SWEETTWEET : A SEMANTIC ANALYSIS FOR MICROBLOGGING ENVIRONMENTS

by
Emre Yurtsever
B.S., Computer Engineering, Dokuz Eylul University, 2002

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering
Boğaziçi University
2010

ACKNOWLEDGEMENTS

With my deepest gratitude; I would like to thank to my thesis supervisor Dr. Suzan Üsküdarlı for her guidance, endless support and great patience throughout this thesis. I am grateful to Assist. Prof. Haluk Bingöl and Assoc. Prof. Yağmur Denizhan for participating in my thesis jury and their helpful comments.

I am deeply grateful to my dear wife Özge İkiz Yurtsever for her love, endless support and patience that helped me a lot all through this thesis.

Also I would like to express my gratitude to my team leader Erdem Günay and my colleagues, especially to Burak Çelebi, at Turkcell Teknoloji for their understanding and support.

ABSTRACT

SWEETTWEET : A SEMANTIC ANALYSIS FOR MICROBLOGGING ENVIRONMENTS

User collaboration became the key factor in the development of today’s Internet applications with the emergence of Web 2.0. Users not only consume the services available on the Internet, but also interact with them and collaborate to provide content generation for the services. Microblogs are recently one of the most interesting applications in the Internet. They are rapid, simple and easy to use when compared to the traditional blogs. These properties of microblogs create user interest and increase the popularity of these services. Twitter is the most popular microblog and it has millions of users posting millions of messages every day. The data available on Twitter is massive and it is growing continuously. This massive data contains valuable information. The work done in this M.S. thesis is to provide a methodology to categorize, analyze this data, understand the user contributions made to microblogs and export valuable information. However, microblogs have some limitations, especially on the size of the content. Same situation also applies for the user posts in Twitter, which are also known as “tweets”. This makes the analysis of the data on Twitter more challenging, since the only information we have for performing an analysis are the words in user tweets. First step in our method is to retrieve user tweets and parse them into words. Next, we need to analyze and understand the content of the user posts. To achieve this goal, we utilized Semantic Web resources. DBpedia, which is a central node on Linked Data effort, is selected as Semantic Web resource in this thesis work. DBpedia provides the data on WikiPedia in RDF format and it has an interface that enables us to perform complex SPARQL queries on the data set available on it. The model we proposed in this thesis work takes the words which are used frequently on users’ posts as input, queries them on Semantic Web resources and finds out the matching categories defined on this resource for these words. At the end of the analysis process, we have a group of category names for the users, which enables us to understand their contributions made to microblogs.

ÖZET

SWEETTWEET : MICROBLOG ORTAMLARININ SEMANTİK ANALİZİ

Web 2.0 kavramı, güncel İnternet uygulamalarının geliştirilmesinde kullanıcıların katılımını önemli bir unsur haline getirdi. Artık kullanıcılar İnternette sunulan servisleri sadece kullanmakla kalmıyor, aynı zamanda bu servislerle etkileşime girerek servisin içeriğinin oluşturulmasına katkıda bulunuyorlar. Microblog'lar son zamanlarda İnternet üzerinde bulunan en ilgi çekici uygulama konumundalar. Alışılageldik blog'larla karşılaştırıldıklarında, hızlı, basit ve kullanımları kolay olan Microblog'lar, bu özellikleri ile kullanıcıların dikkatini çekiyor. Twitter en popüler microblog konumunda ve her gün milyonlarca ileti gönderen milyonlarca kullanıcıya sahip. Bu nedenle, Twitter üzerinde muazzam derecede büyük bir veri bulunuyor ve bu veri büyümeye devam ediyor. Bu yüksek lisans tezinde yaptığımız çalışma, gerçekten değerli bilgileri içeren bu verinin kategorilere ayrılması ve analiz edilmesi, kullanıcıların Microblog'a yaptıkları katkının anlaşılabilir olması ve değerli bilgilerin ortaya çıkartılabilmesi için bir yöntem sunmak şeklinde özetlenebilir. Ancak microblog'larda özellikle içerik boyutu konusunda bazı sınırlamalar bulunuyor. Analiz yapabilmek için elimizde bulunan tek veri, kullanıcının mesajlarında bulunan kelimeler olduğundan, bu özellik Twitter üzerinde bulunan verinin analiz edilmesini zorlaştırıyor. Modelimizde ilk adım, kullanıcıların mesajlarının alınıp kelimelere ayrılması, sonraki adımda ise bu mesajların analiz edilmesi ve anlaşılmaya çalışılması olarak sıralanabilir. Mesajların analiz edilmesi aşamasında semantik ağ kaynakları kullanılıyor. Bu tez çalışmasında, Linked Data girişiminin merkezi bir bileşeni olan DBpedia, semantik ağ kaynağı olarak seçildi. DBpedia, WikiPedia'da bulunan verileri RDF formatında sunar ve bu veri seti üzerinde karmaşık SPARQL sorguları yapabilmek için bir arayüz sağlar. Bu tez çalışmasında sunduğumuz model, kullanıcıların mesajlarında en sık kullanılan kelimeleri alır, semantik ağ kaynaklarında sorgular ve bu kaynaktan dönen kategorileri eşleştirir. Analiz işleminin sonunda, kullanıcıların microblog'a yaptıkları katkıları anlamamıza yarayan grup kategori ismi ortaya çıkmış olur.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

API	Application Programming Interface
FOAF	Friend Of A Friend
GUI	Grpahical User Interface
HTTP	Hypertext Transfer Protocol
JSP	Java Server Pages
N3	Notation 3
OWL	Web Ontology Language
RDF	Resource Description Framework
SIOC	Semantically Interlinked Online Communities
SPARQL	Sparql Protocol And Resource Query Language
SQL	Structured Query Language
URI	Unified Resource Identifier
W3C	World Wide Web Consortium
XML	eXtensible Markup Language

1. INTRODUCTION

Most of the successful applications have common properties, such as; being simple, easy to use and easy to access, having a user-friendly interface that provides user interaction, etc. Web application development trends tend to be more user-centric in order to attract more users to the applications developed using recent Web technologies. Users are no longer passive consumers, but also help to improve applications by providing content to the application itself. User collaboration is a key factor in current Web applications. The so called Web 2.0 [1] offered information sharing, platform for participation, user collaboration and interactivity, which results in development of the applications like wikis, blogs, microblogs, mashups, content sharing platforms, social network services etc.

One of the most interesting concepts that emerged with the paradigm shift in Web development is “Microblogging”. Microblogs are a kind of rapid, easy and mobile blogging systems, where users publish very brief text messages via their cell phones, Web interface or email. The simplicity, ease of use, accessibility and popularity of microblogs attract users and motivate them to use these services frequently, resulting in a rapid increase of content. This massive amount of data contains valuable but unclassified information. If we can understand user posts sent to microblogs, in other words; understand what users are talking about in microblogging environments, then we can categorize and extract valuable information from this dynamic, evolving and ever-growing information pool on the Web.

However, there are significant challenges in understanding of microblogs. First, the content of user posts is limited. Limited posts contain limited information to analyze. Few words or tokens are the only information we have to understand. Furthermore, the results found by using keyword-based search will not be sufficient to understand the contribution of users. Take the keyword “Java” as an example. When a user posts a message that contains the keyword “Java”, is he talking about Java [2], an object-oriented programming language or Java, an island in Indonesia, or Java, a kind of coffee. In order to understand the nature of what a user contributes, we need a way to deal with such ambiguity. Natural language processing in not much useful due to the compact syntax conventions used by microbloggers. To provide that, we need a semantic approach that helps us to understand and classify user contributions made to microblogging systems.

This thesis investigates how Semantic Web [3] technologies can be successfully applied in determining what kind of contributions a microblogger makes. Yielding a general description of a microblogger can result in more successful seacrhes, recommendations and other further processing.

To summarize, our goal in this work is to analyse, describe microbloggers in terms of their contributions. Semantic Web resources are utilized for this purpose.

2. BACKGROUND

This section builds background information about the technologies used in this work to provide better understanding of the proposed model. Section 2.1. presents brief information about evolution of web, including Web 2.0 and user collaboration, taxonomies, folksonomies and social networks. In section 2.2. we start with the definition of Semantic Web and provide information about Semantic Web topics we used in our thesis work, such as RDF [4], Linked Data [5]. And section 2.3. provides information about microblogging environments and one of the most popular microblogs, Twitter [6].

2.1. Evolution of Web

The Internet and the services provided through it form an indispensable part of human life. The most popular service available over the Internet is World Wide Web (WWW) which carries massive amout of interlinked information about the resources. The WWW, or the Web, provides access to this information from any location using a web browser when user is “online” – connected to the Internet. The web, as we know today, had some milestones throughout the time. The following image shows these milestones during the evolution of Web.

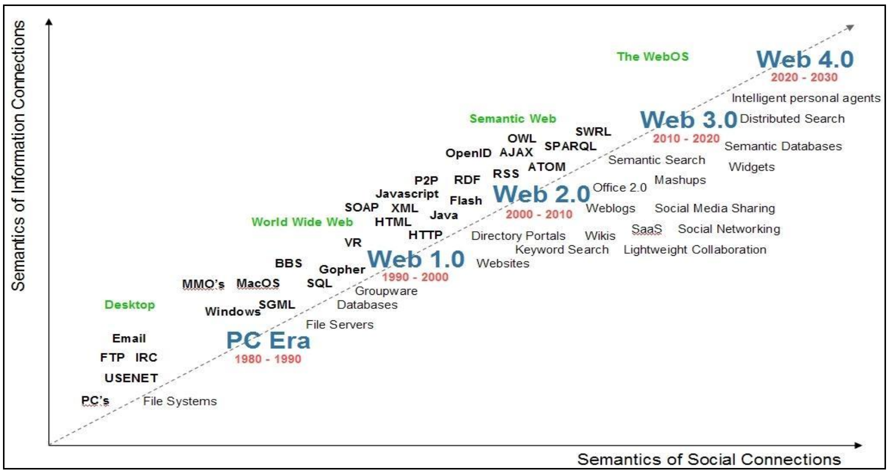


Figure 2.1. Evolution of Web [7]

2.1.1. Web 2.0

At first, there was Web 1.0 concept, in which the most of the resources available over the Web were static. As the technology evloved, the users needed a dynamic environment that adapts to the new paradigms in the world. The information offered over Web needed the collaboration of users, since the user of Web grew drastically over the time. Then we have met with the concept of Web 2.0, which provides a paradigm shift especially on the design of the Web. The concept of Web 2.0 is a vision of O'Reilly. At first it was not clearly understood. Tim Berners-Lee commented on Web 2.0 as follows.

"*Nobody really knows what it means... If Web 2.0 for you is blogs and wikis, then that is people to people. But that was what the Web was supposed to be all along*". [8]

Despite the negative comments on Web 2.0 concept, as the time passed, it turned out to be a new generation of web development and design.

Web 2.0 offers information sharing, architecture of participation, user collaboration and interactivity. This paradigm shift in Web design resulted in discovery and development of the applications like wikis, blogs, microblogs, mashups, content sharing platforms, social network services etc.

2.1.2. Social Networking

Social Networks are mostly web based online communities of users that share common interests in hobbies, religion, or politics. A social networking service essentially consists of user profile, social links, various applications, thus providing a focus on building and reflecting of social networks or relations among people. Most of the social networking applications enable their users to interact with the service and the other users of the service. Interactions between users provide user collaboration on the usage and the generation of the content of the service.

To provide better understanding of the user collaboration concept, we will briefly talk about taxonomies and folksonomies, which are also quite popular concepts in Web 2.0 domain. Taxonomy is the practice and science of classification. Since we deal with the resources available on the Web in this study, taxonomy we are talking about is the classification of the resources on the Web. For taxonomies, classification process is controlled and hierarchical. They are created by a small number of individuals, authors, originators. Folksonomies, on the other hand, are user generated taxonomies. The word “folksonomy” is constructed with the merge of “folk” and “taxonomy” words. Folksonomies became popular with the emergence of the concepts that rely on user collaboration, since user collaboration is an indispensible part of the many Web 2.0 applications, such as; collaborative tagging, social classification, social indexing, and social tagging.

A tag is a keyword associated with a piece of information (like picture, article, or video clip) on a resource on the Web, thus describing the item and enabling keyword-based classification of information it is applied to. Tagging is being done by everyone, no longer by only a small group of experts and allows users to collectively classify and find information based on the tags attached to resources on the Web.

The tags are being made public and shared, providing definition and categorization of the resources on the Web via user collaboration. Typically tags are displayed in a tag cloud on many sites. Below is the tag cloud for “Web 2.0” and its supportive concepts.

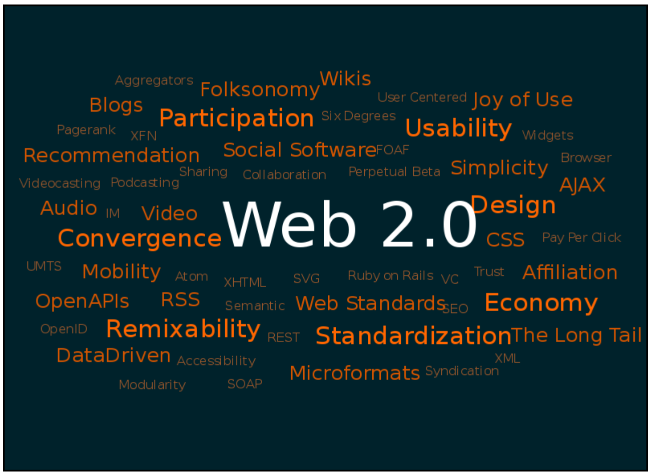


Figure 2.2. A tag cloud about Web 2.0 and supportive themes [9]

2.2. Semantic Web

Another significant concept that emerged during the evolution of the Web is “Semantic Web”. Semantic Web is an effort that tries to provide a Web with a meaning, making it possible for the machines to understand and process the data available on the Web. The idea is the following : if machines could use the web content, it would be easier to satisfy the requests of the people – e.g. finding and sharing information – since the machines can process the data faster than humans.

Tim Berners-Lee commented on his vision about Semantic Web in 1999 as follows:

“*I have a dream for the Web [in which computers] become capable of analyzing all the data on the Web – the content, links, and transactions between people and computers. A ‘Semantic Web’, which should make this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines. The ‘intelligent agents’ people have touted for ages will finally materialize.*” [10]

The Semantic Web effort consists of various working groups and specifications, including languages specifically designed for defining and querying data, such as eXtensible Markup Language (XML) [11], Resource Description Framework (RDF), Web Ontology Language (OWL) [12], Sparql Protocol And Resource Query Language (SPARQL) [13**]**. All these standarts are building blocks of Semantic Web architecture. They form Semantic Web stack, as shown in Figure 2.3.

In this thesis work, we will deal with a sub-topic of Semantic Web, which is called as “Linked Data” and one of the RDF based data sets defined on Linked Data Project, called as “DBpedia” [15]. To do our analysis process, we run queries on our semantic data set, DBpedia, written in SPARQL. In the following sections we will provide brief information about each topic of Semantic Web technologies we dealt with in this thesis work.

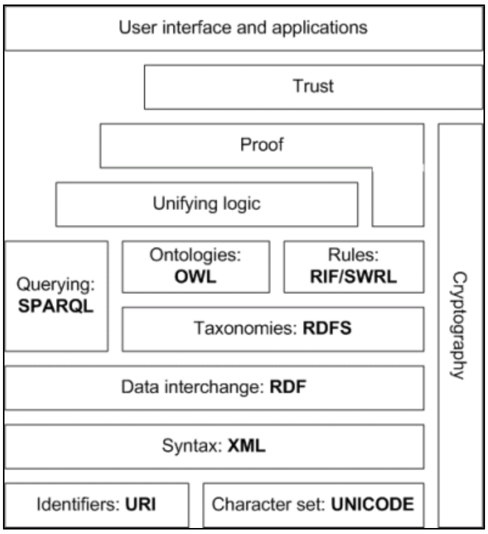


Figure 2.3. Semantic Web stack [14]

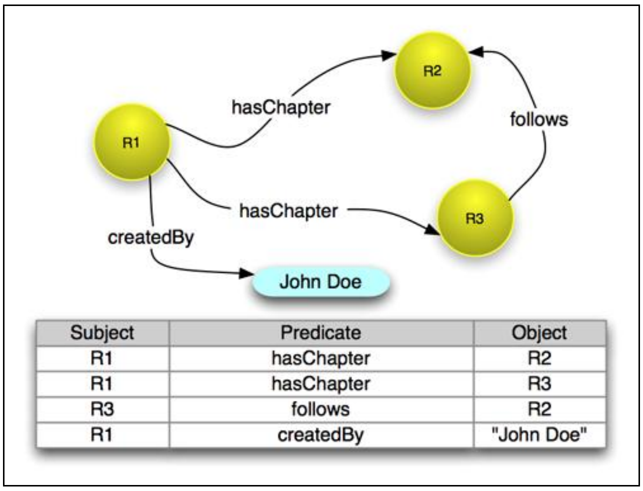
2.2.1. RDF

The Resource Description Framework (RDF), is a W3C recommendation, a standart model and a language for representing information about resources in the Web and for describing qualified relationships between them. It allows interoperability among applications exchanging machine-understandable information on the web.

RDF has a directed, labeled graph data format for representing information about resources in the Web. RDF statements are expressed in a triple format which consists of subject, object and predicate. In an RDF triple, a resource (the *subject*) is linked to another resource (the *object*) through an arc labeled with a third resource (the *predicate*). In other words :

- Subject : identifies the thing the statement is about
- Predicate : identifies the property of the subject that the statement specifies
- Object : identifies the value of that property

Figure 2.4. shows RDF triples for R1, R2 and R3 resources along with their subject-predicate-object relations :



Subject	Predicate	Object
R1	hasChapter	R2
R1	hasChapter	R3
R3	follows	R2
R1	createdBy	"John Doe"

Figure 2.4. RDF triples – subject, predicate and object [16]

R1, R2 and R3 are the resources in the RDF graph above. According to the figure, R1 is the subject of three triples, it has three outgoing edges. R2 is the object of two triples so that it has two incoming edges. Relations between resources are listed as predicates. For example, R1 created by the object "John Doe", it has two chapters called R2, R3. And finally R2 is followed by R3.

RDF has different serialization formats. Two most common serialization formats are RDF/XML [17] and Notation 3 (N3) [18]. RDF/XML format has an XML syntax for writing down and exchanging RDF graphs, while N3 format is more compact and human-readable. Figure 2.5. and figure 2.6. show the RDF statements for the same resource, the Wikipedia article about Tony Benn, with RDF/XML and N3 syntaxes.

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://en.wikipedia.org/wiki/Tony_Benn">
    <dc:title>Tony Benn</dc:title>
    <dc:publisher>Wikipedia</dc:publisher>
  </rdf:Description>
</rdf:RDF>
```

Figure 2.5. RDF/XML example

In RDF/XML example above, we have “`http://www.w3.org/1999/02/22-rdf-syntax-ns#`” and “`xmlns:dc="http://purl.org/dc/elements/1.1/`” namespaces defined. According to the namespaces in RDF statement, there is a “`title`” definition available in “`http://purl.org/dc/elements/1.1/`” URL. These namespaces define the meaning of the XML tags in RDF statements. If we replace “`<dc:title>`” and “`<dc:publisher>`” with “`<http://purl.org/dc/elements/1.1/title>`” and “`<http://purl.org/dc/elements/1.1/publisher>`” respectively, without providing the namespace on RDF statement, the document remains valid.

```
@prefix dc: <http://purl.org/dc/elements/1.1/>.

<http://en.wikipedia.org/wiki/Tony_Benn>
  dc:title "Tony Benn";
  dc:publisher "Wikipedia".
```

Figure 2.6. N3 example for same resource

The rules in RDF/XML namespace definitions apply for N3 example shown above. We have “`title`” and “`publisher`” elements defined in “`dc`” namespace.

The purpose of RDF is to enable information about the resources on the Web to be formally described, so that machines can process them. RDF can be utilized in many applications with intelligent software agents for resource description, discovery, and cataloging. The following paragraph provides information about the future of RDF.

RDF encourages the view of "metadata being data" by using XML as its encoding syntax. Once the web has been sufficiently "populated" with rich metadata, what can we expect? First, searching on the web will become easier as search engines have more information available, and thus searching can be more focused. Doors will also be opened for automated software agents to roam the web, looking for information for us or transacting business on our behalf. [19]

2.2.2. Linked Data

Linked Data is a community effort that aims to publish the data over Web as RDF data sets and to provide links between resources on the Web using RDF links. Wikipedia defines Linked Data as "*a term used to describe a recommended best practice for exposing, sharing, and connecting pieces of data, information, and knowledge on the Semantic Web using URIs and RDF*."

Linked Data contains data sets of many domains, such as reference (DBpedia, Freebase), music (Musicbrainz, BBC Music), science (GEO Species), bibliographic and much more. Figure 2.7. depicts the state of Linked Data as of March 2009.

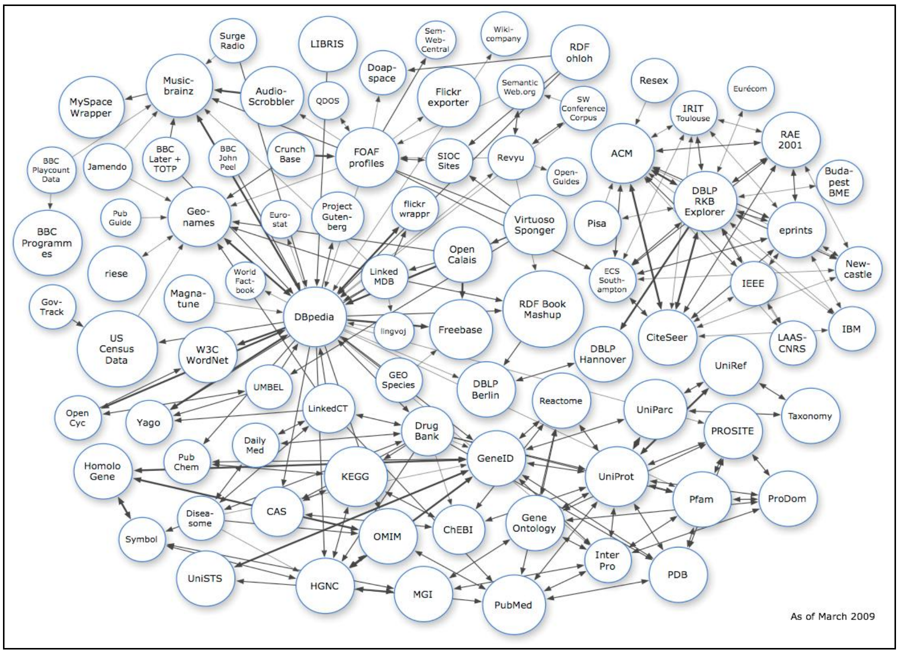


Figure 2.7. Data sets available on Linked Data [5]

Tim Berners-Lee described that the Semantic Web isn't just about putting data on the web, but is about making links, so that a person or machine can explore the web of data. The following rules are the design issues he outlined on putting data on the web for Linked Data:

1. Use URIs as names for things

2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)
4. Include links to other URIs. so that they can discover more things. [20]

2.2.3. DBpedia

DBpedia is one of the data sets defined on Linked Data project. It contains the data from WikiPedia extracted as RDF, including links to other data sets available on Linked Data. It also provides an endpoint to querying data available on it.

The DBpedia knowledge base currently describes more than 2.9 million things, including at least 282,000 persons, 339,000 places (including 241,000 populated places), 88,000 music albums, 44,000 films, 15,000 video games, 119,000 organizations (including 20,000 companies and 29,000 educational institutions), 130,000 species and 4400 diseases. The DBpedia knowledge base features labels and abstracts for these things in 91 different languages; 807,000 links to images and 3,840,000 links to external web pages; 4,878,100 external links into other RDF datasets, 415,000 Wikipedia categories, and 75,000 YAGO categories. The knowledge base consists of 479 million pieces of information (RDF triples) out of which 190 million were extracted from the English edition of Wikipedia and 289 million were extracted from other language editions. [15]

The knowledge base on DBpedia covers many domains and complex queries can be performed against these domains. We can query this knowledge base available on DBpedia using SPARQL, which is a W3C's standart SQL-like query language for RDF. The following section provides brief information about SPARQL.

2.2.4. SPARQL

SPARQL is a query language to query data stored as RDF. It is an official W3C Recommendation. It is pronounced as “sparkle” and stands for a recursive acronym as “Sparql Protocol And Rdf Query Language”.

SPARQL supports for queries with required and optional graph patterns as well as their conjunctions and disjunctions. SPARQL also supports extensible value testing and constraining queries against the source RDF graph. The results of SPARQL queries may be results sets or RDF graphs.

The syntax of SPARQL queries resembles SQL. For example, the following query returns name and nick values from FOAF (Friend Of A Friend) file of Tim Berners-Lee defined on "`http://www.w3.org/People/Berners-Lee/card`".

