceedings of the 2010 EDBT/ICDT Workshops*, EDBT '10, 2010.

[108] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[109] Bernard J Jansen, Mimi Zhang, Kate Sobel, and Abdur Chowdury. Twitter power: Tweets as electronic word of mouth. *Journal of the American society for information science and technology*, 60(11):2169–2188, 2009.

[110] Mario Jarmasz. Roget's thesaurus as a lexical resource for natural language processing. *arXiv preprint arXiv:1204.0140*, 2012.

[111] Xin Jiang, Yunhua Hu, and Hang Li. A ranking approach to keyphrase extraction. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 756–757. ACM, 2009.

[112] Diane Kelly. Methods for evaluating interactive information retrieval systems with users. *Foundations and Trends in Information Retrieval*, 3(12):1–224, 2009.

[113] Su Nam Kim and Timothy Baldwin. Extracting keywords from multi-party live chats. In *Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation*, pages 199–208, 2012.

[114] Su Nam Kim and Min-Yen Kan. Re-examining automatic keyphrase extraction approaches in scientific articles. In *Proceedings of the workshop on multiword expressions: Identification, interpretation, disambiguation and applications*, pages 9–16. Association for Computational Linguistics, 2009.

[115] Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. Automatic keyphrase extraction from scientific articles. *Language resources and evaluation*, 47(3):723–742, 2013.

[116] Kevin Knight and Daniel Marcu. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1):91–107, 2002.

[117] Robert R Korfhage. To see, or not to see–is that the query? In *Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 134–141. ACM, 1991.

[118] Alexander Kotov and ChengXiang Zhai. Tapping into knowledge base for concept feedback: Leveraging conceptnet to improve search results for difficult queries. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pages 403–412, 2012.

[119] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 457–466. ACM, 2009.

[120] Julian Kupiec, Jan Pedersen, and Francine Chen. A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 68–73. ACM, 1995.

[121] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM, 2010.

[122] José Paulo Leal, Vânia Rodrigues, and Ricardo Queirós. Computing semantic relatedness using dbpedia. 2012.

[123] Lillian Lee. Measures of distributional similarity. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 25–32. Association for Computational Linguistics, 1999.

[124] Douglas B Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.

[125] Lishuang Li, Rongpeng Zhou, and Degen Huang. Two-phase biomedical named entity recognition using crfs. *Computational biology and chemistry*, 33(4):334–338, 2009.

[126] Yang Li, Chi Wang, Fangqiu Han, Jiawei Han, Dan Roth, and Xifeng Yan. Mining evidences for named entity disambiguation. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1070–1078. ACM, 2013.

[127] Chin-Yew Lin and Eduard Hovy. Identifying topics by position. In *Proceedings of the fifth conference on Applied natural language processing*, pages 283–290. Association for Computational Linguistics, 1997.

[128] Feifan Liu, Deana Pennell, Fei Liu, and Yang Liu. Unsupervised approaches for automatic keyword extraction using meeting transcripts. In *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, pages 620–628. Association for Computational Linguistics, 2009.

[129] Zhiyuan Liu, Xinxiong Chen, Yabin Zheng, and Maosong Sun. Automatic keyphrase extraction by bridging vocabulary gap. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, CoNLL '11, pages 135–144, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[130] Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 257–266, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[131] Patrice Lopez and Laurent Romary. Humb: Automatic key term extraction from scientific articles in grobid. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 248–251. Association for Computational Linguistics, 2010.

[132] Hans Peter Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165, 1958.

[133] Inderjeet Mani and Eric Bloedorn. Multi-document summarization by graph search and matching. *arXiv preprint cmp-lg/9712004*, 1997.

[134] Gary Marchionini. Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4):41–46, 2006.

[135] Y. Matsuo and M. Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13:2004, 2004.

[136] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons.

In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 188–191. Association for Computational Linguistics, 2003.

[137] Paul McNamee and James Mayfield. Entity extraction without language-specific resources. In *proceedings of the 6th conference on Natural language learning-Volume 20*, pages 1–4. Association for Computational Linguistics, 2002.

[138] Olena Medelyan, Eibe Frank, and Ian H Witten. Human-competitive tagging using automatic keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1318–1327. Association for Computational Linguistics, 2009.

[139] Olena Medelyan, Ian H Witten, and David Milne. Topic indexing with wikipedia. In *Proceedings of the AAAI WikiAI workshop*, pages 19–24, 2008.

[140] Edgar Meij, Krisztian Balog, and Daan Odijk. Entity linking and retrieval. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 1127–1127. ACM, 2013.

[141] Edgar Meij, Wouter Weerkamp, and Maarten de Rijke. Adding semantics to microblog posts. WSDM '12, pages 563–572, New York, NY, USA, 2012. ACM.