```
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
SELECT ?name ?nick
FROM <http://www.w3.org/People/Berners-Lee/card>
WHERE {
       ?Person foaf:name ?name;
               foaf:nick ?nick .
}
```

Figure 2.8. SPARQL query for retrieving FOAF file

Let's have a closer look on the query. Variables are prefixed with a "?" or "$". The "PREFIX" clause specifies namespace for FOAF "`http://xmlns.com/foaf/0.1/`" with the alias "foaf". This simplifies queries, so that we do not have to type the namespace in full each time it is referenced, since it serves as a reference to a full URI path. The "SELECT" clause specifies what the query should return. The "FROM" clause is optional, unlike SQL. It provides the URI of the data set to be queried. The "WHERE" clause provides the graph patterns in N3 syntax that the query attempts to match against the source graph.

2.3. Microblogging Environments

One of the most interesting applications that emerged with the paradigm shift in Web development triggered by the emergence of Web 2.0, is "Microblogging". Microblogs are a kind of rapid, easy and portable blogging systems in which users mostly publish their current status or their daily activities by posting brief text messages (typically 140 – 200 characters). Posts are often made via mobile phones and devices. The simplicity, ease, accessibility and popularity of microblogs attract users and motivate users to post frequently, resulting in a rapid increase of microblogs. The interest of users shows that microblogging is not a short-term trend, on the contrary, it is an important part of many people’s lives. The most notable examples of microblogs are Twitter, Tumblr [21], Jaiku [22] and identi.ca [23].

2.3.1. Twitter

Twitter is the most popular microblogging site and a social networking service was created in 2006. Twitter is very popular because of its simplicty and its property of providing information about what is happening right now around the globe, so that its name may be considered as synonymour with “microblogging”. It allows user to post text messages about anything; their current status or daily activities, feelings, links to pictures and videos, and breaking news. One of the founders of Twitter, Evan Williams, defined their vision on the service as :

“What we have to do is deliver to people the best and freshest most relevant information possible. We think of Twitter as it's not a social network, but it's an information network. It tells people what they care about as it is happening in the world.” [24]

The posts of users are called as “tweets”. Tweets have size limit of 140 characters. But the content of tweets may also contain links to external services, which may provide additional information about the tweet, when 140 characters are not enough. The links posted in tweets may exceed 140 characters. Users rely on space conserving techniques in order to make the best use at 140 characters. Short URL services shorten full link names. Abbreviations are commonly used on tweets. Words inessential to conprehension are often omitted. As a result, tweets are essentially a sequence of words, links and other special tokens, which don’t form well-formed sentences. A user often posts numerous tweets. The collection of tweets are more like streams of consciousness rather than coherent expressions. Twitter is a quick paced service and mostly preferred by users for its rapidity.

The service is widely used especially in United States. The following items are example tweets randomly selected from Twitter :

- Obama ends "national day of prayer." Nice! http://bit.ly/d9Ighe
- Java Web Application Developer - MCG - Midwest Consulting Group - Dallas, TX http://bit.ly/dAHpMB
- #Chicago Daily Deals - 50% Off Coffee at Boca Java http://bit.ly/aoHCuu
- RT @NiemanLab: Only 20% of TV newsrooms have Facebook pages, but 71% use Twitter "constantly" or "daily" (via @poynter) http://j.mp/dz660a

3. RELATED WORK

The emergence of Web 2.0 resulted in development of user-centric publishing applications, knowledge management platforms and social resource sharing tools that interact with users, such as wikis, blogs, microblogs, mashups, social network services. New Web design methodologies that come in to the picture with Web 2.0 concept, offer communities for information sharing, provide an architecture of participation, and support user collaboration in newly developed services.

Social Networks are kind of user communities where the users have knowledge in a specific domain and share same interests. Searching and finding users in a social network is an important issue to enable the consumption of the data available on these networks. Discovery of people on social networks can be performed based on keyword searching [25] or using some metrics specific to the domain, like reputation – popularity in the community – ranking [26][27][28]. Keyword based search on social networking sites is helpful in identifying and finding resources on the community, however it has some drawbacks like retrieving too many records for your search, or fetching the records that are irrelevant to your search. Keyword based search has also temporal issues. It retrieves most recent rather than most relevant results. For ranking based searches, it is problematic to find the users who contribute valuable information but have lower rankings. Since the metrics on the determination of rankings mostly depend on the number of followers, frequency of usage, it is likely to miss someone who has total relevance to your search, by using this methodology.

Collaborative tagging, taxonomies and folksonomies are also relevant topics of this thesis, since we deal with the categorization of user contributions. A tag is a keyword associated with a piece of information (like picture, article, or video clip) on a resource on the Web. Taxonomy is the classification of the resources provided by a small group of specialists, while folksonomy is a kind of taxonomy created by the users on the Web. There has been a huge expansion in taxonomy and folksonomy usage and there are numerous studies on collaborative tagging, taxonomies and folksonomies. A study by

Hotho, et.al., in this field propose model and a new search algorithm for folksonomies, called *FolkRank*, [29]. takes into account the folksonomy structure for ranking search requests in internet and intranet based folksonomy systems.

There are also approaches that attempt to combine social networks and Semantic Web efforts and to define semantic relations between the tags created by the communities. Gruber, argues in [30] that the efforts on Semantic Web and Social Web should be combined to create a new level of value that is both rich with human participation and powered by well-structured information available on semantic web. The main idea here is to create a “collective intelligence” out of the combination of user-contributed content and machine-gathered data. The idea of combining two promising fields has been influenced by many researchers, as in [31] and [32], to create and extend models that utilize semantic web and social web in conjunction. These studies explore and present approaches for collaborative tagging activities and folksonomies at a semantic level.

Social Networks and other social media sites form a rich source of data with the help of user participation on the generation of the content. There are some efforts to enable the data available on Web 2.0 communities using Semantic Web technologies, such as [33]. This work describes how Semantically Interconnected Online Communities (SIOC [34]) and the Semantic Web can enable linking of data from Web 2.0 community sites by providing a SIOC types module that specifies the type of content items and acts as a "glue" between user posts and the content items created by users.

A model to discover semantic relations between tags by adding the social dimensions is proposed in [35]. In this model, authors extend the traditional bipartite model of ontologies with the social dimension, providing a tripartite model for tags, users and resources. Mika also proposes in [36] a model that present three advances in exploiting the opportunity of semantically-enriched network data: (1) an ontology for the representation of social networks and relationships (2) a hybrid system for online data acquisition that combines traditional web mining techniques with the collection of Semantic Web data and a case study highlighting some of the possible analysis of this data using methods from Social Network Analysis.

Finally, a recent development in search engines domain is worth to be mentioned. Google search engine and Twitter have reached an agreement that enables Google to use Twitter updates in search results [37]. Thus, Google’s search results will be displayed with up-to-the-minute data available on Twitter. This is an important and interesting development to see how real time user contribution can make Google search better.

4. PROPOSED MODEL

The size limitation of microblogs result in conventions for efficient expression, such as the use of abbreviations and sufficiently informative fragments. An example post:

“*RT @mashable Twitter to Developers: Attach Any Data You Want to T.. http://bit.ly/bOybRo #annotations #chirp #development*”

When a user examines microbloggers, he or she will typically read several of their posts in order to get an idea of the nature of their contributions. Analyzing such contributions is not trivial since they are typically partial expressions and rely on many shorthand conventions.

To analyze microbloggers we rely on their posts, which consist of words, abbreviations, special tokens, and links to external resources. The words in posts provide a general idea about what kind of content they contribute. But this doesn’t provide the abstract (higher level) description of their contribution. For example, a microblogger may use the words “women” and, “empowerment” etc. But they are less likely to use words like “gender” or “sociology”. However, when one wants to find users who are interested in gender issues or sociology, they would want to find those who contribute on women issues. In another words, the ability to better characterize users enables finding them more effectively.

In this work, the main goal is to characterize microbloggers based on their posts. In order to obtain higher level descriptions of microbloggers, semantic web resources are utilized. Figure 4.1 presents the overall view of the model.

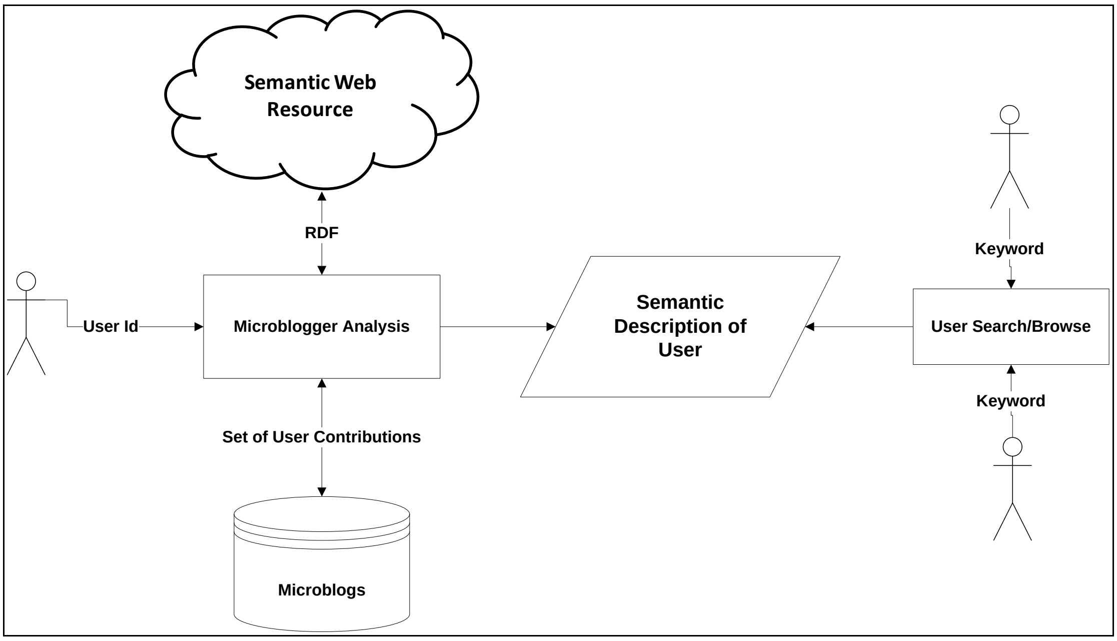


Figure 4.1. Overall view of the model

The semantic analysis of a microblogger is the central goal of this model. For a given user id, user contributions are retrieved from the microblog database. This set of user contributions is processed using semantic web resources to yield a high level description of the user, which consists of words and phrases. This high level description is referred to as the semantic description of a user.

Before investigating the model in details, some of the concepts used in this work, such as "*resource*" and "*category*", are needed to be explained to provide better understanding of the model. A "resource" is an entity available on the semantic web. Each resource has an identity, and this identity is expressed by a well-formed URI. For example; when the keyword "RDF "is queried against semantic web resource, a page defined for "RDF" is retrieved. A "category" also is an entity on the semantic web and is expressed by a well-formed URI. Each resource is created under a category defined in the semantic web resource utilized in this work. For example; categories for the resource "RDF" are defined as "Semantic Web", "World Wide Web Consortium standards" and "Knowledge representation languages". Since the content on the semantic web resource utilized in this work is generated by the users, resources and the categories are created by the users, as well.

Figure 4.2 shows the microblogger classification model proposed in this work. The circled numbers in the figure indicate the execution order of the steps defined in the model.

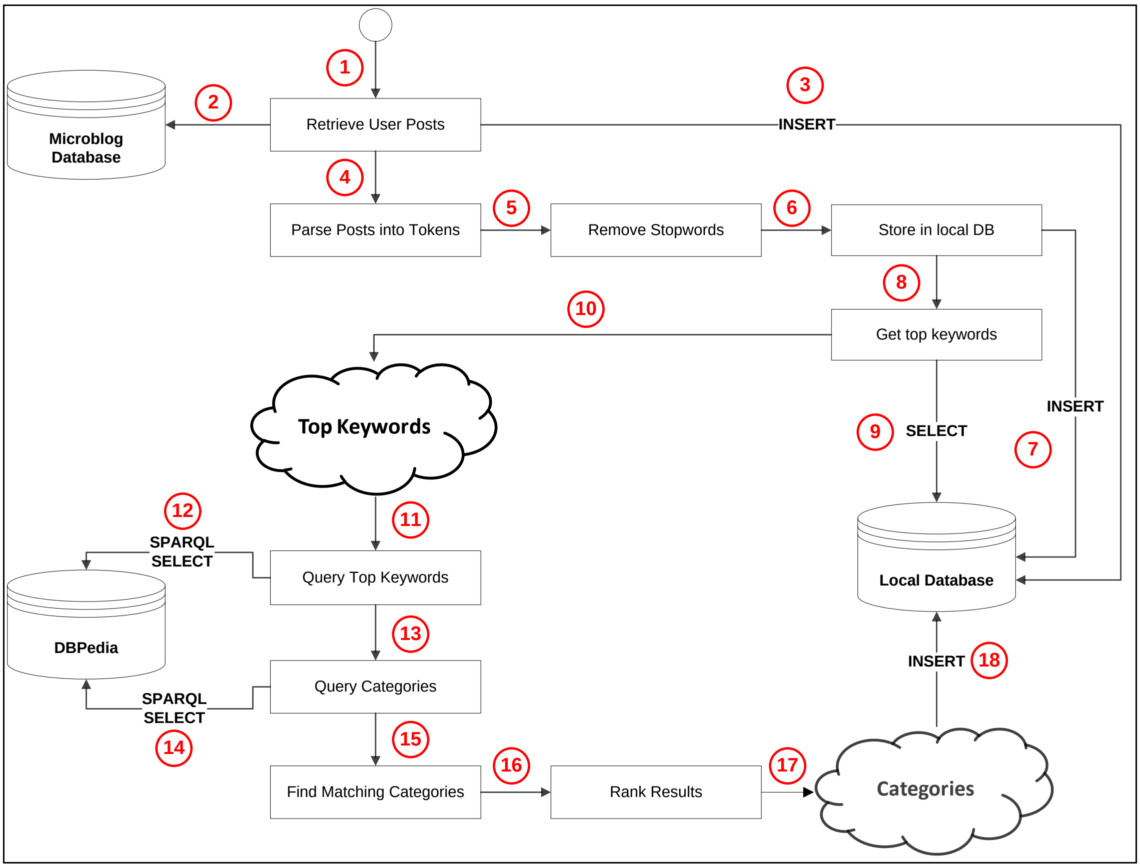


Figure 4.2. Microblogger classification model

First, the posts are retrieved and stored. Then these posts are parsed and filtered stopwords and punctuations are removed yielding a set of potentially useful words. These words are called “*keywords*”. All keywords obtained during processing are stored on database. The keywords that occur most frequently are called the “*top keywords of a user*”.

As stated earlier, keywords provide a limited understanding of a microblogger. In order to better characterize the nature of a microblogger’s contribution, a semantic analysis is needed. This is where semantic web come in to play in this work. In an analysis process, the keywords in the top keyword list of a microblogger are queried with two different approaches against semantic web resource. In the first approach, each keyword in the top keyword list is considered as a resource defined in semantic web. The keywords in the top keyword list are queried seperately and the categories defined for these resources are retrieved. However, this approach may lead us to ambiguity problems for the categories found. For example; querying the keyword “Java” returns categories “Java” and “Island of Indonesia”. It is not clear that whether the microblogger is contributing about an island, a programming language, or a kind of coffee. To eliminate this ambiguity, second approach is applied in the model. In this approach, all resources that contains the keywords in top keyword list in their definitions are retrieved. Then the categories of these resources are fetched. Finally, a category matching algorithm is applied to find the weighted common categories.

Figure 4.3 shows the details of the computation of weighted common categories based on the keyword set which provides semantic descriptions of microbloggers.

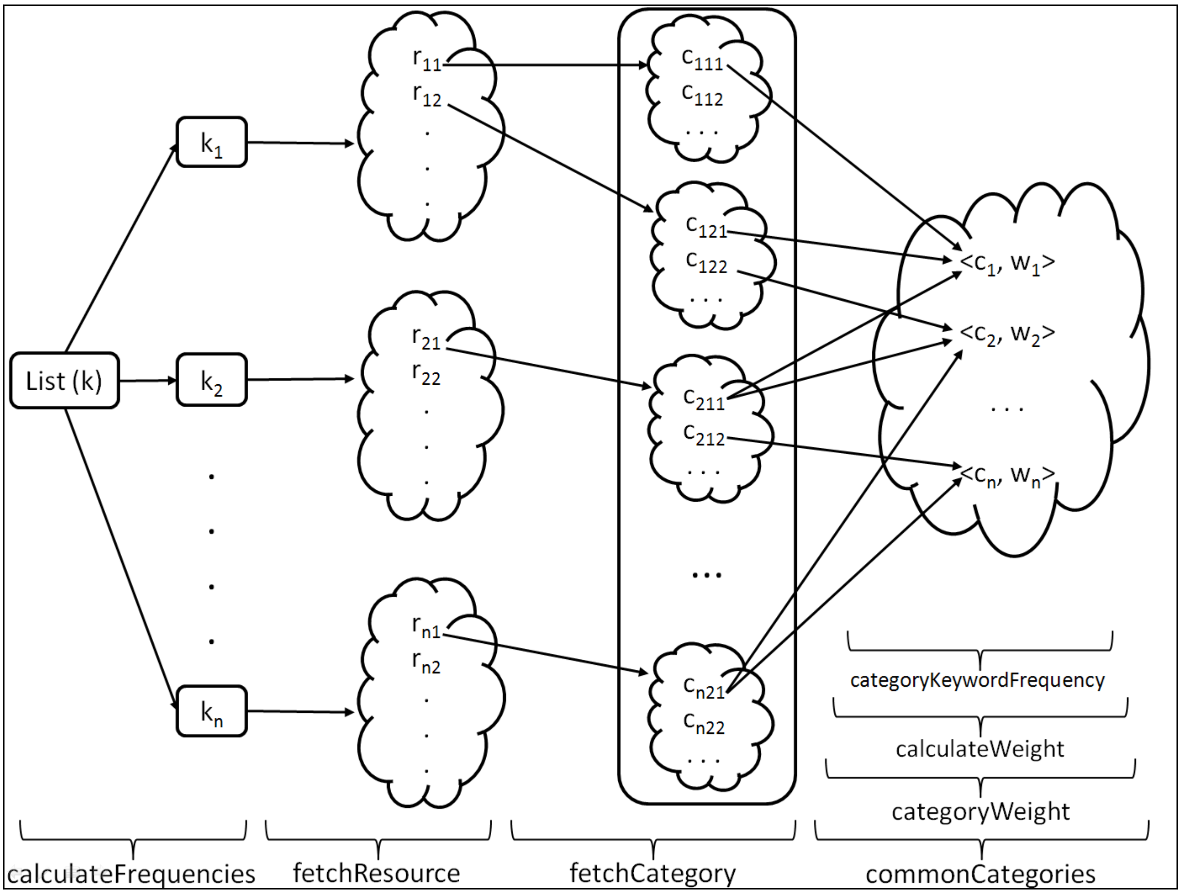


Figure 4.3. Computation of weighted common categories

The following list explains the methodology followed in the computation of weighted common categories :

- *U* is a set of microblog users.

$$U = \{u_1, u_2, u_3, \ldots, u_n\}$$

- A *contribution* is a text message that a user posts to a microblog.
- *K_i* is a sequence of space seperated, tokenized words that exist in the contributions of user *u_i*, where stopwords are excluded and punctuations are removed.

$$K = (k_1, k_2, k_3, \ldots, k_n)$$

- *R* is a set of semantic web resources.

$$R = \{r_1, r_2, r_3, \ldots, r_n\}$$

- *C* is a set of category names defined on the semantic web.

$$C = \{c_1, c_2, c_3, \ldots, c_n\}$$

- *f* is the frequency of each word in the contributions of a microblogger.
- *w* is the weight calculated for each category.
- Let *f_j* be the frequency of each distinct word kj in K_i.
- *calculateFrequencies(x : K)* is a function that takes a sequence of keywords and calculates their frequencies. It returns a sequence of keywords and their corresponding frequencies.

$$\text{calculateFrequencies}(K) := ((k_1, f_1), (k_2, f_2), \ldots, (k_n, f_n))$$

- *fetchResource(x : (k, f), y : R)* is a function that takes a sequence of keywords, their corresponding frequencies and the set of semantic web resources, R. It returns a sequence of semantic web resources, their corresponding keywords and frequencies of these keywords.

$$\text{fetchResource}((\, k, f), R) := ((r_1, k_1, f_1), (r_2, k_1, f_1), (r_3, k_2, f_2), \ldots, (r_n, k_m, f_m))$$

- *fetchCategory(x : (r, k, f), y : C)* is a function that receives a sequence of resources, their corresponding keywords and frequencies, and a set of categories defined on the semantic web. It returns a sequence of categories and their corresponding keywords and frequencies.

$$\text{fetchCategory}((r, k, f), C) := ((c_1, k_1, f_1), (c_1, k_2, f_2), (c_2, k_2, f_2), \ldots, (c_n, k_m, f_m))$$

- *commonCategories(x : (c, k, f))* is a function that takes a sequence of categories, their corresponding keywords and frequencies. It returns a set of categories which are common.

$$\text{commonCategories}((c, k, f)) := \{c_1, c_2, \ldots, c_n\}$$

- Let c_{common} be the set of common categories.
- *categoryWeight(x : c_{common}, y : (c, k, f))* is a function that takes a set of common categories and a sequence of all categories with their corresponding keywords and frequencies. It returns a sequence of common categories and their calculated weights. This function employs *calculateWeight* function to find the weights of all common categories.

$$\text{categoryWeight}(c_{common}, (c, k, f)) := ((c_1, w_1), (c_2, w_2), \ldots, (c_n, w_n))$$

- *calculateWeight(x : c_{common}, y : (c, k, f))* is a function that takes a set of common categories and a sequence of all categories with their corresponding keywords and frequencies. It returns calculated weights for categories.

$$\text{calculateWeight}(\text{c}_{\text{common}}, (\text{c}, \text{k}, \text{f})) := \frac{\text{categoryKeywordFrequency}(\text{c}_{\text{common}}, (\text{c}, \text{k}, \text{f}))}{\sum_{j=0}^{c.size} f_j}$$

- *categoryKeywordFrequency(x : c_i, y : (c, k, f))* is a function that takes a common category name and a sequence of all categories with their corresponding keywords and frequencies. It returns a frequency value, which is the sum of the frequencies of the keywords that correspond to the given common category name.

$$\text{categoryKeywordFrequency}(\text{c}_{\text{i}}, (\text{c}, \text{k}, \text{f})) := \sum_{i=0}^{c_i.size} f_i$$

Computation of weighted common categories begins with a given set of contributions of a microblogger. First step is to perform preprocessing on this set of contributions to find out potentially useful words for the analysis process. Preprocessing phase produces a space seperated keyword set, from which stopwords and punctuations are removed. Then the function *calculateFrequencies* is called. This function takes a keyword set and calculates the frequency, in other words, the number of occurence of each keyword in this set. The output of this function is a sequence of frequency weighted words, which are called as top keywords of a microblogger. Next step is to retrieve resources according to these keywords. For each keyword in the top keyword list of a microblogger, the function *fetchResource* retrieves the resources on semantic web resource that contain this keyword in their labels. This function returns a sequence of resources, their corresponding keywords and frequencies. Then *fetchCategory* function retrieves the categories defined for each resource. Since a resource can be defined in more than one categories, the categories common to more than one resources are needed to be filtered. In this phase, *commonCategories* function comes in to play. Given a set of category names, their corresponding keywords and frequencies, this function finds the categories, which are common to more than one resource. The output of this function is a set of common categories. The function *categoryWeight* takes this set of common categories as an input and returns a sequence of common categories along with their calculated weights. Weight value for each common category, is calculated by calling *calculateWeight* function. To find the weight for each common category, the keyword frequency calculated for one common category is divided to the sum of all keyword frequencies. The keyword frequency of a common category is found by calling *categoryKeywordFrequency* function. This function finds the matching categories for a given common category name, and sums the frequencies of all corresponding keywords to this category name. The result of this function gives us the frequency of the category.

Figure 4.4 shows the categories found based on the top keywords of a user and displays how the model selects the common ones and how it calculates their weights.

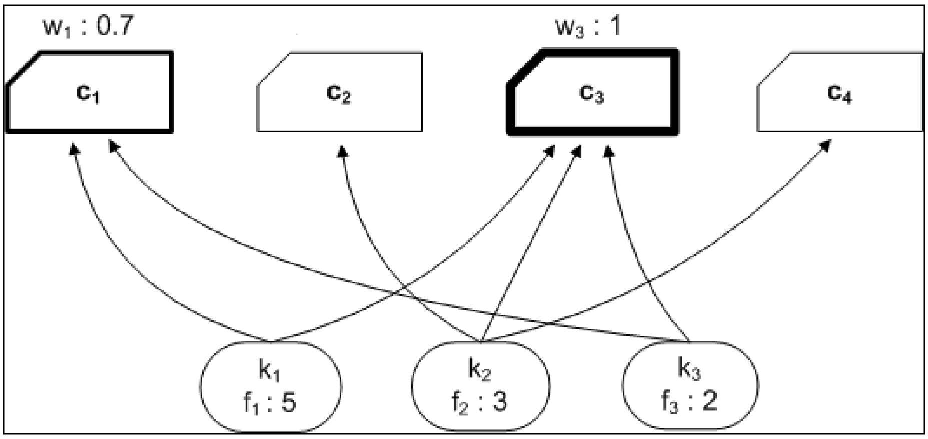


Figure 4.4. Finding common categories and calculating their weight

In Figure 4.4, c_1, c_2, c_3 and c_4 are the categories found by querying top keywords, while k_1, k_2, and k_3 are the top keywords, and f_1, f_2, and f_3 are their frequencies, respectively. The common categories found by using *commonCategories* function are c_1 and c_3. The categories *c2* and *c4* are not in common categories list, since they are only found for *k2*. The categories c_1 and c_3 are common for the following keyword – frequency pairs :

- $c_1 = ((k_1, f_1), (k_3, f_3))$
- $c_3 = ((k_1, f_1), (k_2, f_2), (k_3, f_3))$

The weight for each category is calculated by appliying the formula provided in the definition of *calculateWeight* function. According to this formula, the frequency value for each common category is needed to be calculated to find the weight value. Below are the frequencies of the common categories found by calling *categoryKeywordWeight* function :

- $f_{c1} = (f_1 + f_3) \rightarrow (5+2) = 7$
- $f_{c3} = (f_1 + f_2 + f_3) \rightarrow (5+3+2) = 10$

Total category frequency is calculated as below :

- $f_{c_{total}} = (f_1 + f_2 + f_3) \rightarrow (5+3+2) = 10$

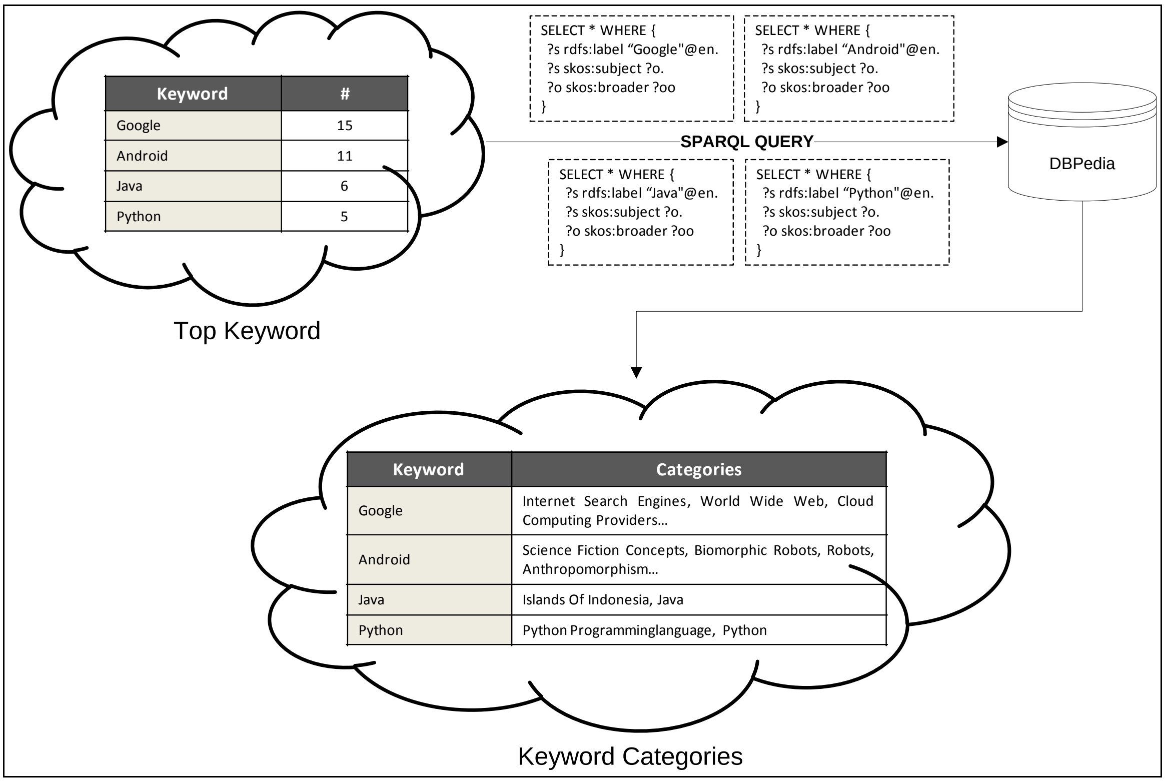
And finally, below are the weights calculated for each common category :

- $w_1 = \frac{f_{c1}}{f_{c_{total}}} \rightarrow \frac{7}{10} = 0.7$
- $w_3 = \frac{f_{c3}}{f_{c_{total}}} \rightarrow \frac{10}{10} = 1$

According to these results, our analysis may come up with the prediction that this user is significantly interested in categories “c_3” and “c_1”. Naturally, in a real case, the number of keywords is much greater.

To understand the model better, we proceed with an example. Consider a case where the top four keywords of a user are “Google”, “Android”, “Java” and “Python”. Figure 4.5 shows the results of querying keywords seperately.

For each keyword, a set of broader categories are queried. Indeed, the categories found for each keyword provide additional information about the contributions of a microblogger, as shown in Figure 4.5.



Keyword	#
Google	15
Android	11
Java	6
Python	5

```
SELECT * WHERE {
  ?s rdfs:label “Google"@en.
  ?s skos:subject ?o.
  ?o skos:broader ?oo
}
```

```
SELECT * WHERE {
  ?s rdfs:label “Android"@en.
  ?s skos:subject ?o.
  ?o skos:broader ?oo
}
```

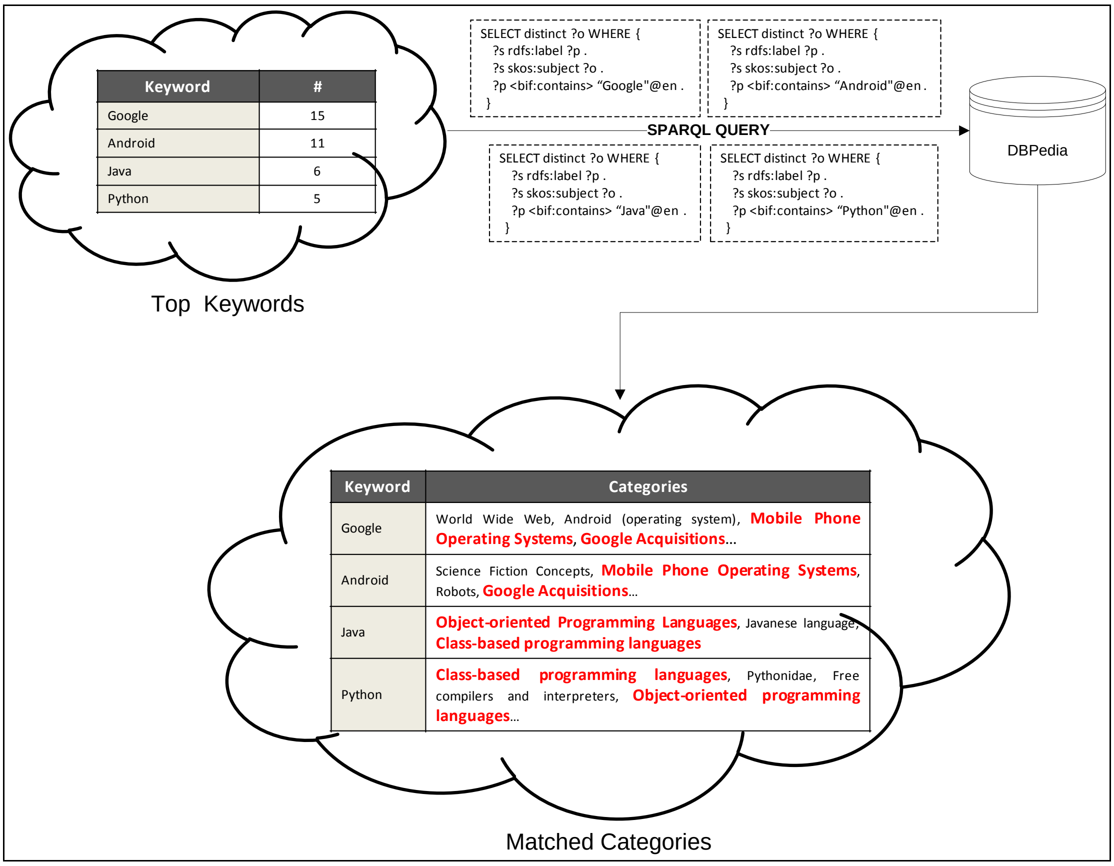
```
SELECT * WHERE {
  ?s rdfs:label “Java"@en.
  ?s skos:subject ?o.
  ?o skos:broader ?oo
}
```

```
SELECT * WHERE {
  ?s rdfs:label “Python"@en.
  ?s skos:subject ?o.
  ?o skos:broader ?oo
}
```

Keyword	Categories
Google	Internet Search Engines, World Wide Web, Cloud Computing Providers…
Android	Science Fiction Concepts, Biomorphic Robots, Robots, Anthropomorphism…
Java	Islands Of Indonesia, Java
Python	Python Programminglanguage, Python

Figure 4.5. Query results for obtaining broader categories

However, the resulting categories do not provide sufficient information to infer the categories related the keywords. Take the keyword “Android”, for example. According to the results of our queries, the categories associated with the keyword “Android” are “Biomorphic Robots”, and “Anthropomorphism”. However, since the user is interested in Google and its products, the user may be referring to the “Android Mobile Operating System” [38]. Which one is correct? “Mobile Operating System” or “Biomorphic Robots”? Same situation applies for the keywords “Java” and “Python”. When trying to interpret these results, one may think that the user is interested in programming languages or software development. However, this may not be the case. Perhaps the user is talking about the “Python”, which is also known as a kind of snake, living in the jungles of “Java”, which is also known as an “Island of Indonesia”. This ambiguity cannot be resolved when keywords are independently queried against semantic resources. Instead, common categories are needed to be founda matching algorithm on the categories retrieved from the keywords should give better results.



Keyword	#
Google	15
Android	11
Java	6
Python	5

```
SELECT distinct ?o WHERE {
   ?s rdfs:label ?p .
   ?s skos:subject ?o .
   ?p <bif:contains> "Google"@en .
}
```

```
SELECT distinct ?o WHERE {
   ?s rdfs:label ?p .
   ?s skos:subject ?o .
   ?p <bif:contains> "Android"@en .
}
```

```
SELECT distinct ?o WHERE {
   ?s rdfs:label ?p .
   ?s skos:subject ?o .
   ?p <bif:contains> "Java"@en .
}
```