[142] Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 1–8. ACM, 2011.

[143] Donald Metzler, Susan Dumais, and Christopher Meek. Similarity measures for short segments of text. In *Advances in Information Retrieval*, pages 16–27. Springer, 2007.

[144] Rada Mihalcea and Andras Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242. ACM, 2007.

[145] Rada Mihalcea and Paul Tarau. TextRank: Bringing Order into Texts. In *Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain, 2004.

[146] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[147] David Milne and Ian H. Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 509–518, New York, NY, USA, 2008. ACM.

[148] David Milne and Ian H. Witten. An open-source toolkit for mining wikipedia. *Artificial Intelligence*, 194(0):222 – 239, 2013.

[149] David N Milne. *Applying Wikipedia to Interactive Information Retrieval*. PhD thesis, University of Waikato, 2010.

[150] Abbe Mowshowitz and Akira Kawaguchi. Measuring search engine bias. *Information Processing & Management*, 41(5):1193–1205, 2005.

[151] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[152] Roberto Navigli. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2):10, 2009.

[153] Roberto Navigli and Mirella Lapata. An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(4):678–692, April 2010.

[154] Mark EJ Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006.

[155] Thuy Dung Nguyen and Min-Yen Kan. Keyphrase extraction in scientific publications. In *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*, pages 317–326. Springer, 2007.

[156] Tien T. Nguyen, Pik-Mai Hui, F. Maxwell Harper, Loren Terveen, and Joseph A. Konstan. Exploring the filter bubble: The effect of using recommender systems on content diversity. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 677–686, 2014.

[157] Daan Odijk, Cristina Garbacea, Thomas Schoegje, Laura Hollink, Victor de Boer, Kees Ribbens, and Jacco van Ossenbruggen. Supporting exploration of historical perspectives across collections. In *Research and Advanced Technology for Digital Libraries*. Springer, 2015.

[158] RichardJayadi Oentaryo, Jia-Wei Low, and Ee-Peng Lim. Chalk and cheese in twitter: Discriminating personal and organization accounts. In *Advances in Information Retrieval*, volume 9022, pages 465–476. 2015.

[159] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report SIDL-WP-1999-0120, Department of Computer Science, Stanford University, 1998.

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

[161] Marius Pasca, Dekang Lin, Jeffrey Bigham, Andrei Lifchits, and Alpa Jain. Organizing and searching the world wide web of facts-step one: the one-million fact extraction challenge. In *AAAI*, volume 6, pages 1400–1405, 2006.

[162] Alexandre Passant. Measuring semantic distance on linking data and using it for resources recommendations. In *AAAI Spring Symposium: Linked Data Meets Artificial Intelligence*, volume 77, page 123, 2010.

[163] Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the 4th International Conference on Computational Linguistics and Intelligent Text Processing*, pages 241–257. Springer-Verlag, 2003.

[164] Maria-Hendrike Peetz, Damiano Spina, Julio Gonzalo, and Maarten de Rijke. Towards an active learning system for company name disambiguation in microblog streams. In *CLEF (Online Working Notes/Labs/Workshop)*, 2013.

[165] Marco Pennacchiotti and Ana-Maria Popescu. Democrats, republicans and starbucks afficionados: user classification in twitter. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 430–438. ACM, 2011.

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

[167] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. In *Computer and Information Sciences-ISCIS 2005*, pages 284–293. Springer, 2005.

[168] M Atif Qureshi, Colm O'Riordan, and Gabriella Pasi. Short-text domain specific key terms/phrases extraction using an n-gram model with wikipedia. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2515–2518. ACM, 2012.

[169] M Atif Qureshi, Colm ORiordan, and Gabriella Pasi. Exploiting wikipedia to identify domain-specific key terms/phrases from a short-text collection. In *Italian Information Retrieval Workshop*, pages 63–74, 2014.

[170] M Atif Qureshi, Arjumand Younus, Lay-Ki Soon, Muhammad Saeed, Nasir Touheed, Colm ORiordan, and Pasi Gabriella. Traces of social media activism from malaysia and pakistan. In *1st Web Science track co-located with 21st International World Wide Web Conference (WWW 2012)*, 2011.

[171] Muhammad Atif Qureshi, Colm O'Riordan, and Gabriella Pasi. Exploiting wikipedia for entity name disambiguation in tweets. In *Natural Language Processing and Information Systems - 19th International Conference on Applications of Natural Language to Information Systems, NLDB 2014, Montpellier, France, June 18-20, 2014. Proceedings*, pages 184–195, 2014.

[172] Muhammad Atif Qureshi, Colm O'Riordan, and Gabriella Pasi. A perspective-aware approach to search: visualizing perspectives in news search results. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 1261–1262. ACM, 2014.