```
SELECT distinct ?o WHERE {
   ?s rdfs:label ?p .
   ?s skos:subject ?o .
   ?p <bif:contains> "Python"@en .
}
```

Keyword	Categories
Google	World Wide Web, Android (operating system), **Mobile Phone Operating Systems**, **Google Acquisitions**...
Android	Science Fiction Concepts, **Mobile Phone Operating Systems**, Robots, **Google Acquisitions**...
Java	**Object-oriented Programming Languages**, Javanese language, **Class-based programming languages**
Python	**Class-based programming languages**, Pythonidae, Free compilers and interpreters, **Object-oriented programming languages**...

Figure 4.6. Query results for category matching

As shown on Figure 4.6, the matching algorithm finds the matching categories for the keywords "Google" and "Android" as "Mobile Phone Operating Systems" and "Google Acquisitions". Thus, it becomes clear that the user is interested in the Android mobile operating system developed by Google for mobile phones. The "Java" and "Python" keywords also have matching categories like "Class-based programming languages" and "Object-oriented Programming Languages". Since we did not find any matching categories defined for "Java" the island and "Python" the snake, we assume that the user is not refering to a snake living on an island of Indonesia, but, rather the programming languages.

Since a category can correspond to more than two keywords, the frequency of the matching categories is stored, which enables their ranking. Category ranking allows us to eliminate less occuring (perhaps erroneous) category matchings. Finally, we are left with a weighted set of significant categories. The category set is stored for further use in searching users according to their contributions made to microblog.

The model proposed on this work also provides a search functionality based on the results of the analysis processes performed on microbloggers' contributions. For a given search keyword, the categories found for microbloggers are investigated and matching ones are listed. For example; when a user wants to find out the microbloggers who are contributing about "web", he or she simply enters "web" keyword and searches the category sets inferred for microbloggers. As a result of this search operation, the microbloggers who contribute about "web" topic are listed, if there are any.

5. EVALUATION

In this chapter, we evaluate the results of our model by semantically tagging some Twitter users and comparing these results with the categories list we have by asking 30 people to manually categorize these users. We also compare our results with the tags available on some web sites, such as twitterholic.com and wefollow.com.

5.1. User Contributed Categorization

To evaluate the analysis results of our model, we selected 30 Twitter users, who show different characteristics in Twitter usage. Some of these 30 Twitter users are selected randomly, some of them are selected from the suggestion lists available on twitterholic.com or wefollow.com. Since we perform the analysis process on the contributions of the users, the number of contribution was an important criteria in the selection of these users. As a result, selected users are the users who have posted more than 100 tweets to Twitter.

We asked 30 individual to manually check the contributions of this set of 30 Twitter users and categorize them. While we gather the manual evaluation responses from 30 different people, we performed the analysis process for the same 30 Twitter users using SweetTweet application. Next step in the evaluation process is to compare the automatic categorization results (please refer to Appendix A) with the manual categorization results we receive from 30 individuals (please refer to Appendix B).

The category names generated by SweetTweet may not match to the category names prepared for manual evaluation. This situation may create a difficulty in evaluation process. Therefore we need a mapping between automatic and manual categories. Appendix C lists this mapping between two categorization scheme in Appendix A and Appendix B.

To understand the methodology we applied in evaluation process, we proceed with the comparison of 5 random Twitter users selected from total 30 Twitter users we analyzed. To see the all comparison results for 30 users, please refer to Appendix D.

The category comparison results are classified as follows :

- **Exact:** SweetTweet category name exactly matches to user contributed category name.
- **Subsume:** User contributed category name contains SweetTweet category name.
- **Related:** SweetTweet category and user contributed category are related according to the related categories table in Appendix C.
- **Unrelated:** SweetTweet category is not related to any user contributed category.

The first user to analyze is "algore". When we compare the SweetTweet categories and user contributed categories we get the following results, shown in Table 5.1.

Table 5.1. Comparison results for the user "algore"

algore	
Exact	None.
Subsume	Environment (*Environmental Economics, International Environmental Organizations*)
Related	Environment (*Energy Development, Energy in the United States, Ecology, Climate Change Organizations, Renewable Energy Economy, Energy Policy, Climate Change*)
Unrelated	*Economic Problems*

The second user we analyze is "mashable". The comparison result for this user is shown in Table 5.2.

.

Table 5.2. Comparison results for the user “mashable”

mashable	
Exact	None.
Subsume	Internet (*Internet Memes, Internet Protocols*), Social Media (*Social Info Processing, Online Social Networking*)
Related	Social Media (Web 2.0), Technology (ITunes)
Unrelated	*Video on Demand Services, Cross-platform Software, Contemporary Arts, Video games with expansion.*

The third user we analyze is “mandy_griffin”. Table 5.3 shows the comparison result for this user.

Table 5.3. Comparison results for the user “mandy_griffin”

mandy_griffin	
Exact	None.
Subsume	None
Related	Movies (*Film Soundtracks, Film based on Novels*), Education (*Learning*)
Unrelated	*Pop Ballads, American Novels, Amerian Indie Rock Groups, Video game Developers, American Poetry Collections, Semantics, English Phrases*

The comparison result for “ScottBourne” is shown in Table 5.4.

Table 5.4. Comparison results for the user “ScottBourne”

ScottBourne	
Exact	None.
Subsume	Photography/Art (*Digital Art, Photography Organizations, Digital Photography, Photography Genre, Photography Magazines, Photography Techniques*)

Related	None
Unrelated	*Independent Record Labels, Graphic File Format, Internet Forums, Monthly Magazines*

The comparison result for the fifth user “mashable” is shown in Table 5.5.

Table 5.5. Comparison results for the user “Steveology”

Steveology	
Exact	Social Media.
Subsume	Internet (*Internet Terminology*, *Internet Marketing*, *Internet Culture*), Social Media (*Social Information Processing*, *Social Groups*)
Related	Internet (*Search Engine Optimization*), Social Media (*Web 2.0*), Business (*Privately held companies of the United States*)
Unrelated	*Communication*

According to these comparisons, we come up with the following table that summarizes the category comparison of SweetTweet and user generated categories.

Table 5.6. Comparison of SweetTweet and user generated categories

Username	Exact	Subsume	Related	Unrelated
algore	0	0.2	0.7	0.1
mashable	0	0.4	0.2	0.4
mandy_griffin	0	0	0.3	0.7
ScottBourne	0	0.6	0	0.4
Steveology	0.1	0.5	0.3	0.1
AVERAGE (%)	2%	34%	30%	34%

If we look at the results we have for these five users, we can see that the categories found by SweetTweet application match to the categories found by manual user contribution at a ratio of 66%. On the other hand, there are also some categories that are not related with the categories defined by the manual contribution of users. The percentage of unrelated categories for these five users is 34%.

The evaluation results shows that for some users our model produces accurate values, while for some others it does not. The analysis process gives best results for information sharing users, like “ScottBourne” or “Steveology”. For users like “mandy_griffin”, who uses Twitter for chit-chatting, the results are not promising.

5.2. Categories Defined by Other Parties

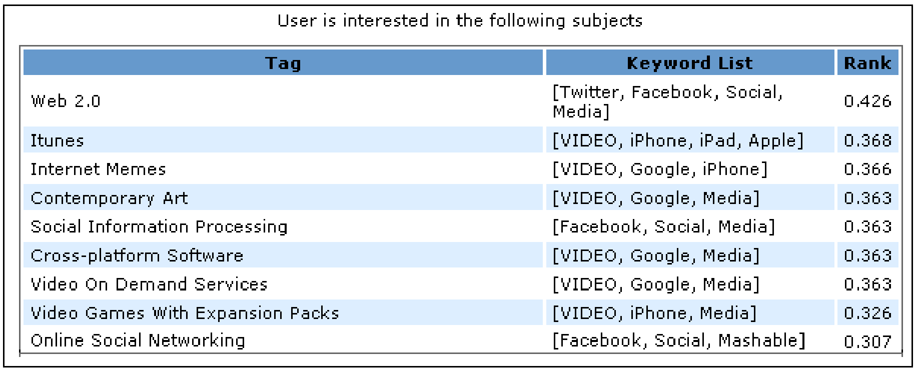
There are some web sites that classify popular Twitter users, such as wefollow.com and twitterholic.com. They list the most popular users within categories like music, social media, news, technology, comedy, etc. We also compare the categories our model produces by performing the analysis process, with the categories defined by these web sites. To examine the results of our system, we selected one popular user in a specific category, one average user and one celebrity.

5.2.1. A Popular Microblogger

A user classified under “Social Media” in both wefollow and twitterholic was selected for analysis. We refer to this user as u_1.The top keywords for u_1 are:

Top Keywords (u1) = {“VIDEO”, “Google”, “Twitter”, “Social”, “Facebook”,
“iPhone”, “Media”, “Mashable”, “iPad”, “Apple”}

Inspecting the top keyword provides a good idea about the content of u_1’s contributions. The semantic tags for u_1 are shown in Figure 5.1. The semantic classification for u_1 is consistent with the social media classification of wefollow and twitterholic lists. One who is interested in “Social Media” is expected to benefit from following u_1’s microblog. In another words, “u_1” could be recommended to users interested in social media.



User is interested in the following subjects

Tag	Keyword List	Rank
Web 2.0	[Twitter, Facebook, Social, Media]	0.426
Itunes	[VIDEO, iPhone, iPad, Apple]	0.368
Internet Memes	[VIDEO, Google, iPhone]	0.366
Contemporary Art	[VIDEO, Google, Media]	0.363
Social Information Processing	[Facebook, Social, Media]	0.363
Cross-platform Software	[VIDEO, Google, Media]	0.363
Video On Demand Services	[VIDEO, Google, Media]	0.363
Video Games With Expansion Packs	[VIDEO, iPhone, Media]	0.326
Online Social Networking	[Facebook, Social, Mashable]	0.307

Figure 5.1. Semantic categories for user “u_1”

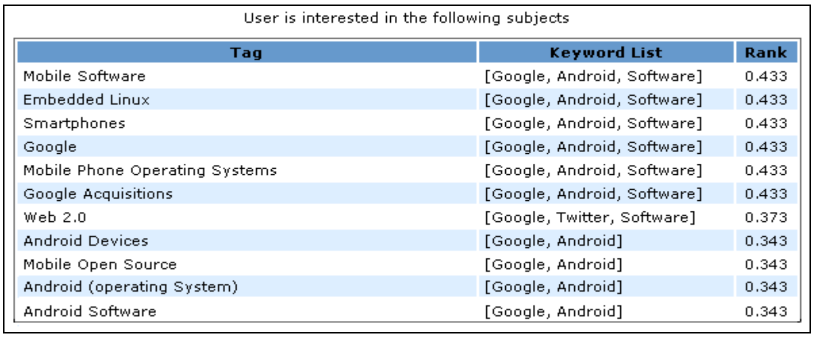
5.2.2. An Unclassified Average Microblogger

The second user, “u_2”, is a “regular” who is not classified. This user was selected based on personal interest of the author of this thesis. The top keywords for u_2 are:

Top Keywords (u_2) = {“Google”, “Android”, “pic”, “Twitter”, “Fwd”,
“Software”, “20”, “bir”, “da”, “ile”}

During the semantic tagging process, candidate categories for each keyword are proposed. For the keyword “Android” the categories “Anthropomorphism”, “Biomorphic Robots”, “Robots”, “Wikiproject Science Fiction Categories”, “Science Fiction Concepts” are proposed. Thus, one may assume that u_2 could be classifed as a contributor related to “Science Fiction”.

However, as the semantic categorization process continues other matching categories are found (see Figure 5.2). Thus, based on the analysis of all the top keywords, it becomes clear that the user is referring to the Mobile Phone Operating System called Android and not to androids in science fiction.



User is interested in the following subjects

Tag	Keyword List	Rank
Mobile Software	[Google, Android, Software]	0.433
Embedded Linux	[Google, Android, Software]	0.433
Smartphones	[Google, Android, Software]	0.433
Google	[Google, Android, Software]	0.433
Mobile Phone Operating Systems	[Google, Android, Software]	0.433
Google Acquisitions	[Google, Android, Software]	0.433
Web 2.0	[Google, Twitter, Software]	0.373
Android Devices	[Google, Android]	0.343
Mobile Open Source	[Google, Android]	0.343
Android (operating System)	[Google, Android]	0.343
Android Software	[Google, Android]	0.343

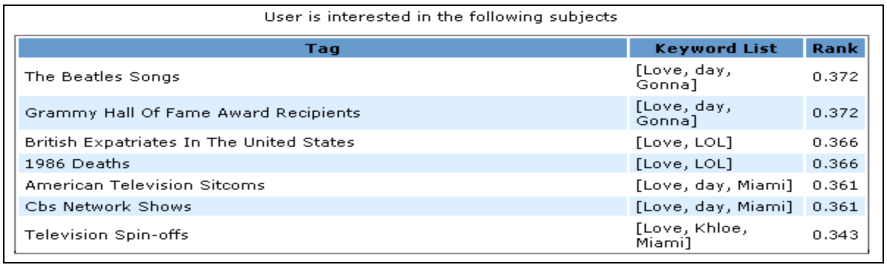
Figure 5.2. Categories for user “u_2”

5.2.3. A Celebrity Microblogger

The third user “u_3” has millions of followers in Twitter. The top keywords list for “u_3” are:

Top Keywords (u_3) = {“Love”, “2”, “LOL”, “Guys”, “day”, “tonight”,
“4”, “time”, “Gonna”, “Khloe”}

These keywords don’t give much insight regarding any domain of interest. The results of the semantic analysis are shown in Figure 5.3. For this user, the semantic analysis based on the same keywords corresponds to various semantic categories as “The Beatles Songs” or “1986 Deaths”, which also don’t give much insight regarding any domain of interest. That is because the contributions of u_3 mostly consist of chit-chat words.



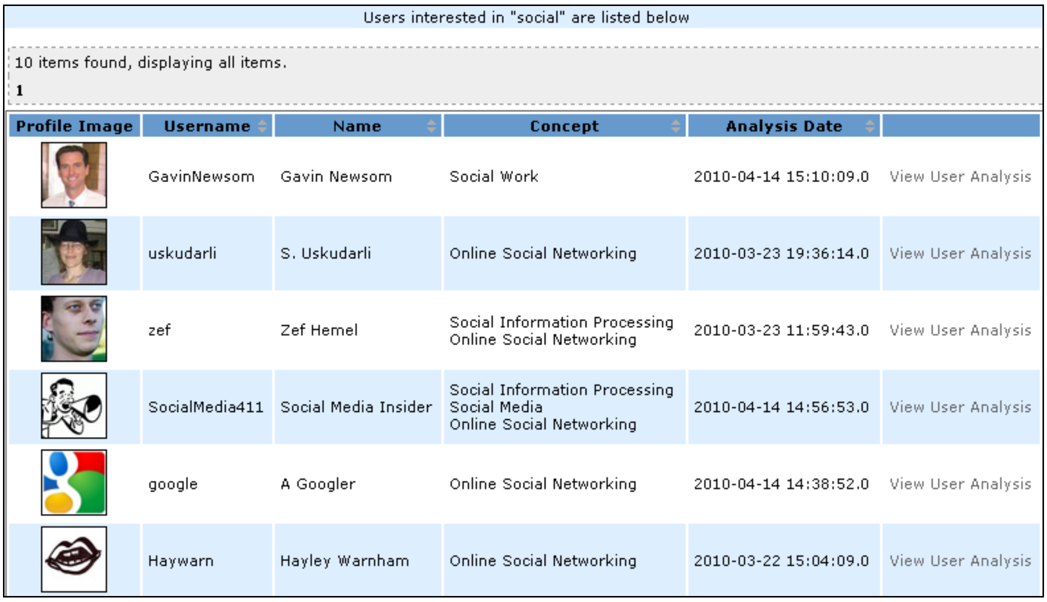
User is interested in the following subjects

Tag	Keyword List	Rank
The Beatles Songs	[Love, day, Gonna]	0.372
Grammy Hall Of Fame Award Recipients	[Love, day, Gonna]	0.372
British Expatriates In The United States	[Love, LOL]	0.366
1986 Deaths	[Love, LOL]	0.366
American Television Sitcoms	[Love, day, Miami]	0.361
Cbs Network Shows	[Love, day, Miami]	0.361
Television Spin-offs	[Love, Khloe, Miami]	0.343

Figure 5.3. Categories for user “u_3”

5.3. Searching Users

The proposed model on this thesis work also provides a user search mechanism based on the categories we inferred by performing analysis process on users. By using this search option, users can find which users are interested in the concepts they are searching for. This property of our model provide a user suggestion mechanism based on the contributions of the users. Below are the user list who are related with the category containing the word “social”. Users can select whom to follow in Twitter by searching concepts they are interested in.



Users interested in "social" are listed below

10 items found, displaying all items.
1

Profile Image	Username	Name	Concept	Analysis Date	
	GavinNewsom	Gavin Newsom	Social Work	2010-04-14 15:10:09.0	View User Analysis
	uskudarli	S. Uskudarli	Online Social Networking	2010-03-23 19:36:14.0	View User Analysis
	zef	Zef Hemel	Social Information Processing Online Social Networking	2010-03-23 11:59:43.0	View User Analysis
	SocialMedia411	Social Media Insider	Social Information Processing Social Media Online Social Networking	2010-04-14 14:56:53.0	View User Analysis
	google	A Googler	Online Social Networking	2010-04-14 14:38:52.0	View User Analysis
	Haywarn	Hayley Warnham	Online Social Networking	2010-03-22 15:04:09.0	View User Analysis

Figure 5.4. Users interested in the categories that contain the word “social”

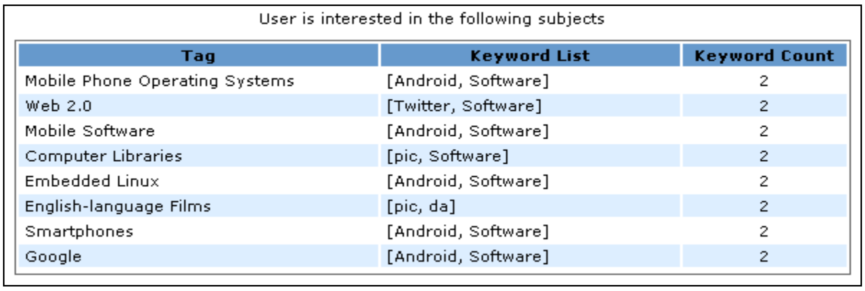
5.4. Discussion

According to the evaluation of the output the analysis processes, it can be seen that the proposed model gives best results for information sharing users. This is not surprising since, category information can be found in semantic references such as DBpedia (encyclopedic). Their contributions are mostly structured and predictable. And their followers are the users who seek information.

On the other hand, for chit chat type of contributions this model does not give good results, as expected. This is because such users use many socialization words (which end up in top keywords) that are not specific to any domain and thus are not useful in distinguishing an area of interest. It is possible that eliminating socialization related words may improve the analysis results. However it is possible to detect the nature of contributions as chit-chat type, which may be of interest to potential followers.

Most Twitter users are not popular nor classified by any external services. Whether they are sharing information or using chit chat words on their contributions, the potential followers of these users are likely to miss them, since they are not visible in any of the services that provides suggestion on who to follow on Twitter. SweetTweet is most beneficial for identifying the “regular” microbloggers in terms of their nature and content of contributions.

At the early stages of the development of this thesis work, semantic categories are ordered according to the total count of keywords that corresponds to these categories. For example; Figure 5.5 shows the former analysis results we have for the user “u_2”.

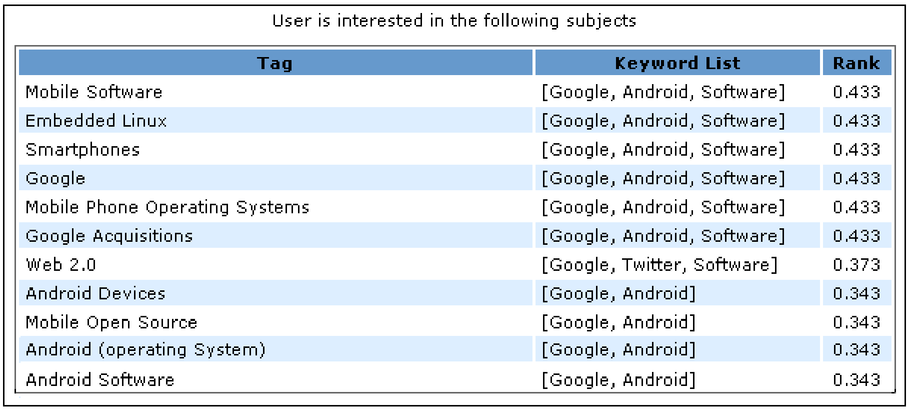


User is interested in the following subjects

Tag	Keyword List	Keyword Count
Mobile Phone Operating Systems	[Android, Software]	2
Web 2.0	[Twitter, Software]	2
Mobile Software	[Android, Software]	2
Computer Libraries	[pic, Software]	2
Embedded Linux	[Android, Software]	2
English-language Films	[pic, da]	2
Smartphones	[Android, Software]	2
Google	[Android, Software]	2

Figure 5.5. Keyword count ranked categories of user “u_2”

In this example, the category “Web 2.0” is retrieved from semantic web resource by querying 2 keywords, “Twitter” and “Software”. Thus, keyword count for this category is “2”. But, how would the results change if we define weights for the categories according to the frequencies of corresponding keywords and rank them? The results are shown in Figure 5.6 :



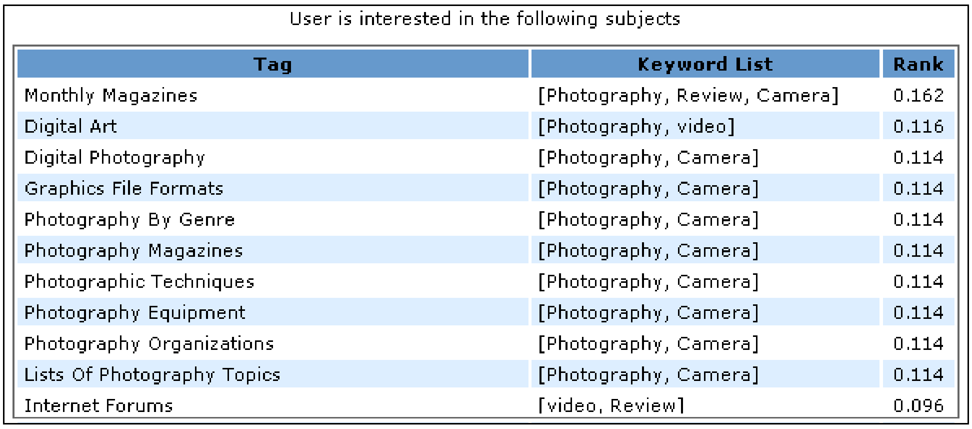
User is interested in the following subjects

Tag	Keyword List	Rank
Mobile Software	[Google, Android, Software]	0.433
Embedded Linux	[Google, Android, Software]	0.433
Smartphones	[Google, Android, Software]	0.433
Google	[Google, Android, Software]	0.433
Mobile Phone Operating Systems	[Google, Android, Software]	0.433
Google Acquisitions	[Google, Android, Software]	0.433
Web 2.0	[Google, Twitter, Software]	0.373
Android Devices	[Google, Android]	0.343
Mobile Open Source	[Google, Android]	0.343
Android (operating System)	[Google, Android]	0.343
Android Software	[Google, Android]	0.343

Figure 5.6. Keyword frequency ranked categories of user “u_2”

Each keyword in the top keyword list of a user has a frequency value. This value defines the number of occurence of the keyword in the contributions of the user. For example; the frequencies of keywords “Google” and “Android” are “20” and “15”, respectively. As explained in the model chapter, the weight for a category is calculated by dividing the sum of the frequencies of the corresponding keywords to the sum of the frequencies of the top keywords of a user. Let’s say that the sum of the frequencies of the top keywords of a user is “100”. We calculate the weight for the categories found by using “Google” and “Android” keywords, such as “Android Software”, by dividing “35” to “100”. When we apply the same methodology to all common categories, we come up with a weighted common category list. The categories found by using the keywords, which have greater frequencies, have greater weight values. Since we are trying to understand the contributions of the microbloggers, the categories that have greater weight values are more valuable for the evaluation of analysis results. Thus, assigning weight values to the categories helps our model to produce more accurate results than the previous approach.

The output of an analysis process for a Twitter user is a list of categories, which are common for more than one keyword in the top keyword list of this user, and their weight values. However, in some cases, it would be better if we could simplify the resulting category list. Because this list may contain more than one similar categories. Figure 5.5 displays such a case we have for user “u_4”.

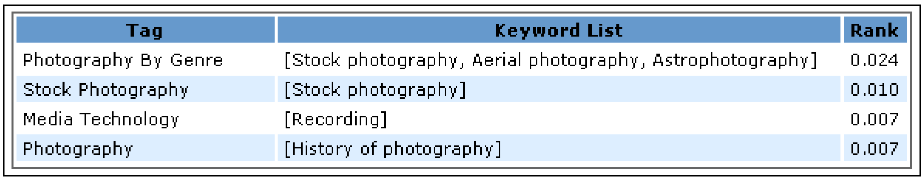


User is interested in the following subjects

Tag	Keyword List	Rank
Monthly Magazines	[Photography, Review, Camera]	0.162
Digital Art	[Photography, video]	0.116
Digital Photography	[Photography, Camera]	0.114
Graphics File Formats	[Photography, Camera]	0.114
Photography By Genre	[Photography, Camera]	0.114
Photography Magazines	[Photography, Camera]	0.114
Photographic Techniques	[Photography, Camera]	0.114
Photography Equipment	[Photography, Camera]	0.114
Photography Organizations	[Photography, Camera]	0.114
Lists Of Photography Topics	[Photography, Camera]	0.114
Internet Forums	[video, Review]	0.096

Figure 5.7. The category list for user “u_4”

According to the analysis results, the user “u_4” is apparently interested in “Photography”. However, the result of our analysis produces more than one “Photography” related categories. Is it possible to simplify this list? Since the categories on the semantic web resource are also defined under other categories, it would be a good idea to perform one more level of processing to find the upper categories for the categories list we have. When the same methodology applied that we applied to the top keyword list, we come up with the following common category list, as seen on Figure 5.6 :

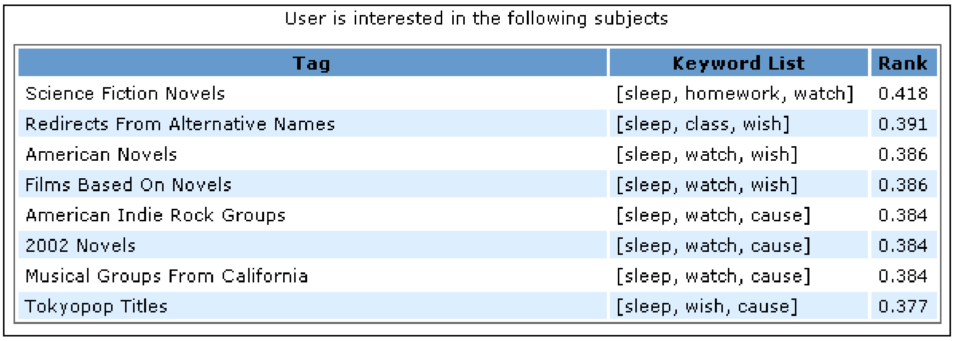


Tag	Keyword List	Rank
Photography By Genre	[Stock photography, Aerial photography, Astrophotography]	0.024
Stock Photography	[Stock photography]	0.010
Media Technology	[Recording]	0.007
Photography	[History of photography]	0.007

Figure 5.8. Upper category list for user “u_4”

The user “u_4” is a kind of a microblogger that shares information on Twitter. The contributions of this kind of users are more structured and predictable and the analysis results we have seem to be more accurate.

Consider a user that uses Twitter mostly for chit-chat. Does finding upper common categories provide useful information about the contributions of this kind of user? Let’s have a look at the category list of the user “u_5”.

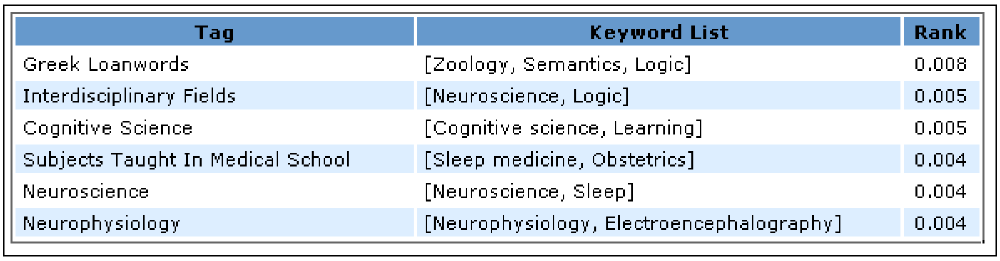


User is interested in the following subjects

Tag	Keyword List	Rank
Science Fiction Novels	[sleep, homework, watch]	0.418
Redirects From Alternative Names	[sleep, class, wish]	0.391
American Novels	[sleep, watch, wish]	0.386
Films Based On Novels	[sleep, watch, wish]	0.386
American Indie Rock Groups	[sleep, watch, cause]	0.384
2002 Novels	[sleep, watch, cause]	0.384
Musical Groups From California	[sleep, watch, cause]	0.384
Tokyopop Titles	[sleep, wish, cause]	0.377

Figure 5.9. The category list for user “u_5”

For chit-chat type of users, the semantic categories don’t give much insight regarding any domain of interest. Finding upper common categories for this kind of users would also do not provide accurate information, as seen on Figure 5.8 below :



Tag	Keyword List	Rank
Greek Loanwords	[Zoology, Semantics, Logic]	0.008
Interdisciplinary Fields	[Neuroscience, Logic]	0.005
Cognitive Science	[Cognitive science, Learning]	0.005
Subjects Taught In Medical School	[Sleep medicine, Obstetrics]	0.004
Neuroscience	[Neuroscience, Sleep]	0.004
Neurophysiology	[Neurophysiology, Electroencephalography]	0.004

Figure 5.10. Upper category list for user “u_4”

How about taking the contributions of a microblogger for a certain time interval and analyzing them? Does the analysis result change for this microblogger? Yes, it does. Microblogs are fast paced blogging environments. The content of user contributions change rapidly. Within a certain time interval, users may post tweets about the latest developments in a specific domain or flash news. Some new topics may emerge and become popular, or existing ones gain interest from Twitter users in that time period. These topics are called as “*trending topics*”. After some time, trending topics begin to lose their popularities, as expected. However, new trending topics continuously emerge, or a user begin to share information on Twitter about the latest developments in another domain that he or she is not interested at all before. The model proposed in this thesis performs analysis process using the top keywords of the users. Within different time intervals, top keyword lists of the users show difference. Thus, taking partial contributions of a user within certain time intervals changes the results of the analysis process.

The evaluation results of the proposed model show that the user on Twitter can broadly be categorized as (1) the users who share information, (2) those who seek information, and (3) those who “chit-chat” about daily events and actitivities. During the experiments we performed on Twitter based on the proposed model, many twitter users have been inspected, which exhibit similar results to these three cases we researched.

6. IMPLEMENTATION

In this section we provide implementation details of SweetTweet application. First of all, in section 4.1, we start with the details of Twitter4J API [39]. This section covers the methods we use in SweetTweet application to access and process tweets of users in Twitter. Section 4.2 explains how our application queries DBpedia using SPARQL. In this section, we give the details of the queries used in the application and show some example results of these queries to provide clear understanding of the methods we used in our analysis. And finally, section 4.3 provides information about the implementation of the application itself. The application consists of two parts : first one is the persistance layer by which we store and use the data we receive, and second one is the interface layer that provides user access to the application.

6.1. Twitter4J API

Twitter exposes its data via an Application Programming Interface (API). Users can access Twitter data using this API. There is a bunch of applications that use Twitter data to understand the nature of microblogs, to accomplish some analysis tasks using tweets of users etc.

In SweetTweet application we also use Twitter data to provide the analysis we aimed. To be able to that, we utilize a wrapper API, called “Twitter4J”, that facilitates the usage of the functionalities of Twitter API for Java. With Twitter4J, Java applications can easily integrate with the Twitter service. Twitter4J provides a simply uses username/password pair to create a new Twitter object by using its constructor as in Figure 6.1.

```
Twitter twitter = new Twitter(username, password);
```

Figure 6.1. Twitter constructor in Twitter4J

SweetTweet application needs the information of users in Twitter. To get this information from Twitter, we use “showUser” method of Twitter4J API. It returns a User object that contains the fields such as; user id, name, screen name, status count (also known as tweet count), followers count, friends count, location, last status post date etc. An example call sent to “showUser” method is shown in Figure 6.2.

```
User user = twitter.showUser(twitterUsername);
```

Figure 6.2. Get user info from Twitter

To retrieve the tweets of the user, we use following code fragments. However, Twitter limited the maximum fetch size for one user’s tweets to 3200. In order to get all the tweets that can be retrieved within these limits, we have to use a paging mechanism with as shown in Figure 6.3.

```
Paging page = new Paging(pageCount, 100);
```

Figure 6.3. Pagination for retrieving user’s tweets

As shown in the code fragment listed above, we get each page with the limit 100. To get all the available tweets, it is needed to define a loop. Being “pageCount” variable the index of this loop, we can get user’s tweets page by page using “getUserTimeLine” method of Twitter class.

```
List<Status> userStatuses = twitter.getUserTimeline(username, page);
```

Figure 6.4. Getting user tweets from user timeline

The final code to get available tweets is the following :

```
For (int pageCount = 1; pageCount <= 32; pageCount++) {
   Paging page = new Paging(pageCount, 100);
   List<Status> userStatuses = twitter.getUserTimeline(username, page);
   // ...
   // process userStatuses list
   // ...
}
```

Figure 6.5. Getting available tweets

6.2. DBpedia SPARQL

DBpedia is a community effort that extracts structured information from Wikipedia and makes this information available on the Web. We can query this information using SPARQL, W3C’s standart query language for RDF. In SweetTweet application we query DBpedia to understand and analyse the tweets of users, and tag users with respect to their interests.

DBpedia provides a SPARQL endpoint at “http://dbpedia.org/sparql” URL where we can send queries and get responses for them. In order to query DBpedia from SweetTweet application, we use Jena, a Java framework for building semantic web applications.

In SweetTweet application we query top keywords of the users to find out what they are interested in and to tag users according to their interests to be able to find connections between them. To provide such a functionality, we implemented different SPARQL queries to be sent to DBpedia endpoint. The syntax of SPARQL query looks like an SQL query, of course with some differences. But the main idea is the same : to filter and retrieve records that satisfy the conditions defined in query from a data storage. First things first; to write a SPARQL query, we need to define some namespaces with “PREFIX” keyword, as shown below. Each query sent to DBpedia includes this “PREFIX” part that associates a short label with a specific URI.

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX : <http://dbpedia.org/resource/>
PREFIX dbpedia2: <http://dbpedia.org/property/>
PREFIX dbpedia: <http://dbpedia.org/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
```

Figure 6.6. Prefixes in SPARQL query

Queries we send to DBpedia aim to find the categories of the keywords. If we can extract the categories of resources by giving the top keywords of the users as input to these queries, then we can take these results as a first step in understanding and analysing what users are talking about in Twitter.

First query we send is shown in Figure 6.7. This query takes a keyword – one of the top keywords of selected user – and tries to fetch broader subject, subject and label information from the resource which rdfs:label predicate matches the keyword.

```
SELECT *
      WHERE {
            ?s rdfs:label \"" + keyword + "\"@en.
            ?s skos:subject ?o.
            ?o skos:broader ?oo
      }
```

Figure 6.7. Broader subject query for given keyword

For example, if we give “Android” keyword as an input to this query, we get the label, subject and broader subject of the resource “Android” defined on semantic web resource. The output of this query is shown in Table 6.1.

Table 6.1. Example result for broader subject query

Keyword	Label	Subject	Broader Subject
Android	Android	Science Fiction Themes Humanoid Robots	In Popular Culture Wikiproject Science Fiction Categories Science Fiction Concepts Biomorphic Robots Robots Anthropomorphism

Depending on the keyword given as input, this query may return an empty result set. It is likely that not all resources available on DBpedia provides a broader subject property. In this case, we simply narrow our search to query for subject and the label of the resource, using the following query.

```
SELECT ?s ?o
      WHERE {
            ?s rdfs:label \"" + keyword + "\"@en.
            ?s skos:subject ?o
      }
```

Figure 6.8. Subject query for given keyword

Same case for broader subject that explained above may apply for subject property of the resource. The queried resources available on DBpedia may not contain a subject property. So, we narrow our query once more. The query listed below returns the label property of the resources which have a rdfs:label property that matches with the given keyword.

```
SELECT ?s
      WHERE {
            ?s rdfs:label \"" + keyword + "\"@en.
      }
```

Figure 6.9. Label query for given keyword

Example results of label query for given keyword “Model” is shown below. “Broader Subject” and “Subject” fields are marked as “Not Applicable” in this example.

Table 6.2. Example result for label query

Keyword	Label	Subject	Broader Subject
Model	Model	[N/A]	[N/A]

If the result set we receive is still empty, we define this keyword as Not Applicable (N/A) in the results pane, as shown below.

Table 6.3. Subject query for given keyword

Keyword	Label	Subject	Broader Subject
Fwd	[N/A]	[N/A]	[N/A]

After querying top keywords of user from DBpedia, we proceed the second phase of our analysis. In this phase, we try to find the mutual categories for top keywords of the user to tag him. To be able to do that, we provide a SPARQL query, as shown below, that searches the resources available in DBpedia if there is any that contains the given keyword within its label. This query has the same structure of SQL queries with “LIKE” statements to provide string pattern match operation. By using the results retrieved from this query, we can find the mutual categories of the keywords that the user utilized most.

```
SELECT DISTINCT ?o
      WHERE {
            ?s rdfs:label ?p .
            ?s skos:subject ?o .
            ?p <bif:contains> \"" + keyword + "\"@en .
      }
```

Figure 6.10. String pattern match query

For example, we have a user whose top two keywords are Android and Google. When we query these keywords individually, we find out that Android is some kind of humanoid robot while Google is a company. The query listed above tries to find mutual categories of these two keywords. When we send query Android and Google to find their mutual categories, we get the following result which is quite promising.

Table 6.4. Mutual categories found for “Android” and “Google” keywords

Mutual Categories
Cloud Clients Google Google Acquisitions Mobile Software Android Operating System Mobile Phone Operating Systems Smartphones Mobile Linux Mobile Open Source Embedded Linux

6.3. SweetTweet

SweetTweet is a web application implemented in Java using Spring Framework [40] and MySQL [41] as database. The application aims to understand and analyse users’ interests by processing their posts – tweets – sent to Twitter. SweetTweet also provides common interest areas of users which may finally construct a network between users with same interests. To offer such a functionality, the application needs to retrieve users and their posts from Twitter via an graphical user interface that provides ease-of-use of the application. Below are the two sections that gives detailed information about the implemantation of the application. The former section explains the data model used in SweetTweet. The latter section provides the details of web application structure.

6.3.1. SweetTweet Application

In our model, we aim to understand the content of the user posts sent to Twitter, called “tweets”, to find out what the users are talking about or what they are really interested in. To be able to do that, we need to retrieve user posts. Thanks to the API provided by Twitter, we can access to Twitter resources and can get this data. Once we retrieve the tweets of the user, we store them in our database to do the further processing. The next step in our application is to parse user’s tweets into words in order to find the keywords that the user utilized frequently. While we parse the tweets into keywords, we eliminate stopwords, such as “the”, “and”, “to”, etc., to provide beter understanding of what user is talking about in Twitter. When we have the keywords stored in database, we can proceed with analysis process. In analysis process, we send SPARQL queries to DBpedia using top keywords of the user to find out the categories of these keywords. During this analysis process, if we can find matching categories for the top keywords of user, we can infer that the user is interested in these subjects. Furthermore, we can find relations between users which are interested in the same categories.

Below, we provide the overall architecture of our model, including the retrieval and analysis of user tweets from Twitter.

SweetTweet is a web application written in Java using Spring Framework. And it needs a servlet container, like Tomcat, to be deployed and run. After deploying the web application and configuring database parameters, SweetTweet GUI is up, running and accessible from the URL below :

http://<hostname-or-ip-address-of-server>/SweetTweet

When you type this URL into the address bar of your browser, you get a login page (index.jsp) to check the authentication of the application for given username and password, as shown in Figure 6.11.

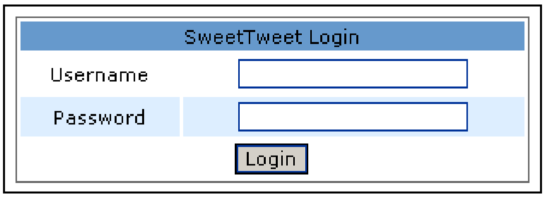


Figure 6.11. SweetTweet login screen

When user hits the “Login” button, LoginFormController.java gets username and password from login page and checks user authentication. After a successful login operation, page is redirected to Main.jsp. Beginning with Main.jsp, each JSP includes LeftMenu.jsp that renders user menu and user information as shown in Figure 6.12 and Figure 6.13, respectively. The users of SweetTweet can select “Retrieve User” link to add new Twitter users to SweetTweet or to update – synchronize – the data in SweetTweet database. “Analyse User” link is used for analysing tweets of users retrieved from Twitter. “Search Concept” is used for finding the Twitter users tagged by SweetTweet with the given concepts. Finally, “Logout” link simply logs out and terminates user session.

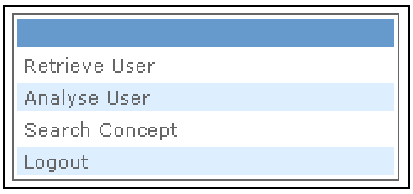
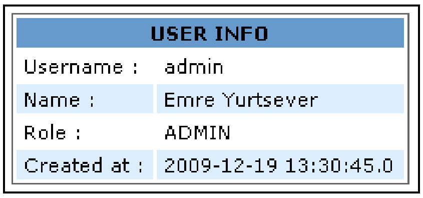


Figure 6.12. User menu

User Info table provides information about the username, name, role and creation date of the logged user.



USER INFO	
Username :	admin
Name :	Emre Yurtsever
Role :	ADMIN
Created at :	2009-12-19 13:30:45.0

Figure 6.13. Information about the system user

Figure 6.14 shows the flow of the application for user retrieval process :

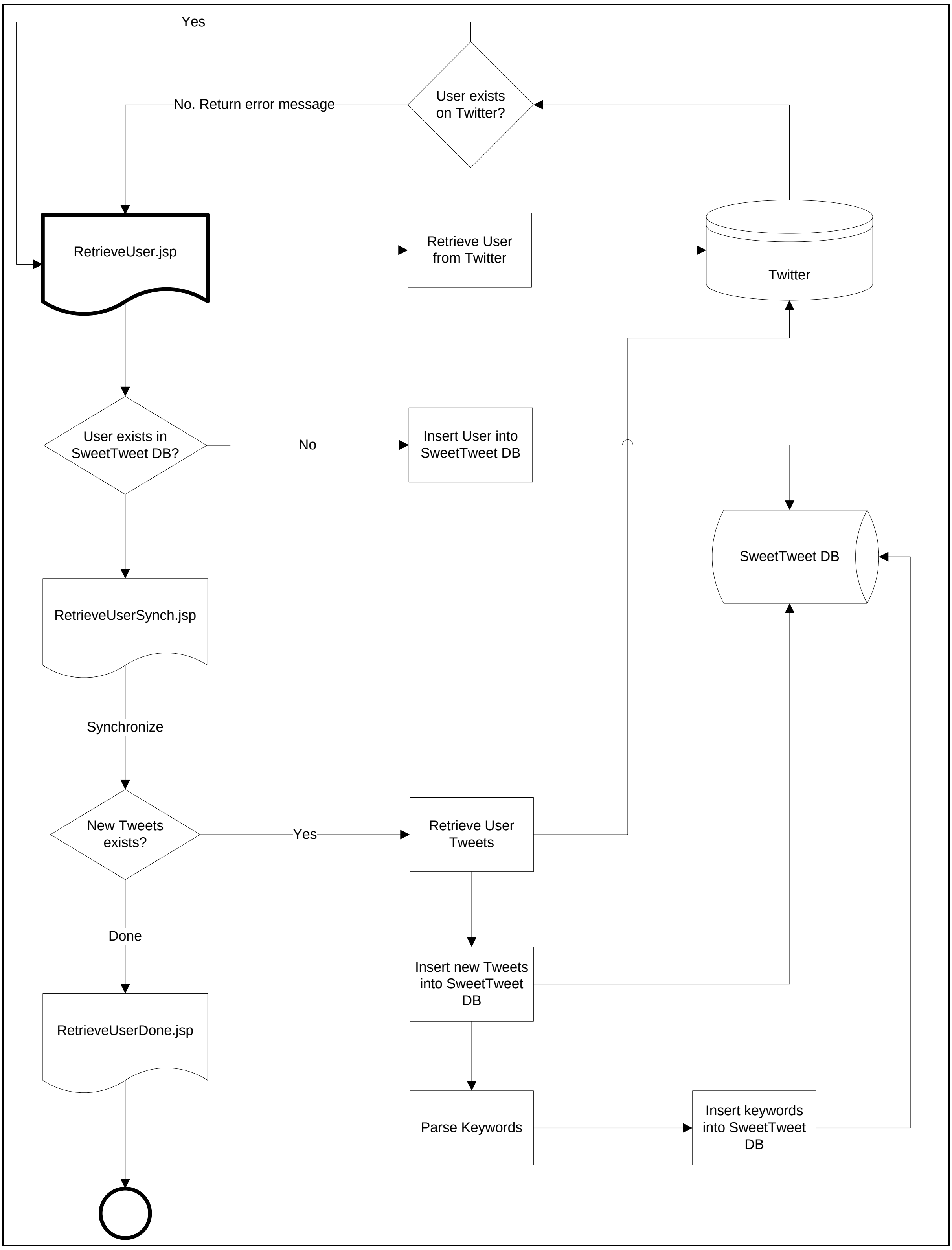
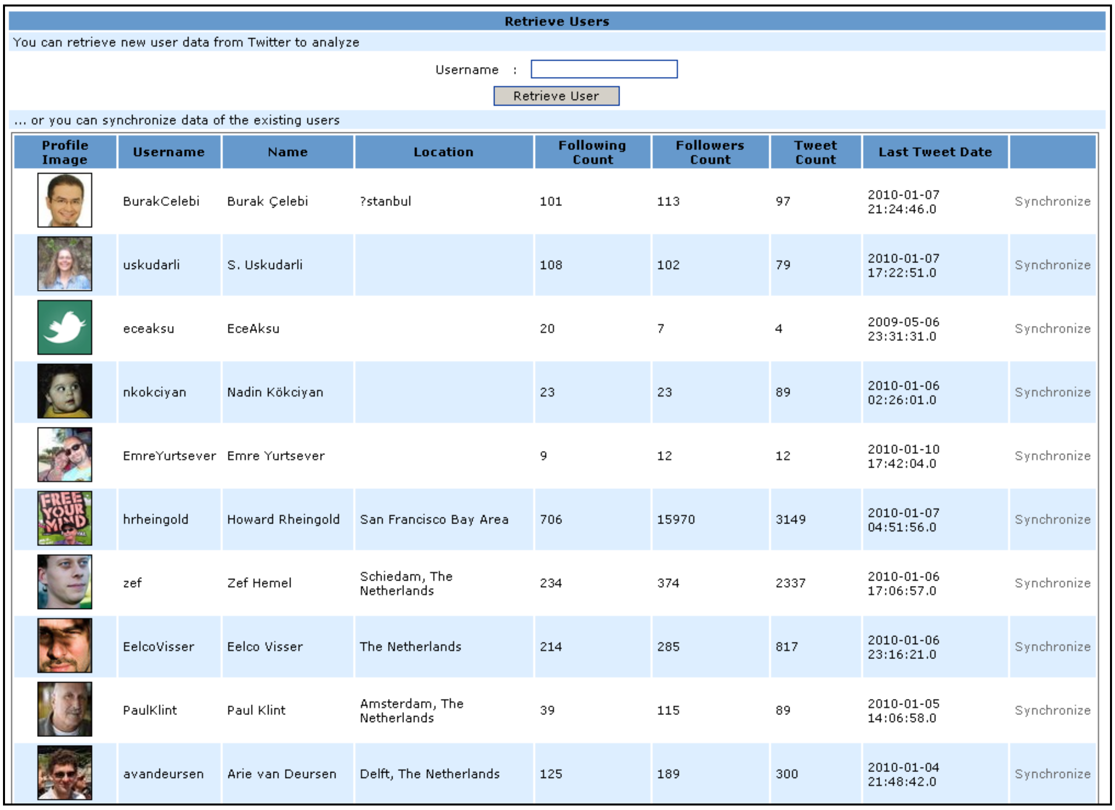


Figure 6.14. Retrieve user process

When user clicks “Retrieve User” menu, RetrieveUser.jsp is displayed filled with the list of Twitter users retrieved before. User can enter a Twitter username to retrieve, or select any of user from user list displayed to update – synchronize with Twitter - its information on SweetTweet database. Screenshot of Retrieve User screen is displayed below.



Retrieve Users

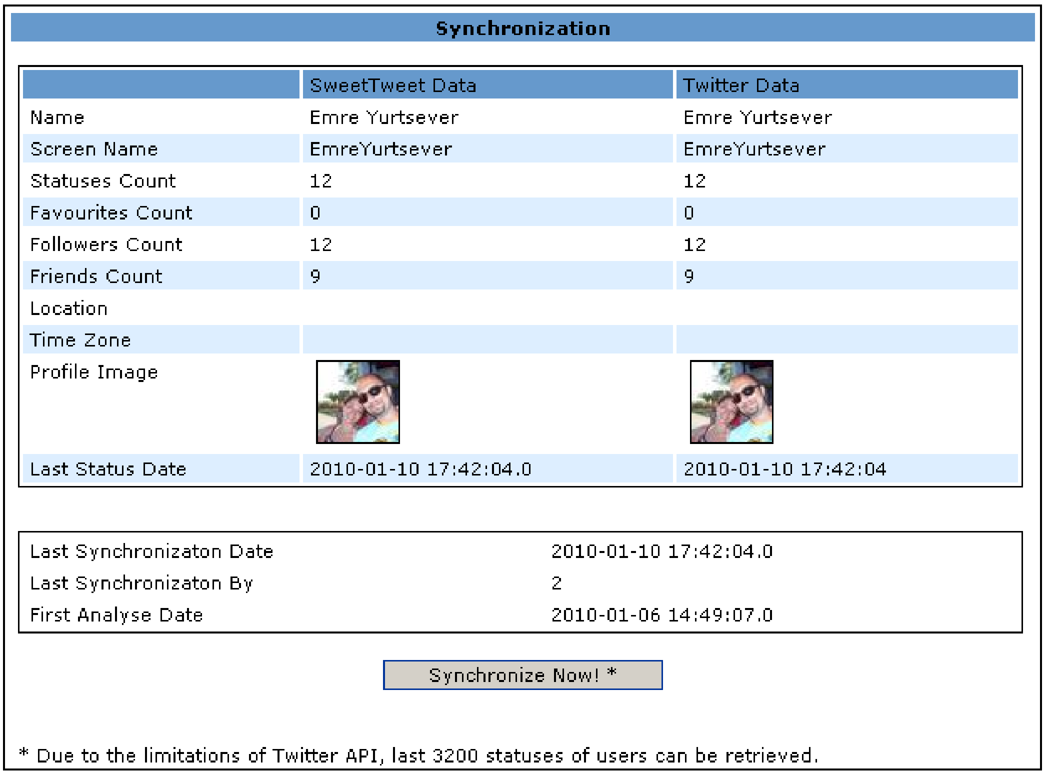
You can retrieve new user data from Twitter to analyze

... or you can synchronize data of the existing users

Profile Image	Username	Name	Location	Following Count	Followers Count	Tweet Count	Last Tweet Date	
	BurakCelebi	Burak Çelebi	?stanbul	101	113	97	2010-01-07 21:24:46.0	Synchronize
	uskudarli	S. Uskudarli		108	102	79	2010-01-07 17:22:51.0	Synchronize
	eceaksu	EceAksu		20	7	4	2009-05-06 23:31:31.0	Synchronize
	nkokciyan	Nadin Kökciyan		23	23	89	2010-01-06 02:26:01.0	Synchronize
	EmreYurtsever	Emre Yurtsever		9	12	12	2010-01-10 17:42:04.0	Synchronize
	hrheingold	Howard Rheingold	San Francisco Bay Area	706	15970	3149	2010-01-07 04:51:56.0	Synchronize
	zef	Zef Hemel	Schiedam, The Netherlands	234	374	2337	2010-01-06 17:06:57.0	Synchronize
	EelcoVisser	Eelco Visser	The Netherlands	214	285	817	2010-01-06 23:16:21.0	Synchronize
	PaulKlint	Paul Klint	Amsterdam, The Netherlands	39	115	89	2010-01-05 14:06:58.0	Synchronize
	avandeursen	Arie van Deursen	Delft, The Netherlands	125	189	300	2010-01-04 21:48:42.0	Synchronize

Figure 6.15. Retrieve user screen

Synchronization screen is displayed when user enters a username and presses “Retrieve User” or selects one of the existing users from the list. In this screen, the data kept in SweetTweet and the data on Twitter for this username are retrieved and displayed. If given username does not exists in SweetTweet database, it is created for the first time with the information retrieved from Twitter. “SweetTweet Data” on the left side shows the information on SweetTweet database, while “Twitter Data”, as the name implies, is up-to-date data retrieved from Twitter. To synchronize the information held for this user, “Synchronize Now!” button should be clicked. Due to the limitations of Twitter API, last 3200 statuses of users can be retrieved.