[173] Muhammad Atif Qureshi, Arjumand Younus, Daniel Abril, Colm O'Riordan, and Gabriella Pasi. Cirg irdisco at replab2013 filtering task: Use of wikipedia's graph structure for entity name disambiguation in tweets. In *CLEF (Online Working Notes/Labs/Workshop)*, 2013.

[174] Dragomir R Radev, Hongyan Jing, and Malgorzata Budzikowska. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *Proceedings of the 2000 NAACL-ANLP Workshop on Automatic summarization*, pages 21–30. Association for Computational Linguistics, 2000.

[175] Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. A word at a time: computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th international conference on World wide web*, pages 337–346. ACM, 2011.

[176] Delip Rao, David Yarowsky, Abhishek Shreevats, and Manaswi Gupta. Classifying latent user attributes in twitter. In *Proceedings of the 2nd international workshop on Search and mining user-generated contents*, pages 37–44. ACM, 2010.

[177] Lisa F Rau and Paul S Jacobs. Creating segmented databases from free text for text retrieval. In *Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 337–346. ACM, 1991.

[178] Ellen Riloff, Rosie Jones, et al. Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI/IAAI*, pages 474–479, 1999.

[179] S. E. Robertson, S. Walker, M.M. Hancock-Beaulieu, and M. Gatfor. Okapi at trec-3. In *The Third Text REtrieval Conference TREC-3*, pages 21–30, 1995.

[180] Peter Roget. *Rogets Thesaurus of English Words and Phrases*. Longman Group Ltd, 1852.

[181] Roy Rosenzweig. Can history be open source? wikipedia and the future of the past. *Journal of American History*, 93(1):117–146, 2006.

[182] Martin Rosvall and Carl T. Bergstrom. Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems. *PLoS ONE*, 6(4):e18209, 04 2011.

[183] Mehran Sahami and Timothy D Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the 15th international conference on World Wide Web*, pages 377–386. ACM, 2006.

[184] Pedro Saleiro, Luis Rei, Arian Pasquali, Carlos Soares, Jorge Teixeira, Fábio Pinto, Mohammad Nozari, Catarina Félix, and Pedro Strecht. Popstar at replab 2013: Name ambiguity resolution on twitter. In *CLEF 2013 Eval. Labs and Workshop Online Working Notes*, 2013.

[185] Gerard Salton. The state of retrieval system evaluation. *Information processing & management*, 28(4):441–449, 1992.

[186] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. In *INFORMATION PROCESSING AND MANAGEMENT*, pages 513–523, 1988.

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

[188] Rodrygo L.T. Santos, Craig Macdonald, and Iadh Ounis. Intent-aware search result diversification. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 595–604, 2011.

[189] Rodrygo LT Santos, Craig Macdonald, and Iadh Ounis. Search result diversification. *Foundations and Trends in Information Retrieval*, 9(1):1–90, 2015.

[190] Amit Singhal, Chris Buckley, and Mandar Mitra. Pivoted document length normalization. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 21–29. ACM, 1996.

[191] Damiano Spina. *Entity-based filtering and topic detection For online reputation monitoring in Twitter*. PhD thesis, Universidad Nacional de Educación a Distancia, 2014.

[192] Damiano Spina, Julio Gonzalo, and Enrique Amigó. Learning similarity functions for topic detection in online reputation monitoring. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 527–536. ACM, 2014.

[193] Anselm Spoerri. Infocrystal: A visual tool for information retrieval & management. In *Proceedings of the second international conference on Information and knowledge management*, pages 11–20, 1993.

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

[195] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM, 2007.

[196] Beth M Sundheim. Overview of results of the muc-6 evaluation. In *Proceedings of a workshop on held at Vienna, Virginia: May 6-8, 1996*, pages 423–442. Association for Computational Linguistics, 1996.

[197] Gabriela Tavares and Aldo Faisal. Scaling-laws of human broadcast communication enable distinction between human, corporate and robot twitter users. *PloS one*, 8(7), 2013.

[198] Simone Teufel and Marc Moens. Sentence extraction as a classification task. In *Proceedings of the ACL*, volume 97, pages 58–65, 1997.

[199] Anastasios Tombros and Mark Sanderson. Advantages of query biased summaries in information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 2–10. ACM, 1998.

[200] Antonio Toral and Rafael Munoz. A proposal to automatically build and maintain gazetteers for named entity recognition by using wikipedia. *NEW TEXT Wikis and blogs and other dynamic text sources*, 56, 2006.

[201] Daniel Tunkelang. Faceted search. *Synthesis lectures on information concepts, retrieval, and services*, 1(1):1–80, 2009.