Synchronization

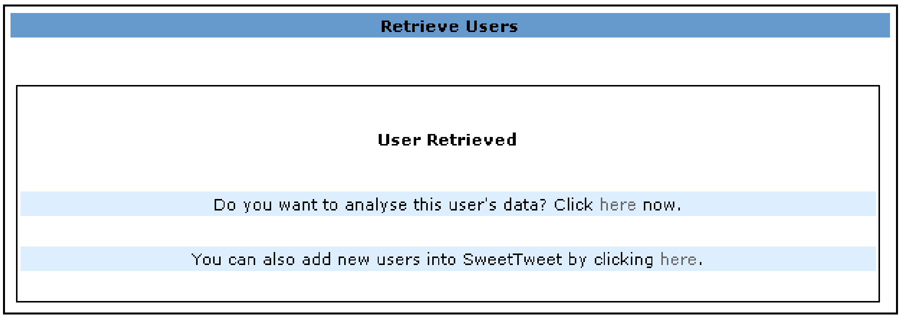
	SweetTweet Data	Twitter Data
Name	Emre Yurtsever	Emre Yurtsever
Screen Name	EmreYurtsever	EmreYurtsever
Statuses Count	12	12
Favourites Count	0	0
Followers Count	12	12
Friends Count	9	9
Location		
Time Zone		
Profile Image		
Last Status Date	2010-01-10 17:42:04.0	2010-01-10 17:42:04

Last Synchronizaton Date	2010-01-10 17:42:04.0
Last Synchronizaton By	2
First Analyse Date	2010-01-06 14:49:07.0

* Due to the limitations of Twitter API, last 3200 statuses of users can be retrieved.

Figure 6.16. User synchronization screen

When a SweetTweet user clicks “Synchronize Now!” button, user data in SweetTweet database is updated from Twitter. If Twitter user posted new tweets after last synchronization with SweetTweet, these tweets are also retrieved, inserted into database, parsed and inserted into keyword table. In this case, a new analysis is needed to tag the user. If synchronization completes successfully “User Retrieved” message is displayed on the screen. SweetTweet users can directly navigate to analyse page to analyse this user, or can retrieve other users by clicking the given links on the page shown below.



Retrieve Users

User Retrieved

Do you want to analyse this user's data? Click here now.

You can also add new users into SweetTweet by clicking here.

Figure 6.17. Synchronization completed screen

Figure 6.18 shows the flow of the application for user analysis process :

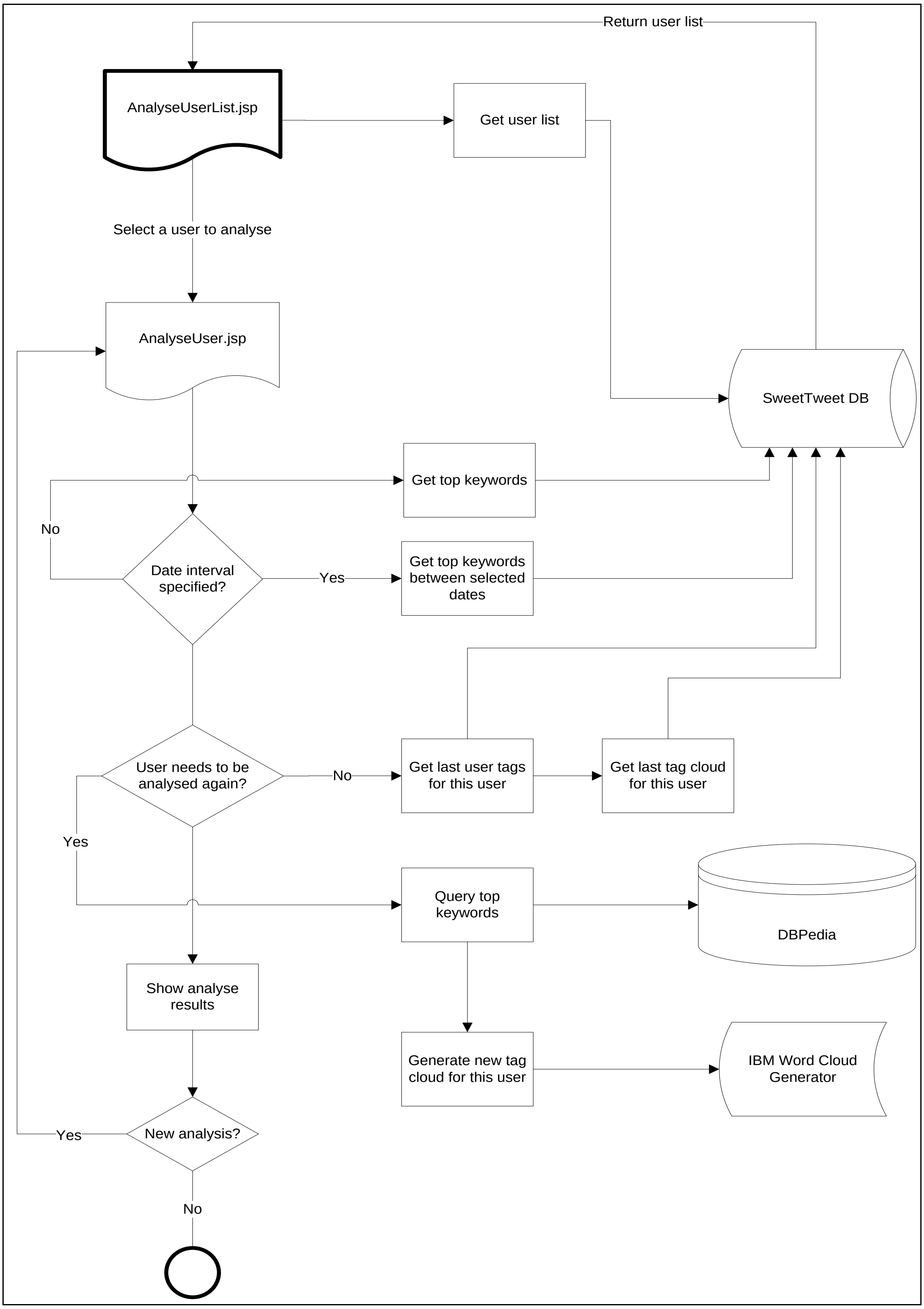
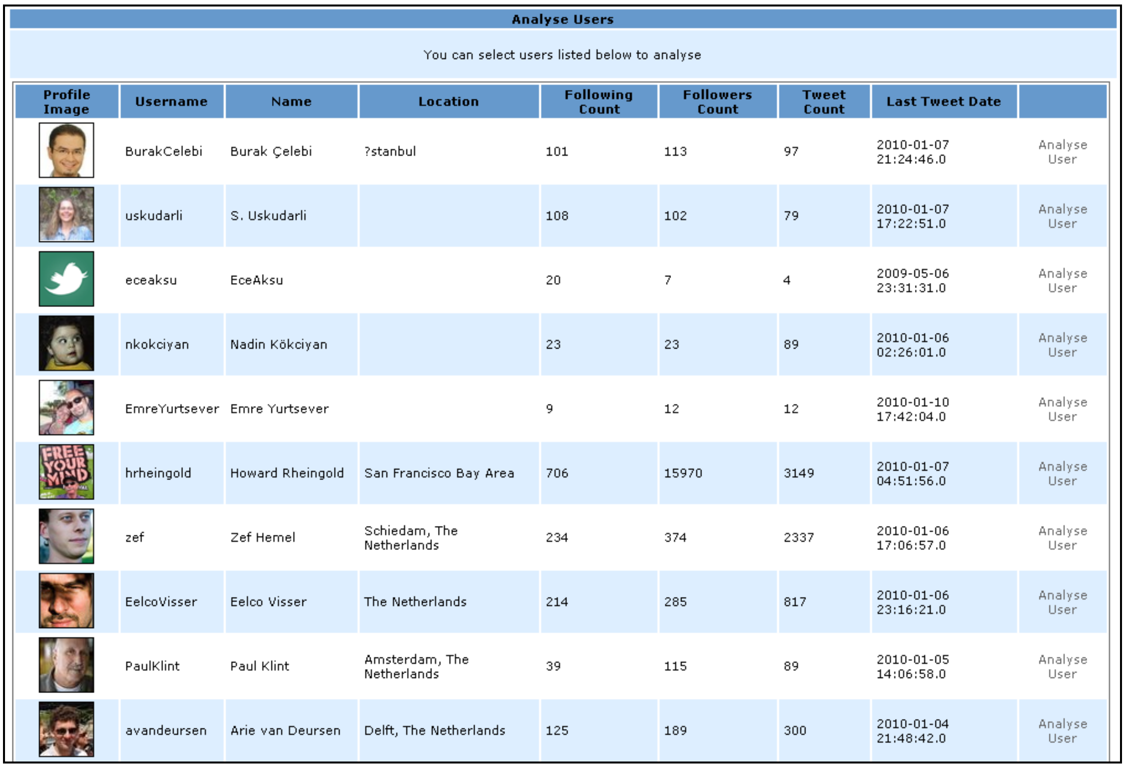


Figure 6.18. Analyse user process

When “Analyse User” link is clicked, the users retrieved from Twitter and stored in SweetTweet database are listed as shown below. AnalyseUserList.jsp shows information about the users listed and provides a link, called “Analyse User”, to analyse user’s tweets.



Analyse Users

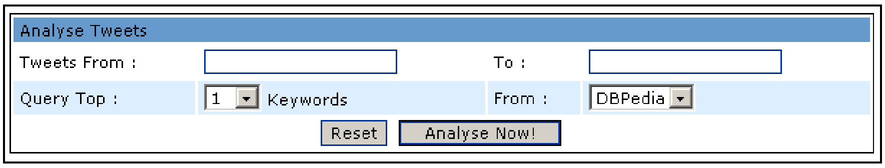
You can select users listed below to analyse

Profile Image	Username	Name	Location	Following Count	Followers Count	Tweet Count	Last Tweet Date	
	BurakCelebi	Burak Çelebi	?stanbul	101	113	97	2010-01-07 21:24:46.0	Analyse User
	uskudarli	S. Uskudarli		108	102	79	2010-01-07 17:22:51.0	Analyse User
	eceaksu	EceAksu		20	7	4	2009-05-06 23:31:31.0	Analyse User
	nkokciyan	Nadin Kökciyan		23	23	89	2010-01-06 02:26:01.0	Analyse User
	EmreYurtsever	Emre Yurtsever		9	12	12	2010-01-10 17:42:04.0	Analyse User
	hrheingold	Howard Rheingold	San Francisco Bay Area	706	15970	3149	2010-01-07 04:51:56.0	Analyse User
	zef	Zef Hemel	Schiedam, The Netherlands	234	374	2337	2010-01-06 17:06:57.0	Analyse User
	EelcoVisser	Eelco Visser	The Netherlands	214	285	817	2010-01-06 23:16:21.0	Analyse User
	PaulKlint	Paul Klint	Amsterdam, The Netherlands	39	115	89	2010-01-05 14:06:58.0	Analyse User
	avandeursen	Arie van Deursen	Delft, The Netherlands	125	189	300	2010-01-04 21:48:42.0	Analyse User

Figure 6.19. User list to analyse

AnalyseUser.jsp shows the results of the analysis made by SweetTweet application. It displays user info on the top of the page. Users can apply filters to default analysis mechanism. By default, top 5 keywords of all times are queried from DBpedia. But SweetTweet users can define the keyword count to be analysed. Users can also specify the date interval that they want to analyse.

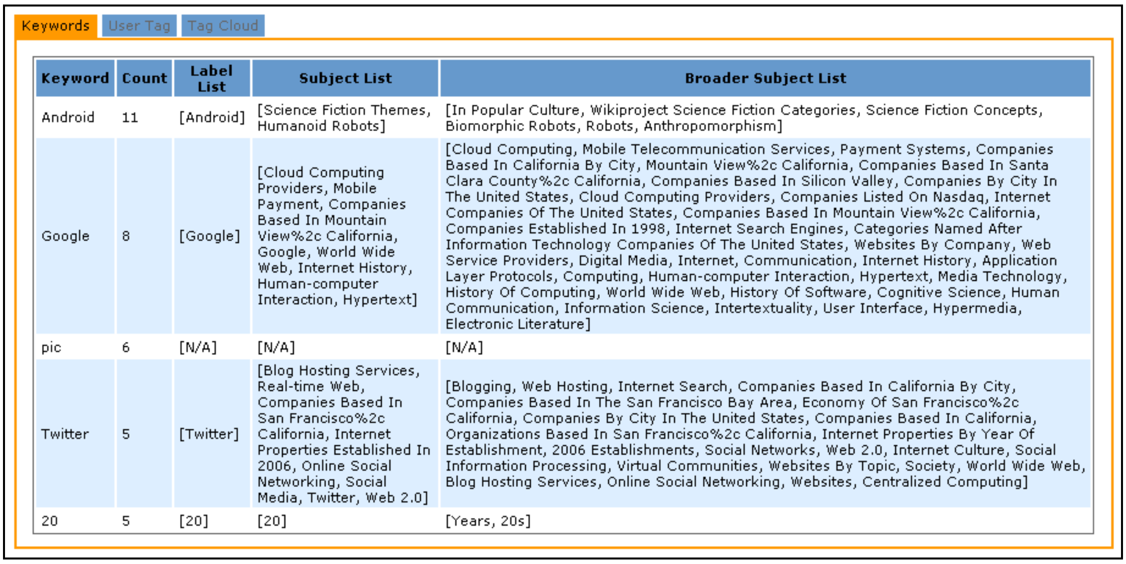
For example; a SweetTweet user may want to analyse a user’s top 15 keywords used in tweets between 2009-10-17 and 2010-01-20. When user hits the “Analyse Now!” button, analysis process is executed again for selected keyword count between given dates.



Tweets From :		To :	
Query Top :	1 Keywords	From :	DBPedia

Figure 6.20. Analyse options

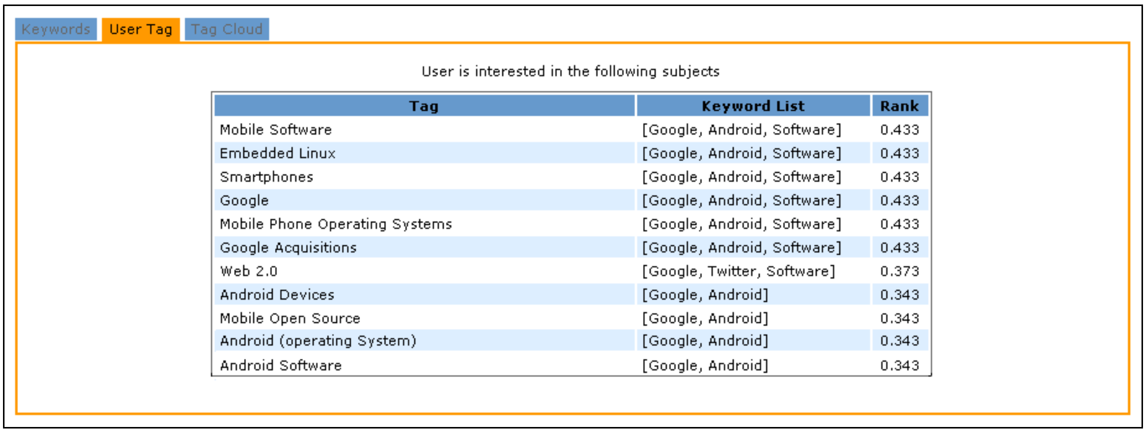
Analysis page has 3 tabs that display the analysis results. First tab is “Keyword” tab that shows the top keywords with respect to their frequencies of use. Categories of these keywords are queried from DBpedia by calling DBPediaQueryManager.java. DBpedia provides different attributes for different resources. One resource may have a broader subject attribute, other may have subject attribute, and another may have only label attribute. It is also possible not to find any resource on DBpedia for queried keywords. In this case we label them as “[N/A]” (not applicable).



Keyword	Count	Label List	Subject List	Broader Subject List
Android	11	[Android]	[Science Fiction Themes, Humanoid Robots]	[In Popular Culture, Wikiproject Science Fiction Categories, Science Fiction Concepts, Biomorphic Robots, Robots, Anthropomorphism]
Google	8	[Google]	[Cloud Computing Providers, Mobile Payment, Companies Based In Mountain View%2c California, Google, World Wide Web, Internet History, Human-computer Interaction, Hypertext]	[Cloud Computing, Mobile Telecommunication Services, Payment Systems, Companies Based In California By City, Mountain View%2c California, Companies Based In Santa Clara County%2c California, Companies Based In Silicon Valley, Companies By City In The United States, Cloud Computing Providers, Companies Listed On Nasdaq, Internet Companies Of The United States, Companies Based In Mountain View%2c California, Companies Established In 1998, Internet Search Engines, Categories Named After Information Technology Companies Of The United States, Websites By Company, Web Service Providers, Digital Media, Internet, Communication, Internet History, Application Layer Protocols, Computing, Human-computer Interaction, Hypertext, Media Technology, History Of Computing, World Wide Web, History Of Software, Cognitive Science, Human Communication, Information Science, Intertextuality, User Interface, Hypermedia, Electronic Literature]
pic	6	[N/A]	[N/A]	[N/A]
Twitter	5	[Twitter]	[Blog Hosting Services, Real-time Web, Companies Based In San Francisco%2c California, Internet Properties Established In 2006, Online Social Networking, Social Media, Twitter, Web 2.0]	[Blogging, Web Hosting, Internet Search, Companies Based In California By City, Companies Based In The San Francisco Bay Area, Economy Of San Francisco%2c California, Companies By City In The United States, Companies Based In California, Organizations Based In San Francisco%2c California, Internet Properties By Year Of Establishment, 2006 Establishments, Social Networks, Web 2.0, Internet Culture, Social Information Processing, Virtual Communities, Websites By Topic, Society, World Wide Web, Blog Hosting Services, Online Social Networking, Websites, Centralized Computing]
20	5	[20]	[20]	[Years, 20s]

Figure 6.21. Top keywords

Second tab on this page is called “User Tag”. In this tab we can see the list of the subjects that the user is interested in according to the analysis results. Analysis results shows the mutual categories that the top keywords of the users belong. For example, DBpedia returns the category of the resource “Android” as “Robots” or “Science Fiction Concepts”. But we have a second keyword, “Google”, that may give us a lead about the interests of the user. Do “Android” and “Google” have a common category? They do have! Android is the mobile operating system of Google. It is clear that we are not interested in “Robots” category in this example. We need to find such relations to be able to understand what user is talking about. With the analysis algorithm we have in this study, we can find the mutual categories for resources, so that we can tag user with appropriate subject, as shown below.



User is interested in the following subjects

Tag	Keyword List	Rank
Mobile Software	[Google, Android, Software]	0.433
Embedded Linux	[Google, Android, Software]	0.433
Smartphones	[Google, Android, Software]	0.433
Google	[Google, Android, Software]	0.433
Mobile Phone Operating Systems	[Google, Android, Software]	0.433
Google Acquisitions	[Google, Android, Software]	0.433
Web 2.0	[Google, Twitter, Software]	0.373
Android Devices	[Google, Android]	0.343
Mobile Open Source	[Google, Android]	0.343
Android (operating System)	[Google, Android]	0.343
Android Software	[Google, Android]	0.343

Figure 6.22. Mutual categories

Third and the last tab shows the tag cloud, as shown below, generated by TagCloudGenerator class. The size of the keywords in tag cloud vary with respect to the frequencies of usage. The more frequent keyword use, the bigger keyword size. By default, top 50 keywords are retrieved, but this value can be configured. One example of generated tag cloud is shown below.

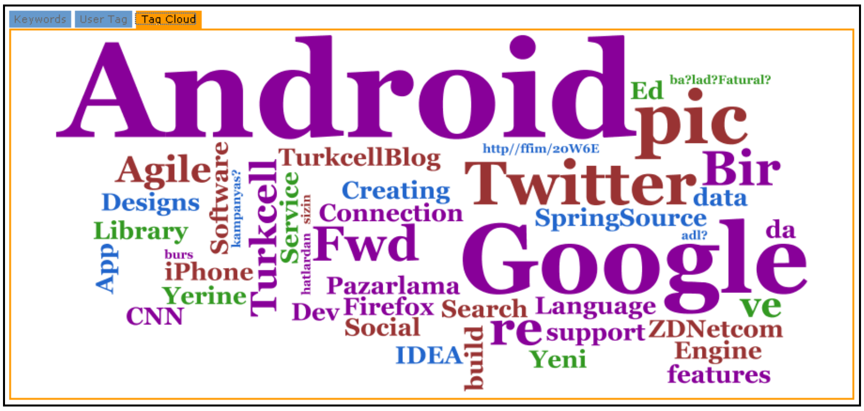


Figure 6.23. Tag cloud

To avoid unnecessary analysis and to increase performance of the application, second analysis for same data is blocked programatically. If the data kept in SweetTweet database is changed after an analysis operation, then the NEEDS_ANALYSE field is set to "1". When analyse user screen is loaded, this parameter is also checked to see if we need a new analysis for this user to find up-to-date tags. If so, analysis process is executed, mutual categories are found, new tag cloud is generated and analysis results are stored in database again. After completing the analysis process, application sets NEEDS_ANALYSE flag to "0" to indicate that no more analysis is needed until a new tweet is retrieved and stored in the SweetTweet database.

Another functionality provided by our model is concept searching. To search and filter the concepts we found and store in our database, we need to provide a keyword as input. Users can be searched and displayed according to this keyword. Figure 6.24. shows the flow of concept search process.

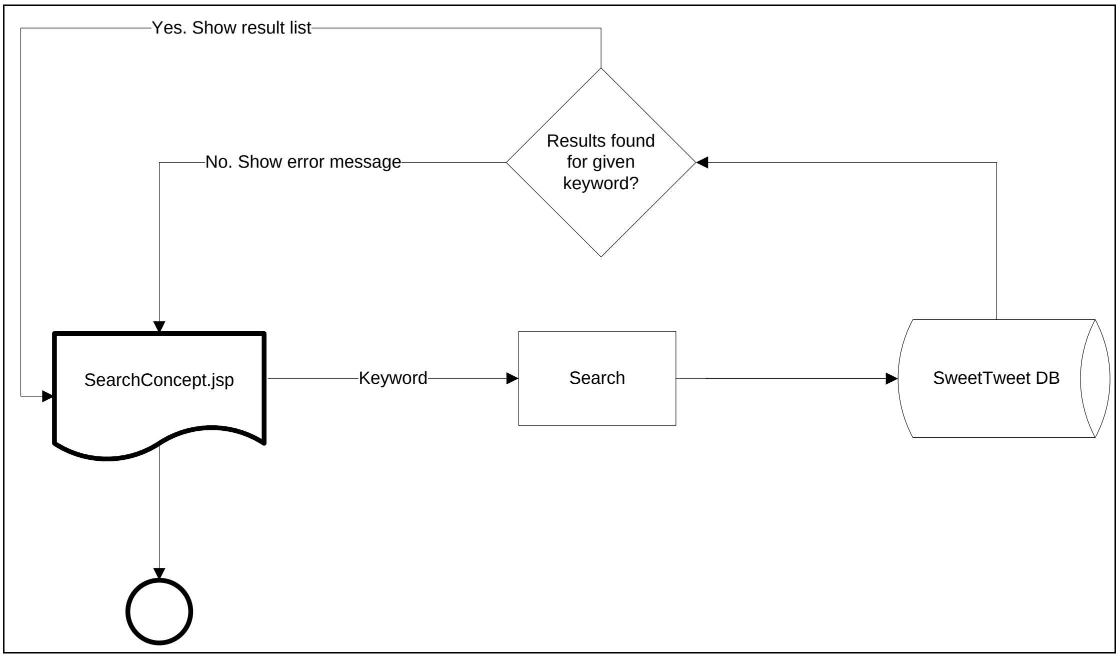
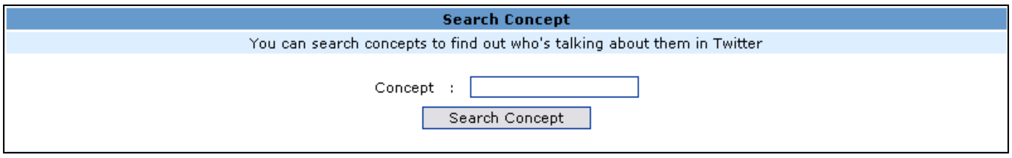


Figure 6.24. Search concept process flow

When user clicks "Search Concept" link on the menu, a search screen is displayed, as shown in the figure below. Users enter a concept name to be searched through the results of the analysis we performed for Twitter users.



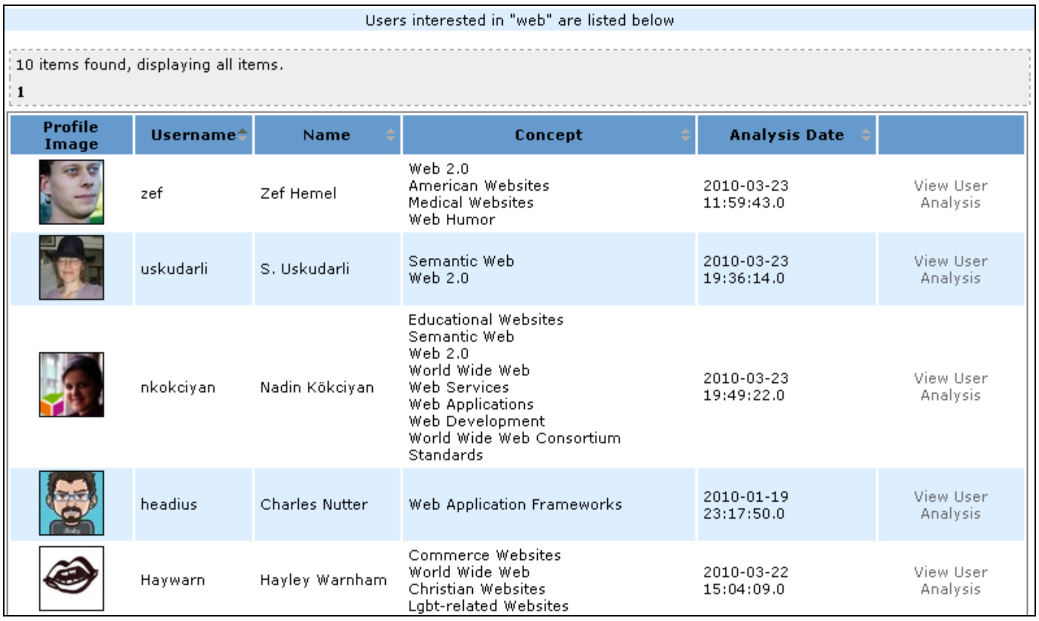
Search Concept

You can search concepts to find out who's talking about them in Twitter

Concept :

Figure 6.25. Search Concept

For a given keyword as input, the analysis results stored in SweetTweet database are searched and the users that tagged with concepts containing given keyword are listed. Figure 6.26 shows an example result of the search performed for category "web". The users who are interested in "web" are listed. For example, the user "zef" is tagged as he is interested in "Web 2.0", "American Websites", "Medical Websites", "Web Humor" categories, which all contains our search keyword "web".



Users interested in "web" are listed below

10 items found, displaying all items.
1

Profile Image	Username	Name	Concept	Analysis Date	
	zef	Zef Hemel	Web 2.0 American Websites Medical Websites Web Humor	2010-03-23 11:59:43.0	View User Analysis
	uskudarli	S. Uskudarli	Semantic Web Web 2.0	2010-03-23 19:36:14.0	View User Analysis
	nkokciyan	Nadin Kökciyan	Educational Websites Semantic Web Web 2.0 World Wide Web Web Services Web Applications Web Development World Wide Web Consortium Standards	2010-03-23 19:49:22.0	View User Analysis
	headius	Charles Nutter	Web Application Frameworks	2010-01-19 23:17:50.0	View User Analysis
	Haywarn	Hayley Warnham	Commerce Websites World Wide Web Christian Websites Lgbt-related Websites	2010-03-22 15:04:09.0	View User Analysis

Figure 6.26. Search results for a given concept

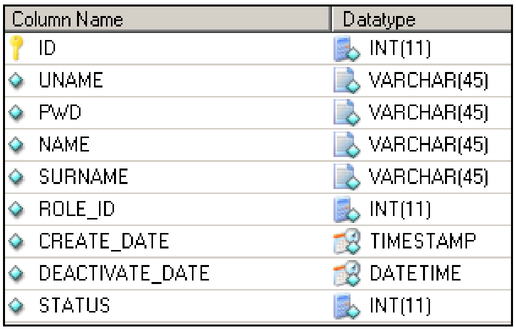
6.3.2. Data Model of SweetTweet

SweetTweet application persists its data in MySQL database. The database schema created for this application is also called "SweetTweet" and it contains the following tables.

Table 6.5. SweetTweet database tables

Table Name	Comments
USERS	Keeps the data of users of SweetTweet application
USER_ROLE	Roles of users defined in the system
TUSER	Users retrieved from Twitter to analyse
TUSER_TWEETS	Tweets of retrieved Twitter users
TUSER_KEYWORD	Keywords in user tweets
TUSER_TAGS	Tags assigned to users after completing analysis process
TUSER_ANALYSE	Analysis results for Twitter users
STOPWORDS	Keywords to be omitted in user tweets while inserting into TUSER_KEYWORD
TWITTER_STOPWORDS	Twitter specific keywords to be omitted in user tweets while inserting into TUSER_KEYWORD
PUNCTUATION	Puncutation marks to be ommited in user tweets while inserting into TUSER_KEYWORD

6.3.2.1. USERS Table. This table keeps the data of the users of SweetTweet application. Fields in this table and their explanations are given in Figure 6.27 :

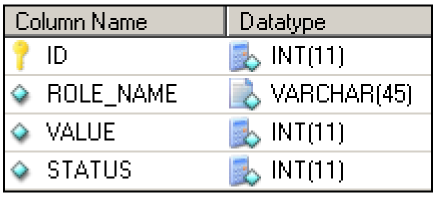


Column Name	Datatype
ID	INT(11)
UNAME	VARCHAR(45)
PWD	VARCHAR(45)
NAME	VARCHAR(45)
SURNAME	VARCHAR(45)
ROLE_ID	INT(11)
CREATE_DATE	TIMESTAMP
DEACTIVATE_DATE	DATETIME
STATUS	INT(11)

Figure 6.27. USERS table

- ID : Auto-incremented id, also primary key of the table
- UNAME : Username
- PWD : Password
- NAME : Name of the user
- SURNAME : Surname of the user
- ROLE_ID : Role of the user
- CREATE_DATE : Date of creation
- DEACTIVATE_DATE : Date of deactvation
- STATUS : 0 (Active) – 1 (Deactive)

6.3.2.2. USER_ROLE Table. USER_ROLE table defines the access levels of users in SweetTweet system. Users of SweetTweet may have ADMIN, MEMBER or VISITOR priviliges according to role id defined in USERS table. Fields in this table and their explanations are given in Figure 6.28:

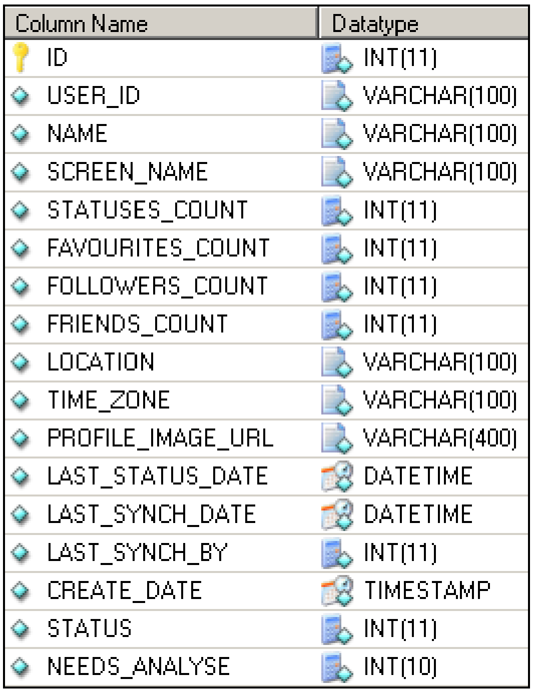


Column Name	Datatype
ID	INT(11)
ROLE_NAME	VARCHAR(45)
VALUE	INT(11)
STATUS	INT(11)

Figure 6.28. USER_ROLE table

- ID : Auto-incerementеd id, also primary key of the table
- ROLE_NAME : Name of the role
- VALUE : Value of the role
- STATUS : 0 (Active) – 1 (Deactive)

6.3.2.3. TUSER Table. TUSER table holds information about the users retrieved from Twitter. Data on this table is updated with synchronization process defined in SweetTweet GUI. Fields in this table and their explanations are given below :



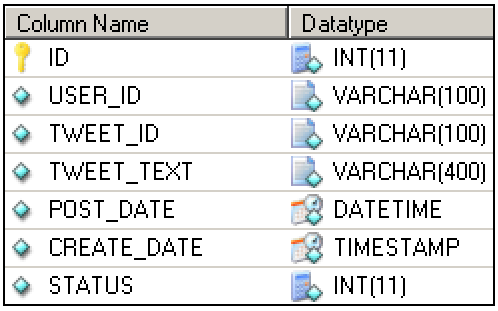
Column Name	Datatype
ID	INT(11)
USER_ID	VARCHAR(100)
NAME	VARCHAR(100)
SCREEN_NAME	VARCHAR(100)
STATUSES_COUNT	INT(11)
FAVOURITES_COUNT	INT(11)
FOLLOWERS_COUNT	INT(11)
FRIENDS_COUNT	INT(11)
LOCATION	VARCHAR(100)
TIME_ZONE	VARCHAR(100)
PROFILE_IMAGE_URL	VARCHAR(400)
LAST_STATUS_DATE	DATETIME
LAST_SYNCH_DATE	DATETIME
LAST_SYNCH_BY	INT(11)
CREATE_DATE	TIMESTAMP
STATUS	INT(11)
NEEDS_ANALYSE	INT(10)

Figure 6.29. TUSER table

- ID : Auto-inceremented id, also primary key of the table
- USER_ID : Twitter user id
- NAME : Twitter name
- SCREEN_NAME : Twitter screen name
- STATUSES_COUNT : Status count retrieved from Twitter
- FAVOURITES_COUNT : Favourite count
- FOLLOWERS_COUNT : Followers count
- FRIENDS_COUNT : Friends count
- LOCATION : Location
- TIME_ZONE : Time Zone
- PROFILE_IMAGE_URL : Profile Image URL
- LAST_STATUS_DATE : Last status date
- LAST_SYNCH_DATE : Last synchronization with Twitter
- LAST_SYNCH_BY : Last synchronization is made by user id
- CREATE_DATE : Create date of this row
- STATUS : 0 (Active) – 1 (Deactive)

- NEEDS_ANALYSE : 0 (No) – 1 (Yes)

6.3.2.4. TUSER_TWEET Table. TUSER_TWEET table holds the tweets of the user. It contains user id, tweet id, tweet text, and post date of the tweets received from Twitter. Fields in this table and their explanations are given in Figure 6.30 :

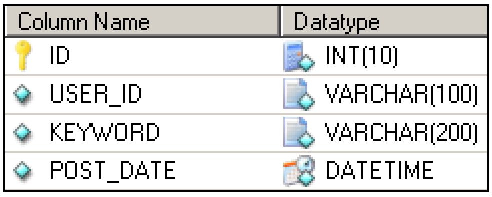


Column Name	Datatype
ID	INT(11)
USER_ID	VARCHAR(100)
TWEET_ID	VARCHAR(100)
TWEET_TEXT	VARCHAR(400)
POST_DATE	DATETIME
CREATE_DATE	TIMESTAMP
STATUS	INT(11)

Figure 6.30. TUSER_TWEET table

- ID : Auto-incerremented id, also primary key of the table
- USER_ID : User id
- TWEET_ID : Tweet id in Twitter
- TWEET_TEXT : Text of the tweet
- POST_DATE : Post date of the tweet
- CREATE_DATE : Create date of this record
- STATUS : 0 (Active) – 1 (Deactive)

6.3.2.5. TUSER_KEYWORD Table. TUSER_KEYWORD is the table that holds the words parsed from user’s tweets. It also includes the date the keyword is used in corresponding tweet. Fields in this table and their explanations are given in Figure 6.31 :

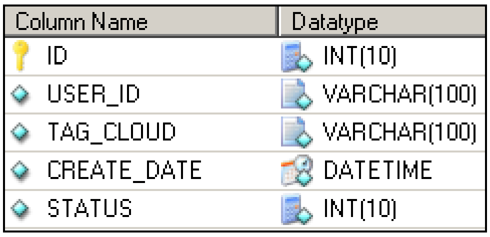


Column Name	Datatype
ID	INT(10)
USER_ID	VARCHAR(100)
KEYWORD	VARCHAR(200)
POST_DATE	DATETIME

Figure 6.31. TUSER_KEYWORD table

- ID : Auto-inceremented id, also primary key of the table
- USER_ID : User id
- KEYWORD : Keyword parsed from user’s tweets
- POST_DATE : Date of the tweet that contains this keyword

6.3.2.6. TUSER_ANALYSE Table. TUSER_ANALYSE table holds the information of the analysis made for SweetTweet users. Fields in this table and their explanations are given below :

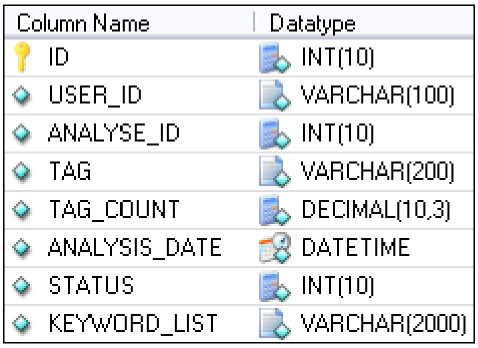


Column Name	Datatype
ID	INT(10)
USER_ID	VARCHAR(100)
TAG_CLOUD	VARCHAR(100)
CREATE_DATE	DATETIME
STATUS	INT(10)

Figure 6.32. TUSER_ANALYSE table

- ID : Auto-inceremented id, also primary key of the table
- USER_ID : User id
- TAG_CLOUD : Tag cloud image path
- CREATE_DATE : Date of creation of this record
- STATUS : 0 (Active) – 1 (Deactive)

6.3.2.7. TUSER_TAGS Table. TUSER_TAGS is the table that contains the details of analysis made for users. Tag field holds the subject that the user is interested in according to the analysis made base on the tweets of the user. Fields in this table and their explanations are given below :

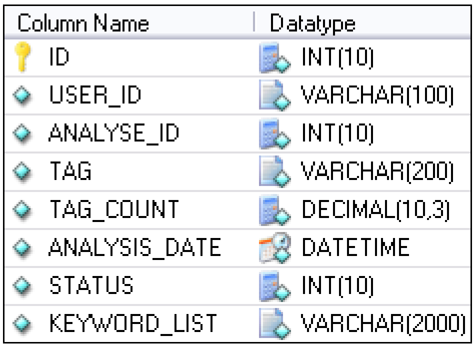


Column Name	Datatype
ID	INT(10)
USER_ID	VARCHAR(100)
ANALYSE_ID	INT(10)
TAG	VARCHAR(200)
TAG_COUNT	DECIMAL(10,3)
ANALYSIS_DATE	DATETIME
STATUS	INT(10)
KEYWORD_LIST	VARCHAR(2000)

Figure 6.33. TUSER_TAGS table

- ID : Auto-inceremented id, also primary key of the table
- USER_ID : User id
- ANALYSE_ID : Reference to TUSER_ANALYSIS
- TAG : User tag generated after analysis process
- TAG_COUNT : Occurence count of the tag in the analysis
- ANALYSIS_DATE : Date of the analysis
- STATUS : 0 (Active) – 1 (Deactive)
- KEYWORD_LIST : List of keywords which are used in analysis process providing the user tag

6.3.2.8. TUSER_UPPERTAGS Table. TUSER_UPPERTAGS is the table that contains the categories of the tags we found for users and stored to TUSER_TAGS table. Fields in this table and their explanations are given below :

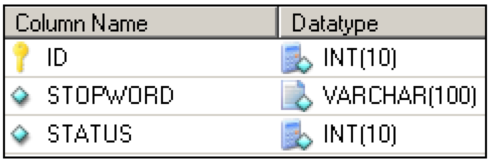


Column Name	Datatype
ID	INT(10)
USER_ID	VARCHAR(100)
ANALYSE_ID	INT(10)
TAG	VARCHAR(200)
TAG_COUNT	DECIMAL(10,3)
ANALYSIS_DATE	DATETIME
STATUS	INT(10)
KEYWORD_LIST	VARCHAR(2000)

Figure 6.34 TUSER_TAGS table

- ID : Auto-inceremented id, also primary key of the table
- USER_ID : User id
- ANALYSE_ID : Reference to TUSER_ANALYSIS
- TAG : User tag generated after analysis process
- TAG_COUNT : Occurence count of the tag in the analysis
- ANALYSIS_DATE : Date of the analysis
- STATUS : 0 (Active) – 1 (Deactive)
- KEYWORD_LIST : List of keywords which are used in analysis process providing the user tag

6.3.2.9. STOPWORDS Table. STOPWORDS table keeps the keywords that will be ignored during tweet parsing process. Thus, commonly used English words, such as “always”, “often”, “or”, “the” etc., will not be counted as input for our analysis process. Fields in this table and their explanations are given below :

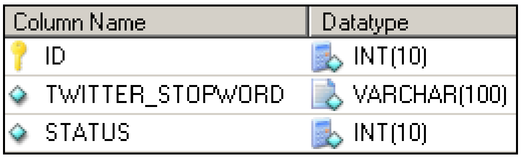


Column Name	Datatype
ID	INT(10)
STOPWORD	VARCHAR(100)
STATUS	INT(10)

Figure 6.35. STOPWORDS table

- ID : Auto-inceremented id, also primary key of the table
- STOPWORD : Keyword to be ignored
- STATUS : 0 (Active) – 1 (Deactive)

<u>6.3.2.10. TWITTER_STOPWORDS Table.</u> TWITTER_STOPWORDS table keeps the keywords which are used frequently in Twitter, thus it provides Twitter specific keyword list. Keywords in this table will also be ignored and will not be counted as input for our analysis process. Fields in this table and their explanations are given below :

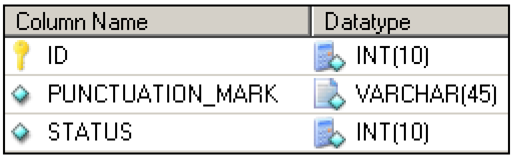


Column Name	Datatype
ID	INT(10)
TWITTER_STOPWORD	VARCHAR(100)
STATUS	INT(10)

Figure 6.36. TWITTER_STOPWORDS table

- ID : Auto-inceremented id, also primary key of the table
- TWITTER_STOPWORD : Keyword to be ignored
- STATUS : 0 (Active) – 1 (Deactive)

<u>6.3.2.11. PUNCTUATION Table.</u> PUNCTUATION table holds punctuation marks that will be omitted during tweet parsing process. Fields in this table and their explanations are given below :



Column Name	Datatype
ID	INT(10)
PUNCTUATION_MARK	VARCHAR(45)
STATUS	INT(10)

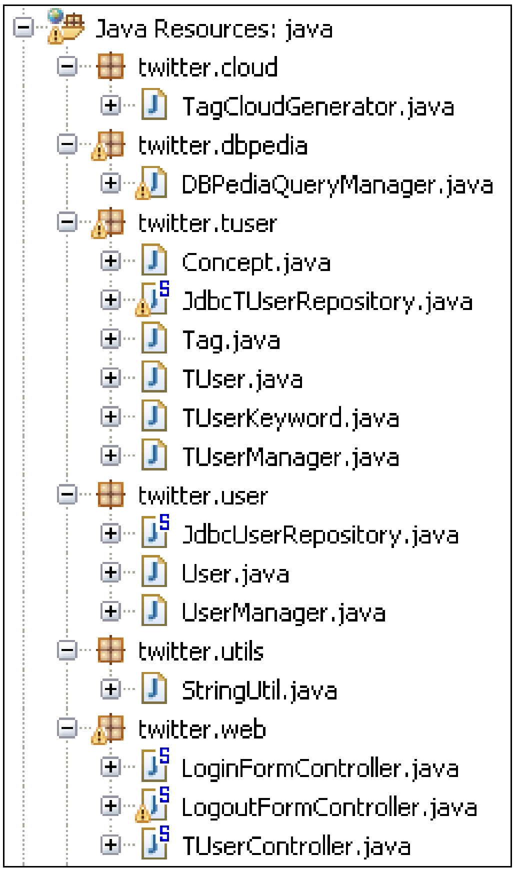
Figure 6.37. PUNCTUATION table

- ID : Auto-inceremented id, also primary key of the table
- PUNCTUATION_MARK : Punctuation to be omitted
- STATUS : 0 (Active) – 1 (Deactive)

6.3.3. SweetTweet Application Structure

SweetTweet has a web based Graphical User Interface (GUI) to provide access from any location using any internet browsers. The application is developed using Spring Framework and Java Server Pages (JSP) [42] and it needs a Servlet [43] container, such as Tomcat [44], to be deployed and run on.

Figure 6.38 shows the Java package and source structure of SweetTweet application.