[202] Peter Turney. Learning to extract keyphrases from text. Technical report, National Research Council Canada, Institute for Information Technology, Technical Report ERB-1057, 1999.

[203] Peter Turney. Coherent keyphrase extraction via web mining. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 434–439, 2003.

[204] Peter Turney. Measuring semantic similarity by latent relational analysis. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland*, pages 1136–1141, 2005.

[205] Peter D. Turney. Learning algorithms for keyphrase extraction. *Inf. Retr.*, 2:303–336, May 2000.

[206] Peter D Turney. Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416, 2006.

[207] Liwen Vaughan and Mike Thelwall. Search engine coverage bias: evidence and possible causes. *Information processing & management*, 40(4):693–707, 2004.

[208] Ellen M Voorhees et al. The trec-8 question answering track report. In *TREC*, volume 99, pages 77–82, 1999.

[209] Xiaojun Wan and Jianguo Xiao. Collabrank: towards a collaborative approach to single-document keyphrase extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 969–976. Association for Computational Linguistics, 2008.

[210] Xiaojun Wan and Jianguo Xiao. Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the 23rd national conference on Artificial intelligence - Volume 2*, AAAI'08, pages 855–860. AAAI Press, 2008.

[211] Xiaojun Wan and Jianguo Xiao. Exploiting neighborhood knowledge for single document summarization and keyphrase extraction. *ACM Transactions on Information Systems (TOIS)*, 28(2):8, 2010.

[212] Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In *Annual Meeting-Association for Computational Linguistics*, volume 45, page 552, 2007.

[213] Rui Wang, Wei Liu, and Chris McDonald. Using word embeddings to enhance keyword identification for scientific publications. In *Databases Theory and Applications*, pages 257–268. Springer, 2015.

[214] Ryen W White and Resa A Roth. Exploratory search: Beyond the query-response paradigm. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 1(1):1–98, 2009.

[215] Peter Willett. Recent trends in hierarchic document clustering: a critical review. *Information Processing & Management*, 24(5):577–597, 1988.

[216] Alexander D Wissner-Gross. Preparation of topical reading lists from the link structure of wikipedia. In *null*, pages 825–829. IEEE, 2006.

[217] I Witten and David Milne. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA*, pages 25–30, 2008.

[218] Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. Kea: practical automatic keyphrase extraction. In *Proceedings of the fourth ACM conference on Digital libraries*, DL '99, pages 254–255, New York, NY, USA, 1999. ACM.

[219] Yi-fang Brook Wu, Quanzhi Li, Razvan Stefan Bot, and Xin Chen. Domain-specific keyphrase extraction. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 283–284. ACM, 2005.

[220] Xinyu Xing, Wei Meng, Dan Doozan, Nick Feamster, Wenke Lee, and Alex C Snoeren. Exposing inconsistent web search results with bobble. In *Passive and Active Measurement*, pages 131–140. Springer, 2014.

[221] Eric Yeh, Daniel Ramage, Christopher D Manning, Eneko Agirre, and Aitor Soroa. Wikiwalk: random walks on wikipedia for semantic relatedness. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, pages 41–49. Association for Computational Linguistics, 2009.

[222] Wen-tau Yih, Joshua Goodman, and Vitor R Carvalho. Finding advertising keywords on web pages. In *Proceedings of the 15th international conference on World Wide Web*, pages 213–222. ACM, 2006.

[223] Peifeng Yin, Nilam Ram, Wang-Chien Lee, Conrad Tucker, Shashank Khandelwal, and Marcel Salathé. Two sides of a coin: Separating personal communication and public dissemination accounts in twitter. In *Advances in Knowledge Discovery and Data Mining*, pages 163–175. Springer, 2014.

[224] Elad Yom-Tov, Susan Dumais, and Qi Guo. Promoting civil discourse through search engine diversity. *Social Science Computer Review*, pages 145–154, 2013.

[225] Arjumand Younus, M Atif Qureshi, Fiza Fatima Asar, Muhammad Azam, Muhammad Saeed, and Nasir Touheed. What do the average twitterers say: A twitter model for public opinion analysis in the face of major political events.

In *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, pages 618–623. IEEE, 2011.

[226] Oren Zamir and Oren Etzioni. Grouper: a dynamic clustering interface to web search results. *Computer Networks*, 31(11):1361–1374, 1999.

[227] Torsten Zesch and Iryna Gurevych. Analysis of the Wikipedia Category Graph for NLP Applications. In *Proceedings of the TextGraphs-2 Workshop (NAACL-HLT)*, 2007.

[228] Hongyuan Zha. Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 113–120. ACM, 2002.

[229] Mi Zhang and Neil Hurley. Avoiding monotony: improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 123–130. ACM, 2008.

[230] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, pages 22–32, 2005.