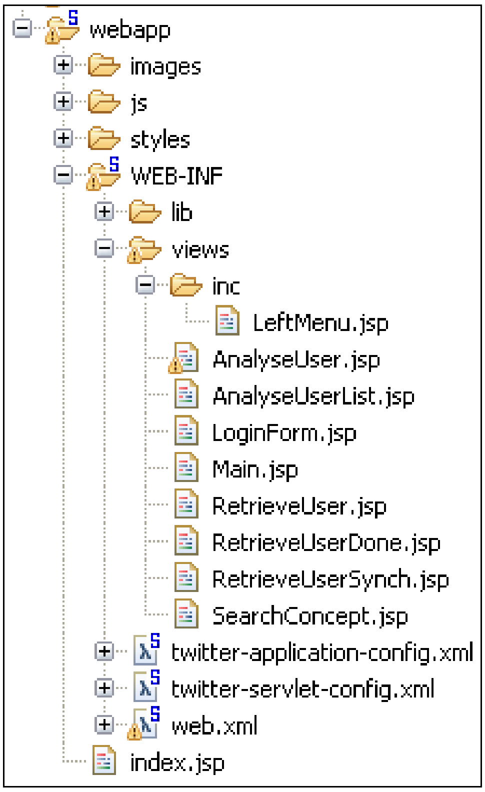


```
Java Resources: java
  twitter.cloud
    TagCloudGenerator.java
  twitter.dbpedia
    DBPediaQueryManager.java
  twitter.tuser
    Concept.java
    JdbcTUserRepository.java
    Tag.java
    TUser.java
    TUserKeyword.java
    TUserManager.java
  twitter.user
    JdbcUserRepository.java
    User.java
    UserManager.java
  twitter.utils
    StringUtil.java
  twitter.web
    LoginFormController.java
    LogoutFormController.java
    TUserController.java
```

Figure 6.38. SweetTweet java package and source structures

- **TagCloudGenerator.java :** Generates tag cloud image for given keyword list and returns the path of the generated image. Creates input files with the keywords and their frequencies, then utilizes IBM’s Word Cloud Generator [45] to generate images using these input files.
- **DBPediaQueryManager.java :** Handles queries sent to DBpedia. Sends queries to DBpedia to find broader subject, subject and label of the categories for a given keyword. It also finds to tag the user.
- **Concept.java :** JavaBean class that holds analysis results for users.
- **JdbcTUserRepository.java :** Handles local database operations for the users retrieved from Twitter. Fetches tweets of the selected user, parses them and finally inserts into database.
- **Tag.java :** JavaBean class that holds tags of users.
- **TUser.java :** JavaBean class that holds information of users retrieved from Twitter.
- **TUserKeyword.java :** JavaBean class that holds query results received from DBpedia for given keyword.
- **TUserManager.java :** Interface that manages access to Twitter user information.
- **JdbcUserRepository.java :** Handles database operations for SweetTweet users.
- **User.java :** JavaBean class that holds information of SweetTweet users.
- **UserManager.java :** Interface that manages access to SweetTweet user information.
- **StringUtil.java :** SweetTweet specific utility class that handles common string operations.
- **LoginFormController.java :** Handles login operations, dispatches requests made through SweetTweet GUI to appropriate JSPs.
- **LogoutFormController.java :** Invalidates user session and redirects to login page.
- **TUserController.java :** Responsible for controlling the operations available in SweetTweet GUI for Twitter users, such as retrieveing user’s tweets, analysing users.
- **LeftMenu.jsp :** Lists available operations in SweetTweet GUI. This menu page is included in other JSPs in SweetTweet web application.
- **AnalyseUser.jsp :** Shows the information of the user that is held in SweetTweet database. SweetTweet users can define a date interval that they want to analyse. This page lists the keywords with respect to their frequencies in user’s tweets. Retieves

analysis results queried from DBpedia for tagging user. Shows tag cloud generated for frequently used keywords.



```
webapp
    images
    js
    styles
    WEB-INF
        lib
        views
            inc
                LeftMenu.jsp
            AnalyseUser.jsp
            AnalyseUserList.jsp
            LoginForm.jsp
            Main.jsp
            RetrieveUser.jsp
            RetrieveUserDone.jsp
            RetrieveUserSynch.jsp
            SearchConcept.jsp
        twitter-application-config.xml
        twitter-servlet-config.xml
        web.xml
    index.jsp
```

Figure 6.39. SweetTweet web application structure

- **AnalyseUserList.jsp :** Lists users retrieved from Twitter and provides analyse link for each user.
- **LoginForm.jsp :** Login screen of SweetTweet GUI. If login fails, this page also shows the error.
- **Main.jsp :** Main page after a successful login operation.
- **RetrieveUser.jsp :** Lists current users retrieved from Twitter and provides a synchronization link to keep the data of the user up-to-date in SweetTweet. New users from Twitter can be also retrieved by using this page.
- **RetrieveUserDone.jsp :** After a successful synchronization process, this page is displayed. It provides a direct link to analyse the retrieved user. Another link to RetrieveUser.jsp is provided to add new users to SweetTweet.

- **RetrieveUserSynch.jsp :** Shows the information of the user stored in SweetTweet and available in Twitter side by side. Provides a "Synchronize" button that updates the information kept in SweetTweet database.
- **SearchConcept.jsp :** Provides a keyword based search mechanism to filter, find and list users according to their interests.
- **index.jsp :** Welcome file of SweetTweet GUI. It has a form that provides login operation.

7. CONCLUSIONS AND FUTURE WORK

This section summarizes the outputs of our model and discusses whether we have reached our goal in this thesis study, or not. Section 7.1 provides information about the conclusions of our model and the contributions we made by proposing this model. In section 7.2, we explore the possibilities to better our methodology we proposed in this thesis study and discuss how this thesis study helps us as a starting point for our future research about the topic.

7.1. Conclusions

Microblogs are rapid, dynamic and continuously evolving environments. The data availabe on microblogs are increasing, as the number of microblog users increases. In this thesis work, we aimed to analyze the contributions of users made to microblogs, to categorize and understand user posts. The methodology we proposed also enables the consumption of the data we analyzed.

The limitations in microblogs, such as limited content size, caused some difficulties in performing analysis process on the data. The only information we had for performing an analysis were the words in user posts. With such a limited data in our hands, performing a keyword based prediction about users' contributions to microblogs would not produce correct results. We needed an extra prediction mechanism that helps us to introduce more reliable results about users in microblogs. That's why we utilized semantic web resources in our model. DBpedia is selected as semantic web resource to provide better prediction and understanding of the words in the users' posts. These words are queried from DBpedia in order to find the categories defined for words. However, the results of these queries may also be misleading, because of the ambiguity of categories we searched and found. For example, when we query "Java" keyword from DBpedia, we find that this keyword is defined in "Java" and "Island of Indonesia" categories. To decide which one is correct and to remove this ambiguity between categories, we needed to search for matching categories between the words in users' posts.

During this study, we performed analysis process for lots of Twitter users. These users are selected randomly or chosen from the “suggested users to follow” lists available on twitterholic.com and wefollow.com. New users also can be added to our application to execute an analysis over their contributions to microblogs. Only by providing a valid user id on Twitter as an input to our application, analysis process we proposed can be performed and we can deduce what users are talking about in Twitter.

The analysis results we deduce about the contributions of the users on microblogs show different characteristics with respect to the users’ purpose of usage of Twitter. For some type of users, who do not post about specific subjects but send updates about anything, including their lives, current situations and feelings etc, our analysis results may not provide desired information, as expected. But, there are also users in Twitter who utilize Twitter for sharing information about specific interest areas. For this type of users, the results of our analysis seem to be much more reliable and useful.

The results we get from the comparison of manual categorization (categorization made by 30 individuals) and automatic categorization (categorization made by SweetTweet) of 30 Twitter users also supports the comment we made above. As seen on Table D.1 on Appendix D, for some users the categories found by SweetTweet application are mostly matched (exact, subsume or related) to the manual categories found by 30 individuals. When we check the tweets of these type of users, we can easily see that this type of users mostly posting structured and predictable contributions to Twitter. For some type of users that use Twitter for chit-chatting, unrelated categories count is the highest value, as seen on Table D.1.

The overall result we have from the comparison of the categories of 30 users shows that the categories found by our model relates to the manual categories at a ratio of 58%. The percentage of unrelated categories produced by our model is 43%. The number of unrelated categories generated by our model may seem high, but considering the characteristics of microblogs, such as having limited post size, or using many socialization words, we think that this value is reasonable.

7.2. Future Work

The model proposed in this thesis study presents useful and interesting analysis results performed by analyzing user contributions made to microblogs. However, there are some potential work of development which will improve our model. We plan to enable the following functionalities in our model, so that the results we have will be more accurate, reliable and useful.

As we stated on previous chapters of this thesis document, we use semantic web resources to understand and categorize users' contributions in microblogs. In our model, we currently use DBpedia as a resource to perform analysis operations. A future work for our model may be the addition of other semantic web resources along with DBpedia. Performing an analysis operation by querying multiple semantic web resources may provide better and more reliable results. Furthermore, cross-checking the results we have from different semantic web resources may reduce the erroneous categories we inferred.

Hashtags and links can be used in our analysis process. Currently we don't use hashtags or links while performing an analysis. Hashtags are the tags created by Twitter community, that consist of words or phrases and starts with a prefix '#'. For example, '#nowplaying' is a popular hashtag defined on Twitter. Users can search '#nowplaying' hashtag and find the tweets about who is listening which song right now. In our analysis, we can classify hashtags by querying them in Twitter. This helps better understanding of users' tweets. Same applies for the links provided in tweets. Since the posts are limited to 140 characters and URLs can be very long, Twitter shortens links in tweets by using some URL shortening services. In fact, links are really valuable sources for understanding user contributions to microblogs. To improve the reliability of our model, we will add metadata about the links provided in tweets to the analysis process, as a future work.

User recommendations will also be an important improvement in our model. On analyse user screen, user recommendations will be displayed. For example; as a result of an analysis process, if a user is classified as he is contributing to the microblog about "Web 2.0", we can also find other users in Twitter who are also interested in the same concept and recommend them in analyse user screen. This will help the users of our model to find and follow different Twitter users that have the same interests.

The language support in our model is currently limited to English. This means that we take the keywords written in English into consideration in our analysis process. As a future work and a possible improvement in our model is the multiple language support.

Finally, collecting user feedback is an another important aspect on evaluating the reliability of the analysis results produced by our model. When a user runs an analysis for a microblogger, the results of the analysis are listed on the screen. In this screen, the application can ask whether the analysis results shown are reliable and helpful, or not. Especially, negative feedbacks on analysis results may force us to better our queries sent to semantic we resources. The feedback information, in general, helps us to find a focus for improvements on our model.

APPENDIX A: SWEETTWEET CATEGORIES

To evaluate the results of our model, we generate the category list for 30 Twitter users by using SweetTweet application. The value between paranthesis on each cell is the calculated weight value for the category. Table A.1 on the next page lists the categories generated by SweetTweet application for 30 Twitter users.

Table A.1. SweetTweet categories

USERNAME	**SWEETTWEET CATEGORIES**									
adamcroft	Members of the UK Parliament (0,42)	Labour Party UK (0,32)	British Female MPs (0,31)	Politicians from Liverpool (0,31)	UK MPs 2005 (0,31)	History of Ireland 1801 – 1922 (0,31)	Liberal Parties (0,31)	Liberal-Labour Politicians UK (0,24)	LGBT wings of political parties (0,24)	Childhood (0,23)
algore	Economic Problems (0,42)	Energy Development (0,38)	Energy in the US (0,38)	Environmental Economics (0,38)	Ecology (0,38)	Climate Change Organizations (0,38)	Renewable Energy Economy (0,38)	Energy Policy (0,38)	International environmental organizations (0, 38)	Climate Change (0, 38)
aplusk	Video games sequels (0,26)	Films set in NewYork City (0,15)	Films shot anamorphically (0,15)	Films directed by actors (0,15)	The Beatles Songs (0,10)	English Film Actors (0,10)	World War II First Person Shooters (0,10)	Prometheus Award Winners (0,10)	CBS Network Shows (0,10)	Love (0,10)
CaliLewis	Itunes (0,27)	Real-Time Web (0,23)	Web 2.0 (0,23)	Iphone OS (0,18)	Multi-Touch (0,18)	Wi-fi devices (0,18)	Touch Screen Portable Media Players (0,18)	Puzzle Video Games (0,17)	Cloud Clients (0,17)	Internet Memes (0,17)
CharlieMars	Euphemisms (0,43)	Number-1 Singles in Switzerland (0,39)	Video Game Culture (0,36)	Video game gameplay (0,36)	Software Comparisons (0,36)	Video Game Magazines (0,36)	Album Types (0,35)	Non-Profit Organizations based in the US (0,35)	Political Terms (0,35)	Phrases (0,35)
ChrisPirillo	Itunes (0,40)	Multi-Touch (0,40)	Wi-Fi Devices (0,40)	Touch Screen Portable Media Players (0,40)	Iphone OS (0,40)	Web 2.0 (0,32)	Computer Hardware (0,31)	Windows Software (0,30)	2010 Introductions (0,29)	Tablet Computer (0,29)
chrisspooner	Windows Software (0,52)	Films and video Technology (0,45)	Graphic Design (0,44)	Computer File Formats (0,43)	Technical Communication (0,43)	International Conferences (0,42)	Marketing (0,42)	Vector graphics editors (0,40)	Learning (0,35)	Cascade Style Sheets (0,35)
CynthiaWare	Web 2.0 (0,38)	Real-Time Web (0,20)	Twitter (0,20)	Text messaging (0,20)	Internet Culture (0,18)	Culture Jamming (0,15)	Internet Terminology (0,12)	Academic Publishing (0,12)	Social Information Processing (0,12)	International Non-Profit Organizations (0,12)
dougfox	MTV Networks (0,63)	American Record Labels (0,60)	Postmodern Art (0,60)	Historical Novels (0,59)	Internet Memes (0,58)	Music Software (0,57)	Lists of People by Occupation (0,57)	Video game franchises (0,57)	Modernism (0,55)	Etiquette (0,55)

Table A.1. SweetTweet categories (continued)

USERNAME	**SWEETTWEET CATEGORIES**									
ebertchicago	Entertainment Rating Organizations (0,41)	Internet Forums (0,40)	Web 2.0 (0,38)	United States Supreme Court Cases (0,35)	Online Movie Databases (0,29)	Film Related Lists (0,29)	Amazon.com (0,29)	Film Review Web Sites (0,29)	TV in the Philippines (0,29)	African-American Film (0,29)
eda49	Web 2.0 (0,35)	Real-Time Web (0,27)	Twitter (0,18)	Lists of Software Extensions (0,18)	Text messaging (0,18)	Marketing (0,17)	Educational Organizations (0,10)	Educational software (0,08)	Internet Hoaxes (0,08)	Software (0,08)
EelcoVisser	Software Architecture (0,56)	Technical Communication (0,56)	Computer File Formats (0,56)	Free cross-platform software (0,55)	Google (0,55)	Cross-platform software (0,55)	Java Platform (0,45)	Free software programmed in C (0,45)	Public Domain Software (0,45)	Python Programming Language (0,45)
EVKwine	American Novels (0,37)	Films based on novels (0,35)	Cross-platform Software (0,34)	Windows Software (0,34)	Companies established in 1998 (0,34)	Linux Software (0,34)	Mac OSX Software (0,34)	Historical Foods (0,29)	Free Software Programmed in C (0,29)	Wine Regions of Germany (0,29)
hrheingold	International non-profit organizations (0,48)	Education in the UK (0,40)	Internet properties established in 2005 (0,38)	Non-governmental organizations (0,33)	Online Social Networking (0,33)	Columbia University (0,32)	International Organizations (0,32)	Disabilities (0,32)	Charities (0,32)	Education in the US (0,32)
ilawton	Science fiction novels (0,56)	Films based on novels (0,50)	Video game sequels (0,42)	CBS Network Shows (0,41)	Population (0,41)	Demography (0,41)	ABC network shows (0,41)	Video games with expansion packs (0,40)	Best picture Academy Award winners (0,38)	Human Rights (0,38)
imhassan	Itunes (0,53)	Smartphones (0,47)	Wi-Fi Devices (0,44)	Multi-Touch (0,44)	Internet Memes (0,42)	Touch Screen Portable Media Players (0,36)	Iphone OS (0,36)	YouTube Videos (0,34)	Viral Videos (0,34)	Computer File Formats (0,32)
kevinrose	Internet Memes (0,25)	Video games with expansion Packs (0,19)	Video game sequels (0,19)	Audio podcasts (0,17)	Companies based on San Francisco, California (0,16)	Twitter (0,16)	Text Messaging (0,16)	Lists of Software Extensions (0,16)	Real-time web (0,16)	Web 2.0 (0,16)
louisgray	Web 2.0 (0,65)	Blog hosting services (0,60)	Technology in Society (0,57)	Social Information Processing (0,56)	Online Social Networking (0,52)	Internet Advertising and Promotion (0,50)	Political Weblogs (0,50)	Internet Terminology (0,48)	Search Engine Optimization (0,48)	American Blogs (0,46)

Table A.1. SweetTweet categories (continued)

USERNAME	SWEETTWEET CATEGORIES									
mandy_gryffin	Pop Ballads (0,36)	Film Soundtracks (0,36)	Films based on novels (0,38)	American novels (0,38)	American Indie Rock Groups (0,38)	Video game developers (0,32)	Learning (0,32)	American poetry collections (0,30)	Semantics (0,30)	English Phrases (0,30)
mashable	Web 2.0 (0,43)	Itunes (0,37)	Internet Memes (0,37)	Internet Protocols (0,36)	Video on demand services (0,36)	Cross-platform software (0,36)	Contemporary Art (0,36)	Video games with expansion packs (0,32)	Social Information Processing (0,31)	Online Social Networking (0,29)
mattcherniss	American Novels (0,47)	List of animated television series episodes (0,39)	Video game developers (0,34)	Baseball terminology (0,34)	ABC Network Shows (0,34)	CBS Network Shows (0,33)	Women's NBA (0,33)	NBC Network Shows (0,33)	1998 television series debuts (0,33)	Video games with expansion packs (0,32)
mrdannyglover	Non-profit organizations based in the US (0,35)	Community Organizing (0,27)	UN Security Council Mandates (0,27)	Anticipatory Thinking (0,27)	Political Organizations (0,27)	Communication (0,27)	Political Corruption (0,27)	Political Slogans (0,27)	Irregular Military (0,27)	Independence Referendum (0,27)
philbowdle	Internet Memes (0,47)	Christianity in Philippines (0,38)	Types of churches (0,27)	September 11 attacks (0,27)	Anti-communism (0,27)	Holy week (0,27)	American Painters (0,27)	CIA Operations (0,27)	Landmarks in Germany (0,27)	1850 Architecture (0,27)
questlove	Slang (0,34)	Association Football Defenders (0,33)	Canadian Business People (0,28)	English Male Singers (0,27)	The Cure Members (0,27)	English Jews (0,27)	British Jazz musicians (0,27)	Internet Memes (0,27)	Internet Slang (0,27)	Texting Codes (0,27)
RobertBluey	Books about Barack Obama (0,07)	Presidency of Barack Obama (0,07)								
ryanvooris	NBC Network Shows (0,51)	American Novels (0,51)	Independent record labels (0,44)	American game shows (0,40)	Video game franchises (0,40)	Video game gameplay (0,36)	Association Football Terminology (0,35)	Game Theory (0,34)	American Record Labels (0,33)	Australian Rules Football (0,2)
ScottBourne	Monthly Magazines (0,16)	Digital Art (0,12)	Independent record labels (0,12)	Photography Organizations (0,11)	Digital Photography (0,11)	Graphics File Formats (0,11)	Photography by Genre (0,11)	Photography Magazines (0,11)	Photography Techniques (0,11)	Internet Forums (0,10)
Steveology	Internet Terminology (0,67)	Search Engine Optimization (0,67)	Web 2.0 (0,67)	Social Information Processing (0,63)	Communication (0,62)	Privately held companies of th US (0,61)	Social Media (0,61)	Internet Marketing (0,61)	Internet Culture (0,58)	Social Groups (0,57)

Table A.1. SweetTweet categories (continued)

USERNAME	**SWEETTWEET CATEGORIES**									
timoreilly	Library and Information Science (0,30)	Network Related Software (0,29)	Film and Video Technology (0,29)	Windows Software (0,29)	Cross-platform Software (0,29)	Google Services (0,28)	Internet Marketing (0,22)	News websites (0,22)	Semantic Web (0,21)	World Wide Web (0,21)
uskudarli	Web 2.0 (0,46)	Online Social Networking (0,43)	Semantic Web (0,34)	Internet Marketing (0,34)	Software Companies of the US (0,34)	Marketing (0,32)	Search Engine Optimization (0,30)	Theories of History (0,26)	Social Classes (0,26)	Social Groups (0,26)

APPENDIX B: USER CONTRIBUTED CATEGORIES

To evaluate the output of our model, we asked 30 people to manually categorize 30 Twitter users. We provided some example categories for these users, just to give an idea for categorization process. Either these categories can be selected or new categories can be added to evaluate the users. According to the responses that we received from these 30 people about 30 Twitter users, we generated a table that summarizes the categorization results. Please refer to Table B.1 on the next page to see user contributed categories defined for these Twitter users.

Table B.1. User contributed categories

USER NAME	**CATEGORIES**										
adamcroft	Chit-Chat (0,22)	Daily Life (0,21)	Politics (0,21)	Technology (0,11)	Social Media (0,10)	News (0,08)	Movies (0,05)	Computer (0,01)	Video Games (0,01)		
algore	Politics (0,36)	News (0,30)	Environment (0,20)	Blogger (0,07)	Daily Life (0,04)	Technology (0,02)	Sports (0,02)				
aplusk	Daily Life (0,37)	Chit-Chat (0,34)	Music (0,08)	Politics (0,07)	News (0,05)	Entertainment (0,05)	Social Media (0,03)				
CaliLewis	Technology (0,46)	Computer (0,14)	Internet (0,11)	Social Media (0,09)	Mobile Devices (0,05)	News (0,05)	Blogger (0,03)	Chit-Chat (0,03)	Web (0,02)	Movies (0,02)	Daily Life (0,02)
CharlieMars	Daily Life (0,45)	Chit-Chat (0,39)	News (0,07)	Movies (0,07)	Music (0,02)						
ChrisPirillo	Technology (0,35)	Social Media (0,13)	Web (0,12)	Internet (0,12)	Computer (0,08)	Chit-Chat (0,06)	Daily Life (0,05)	Literature (0,04)	News (0,04)	Quotations (0,02)	
chrisspooner	Graphic Design (0,30)	Web (0,25)	Chit-Chat (0,12)	Internet (0,10)	Daily Life (0,07)	Blogger (0,06)	Photography (0,06)	Technology (0,03)			
CynthiaWare	Daily Life (0,33)	Chit-Chat (0,28)	Social Media (0,18)	Technology (0,15)	Entrepreneur (0,02)	Blogger (0,02)	Internet (0,02)				
dougfox	Social Media (0,31)	Dance (0,28)	Internet (0,12)	Web (0,10)	Photography (0,10)	Chit-Chat (0,05)	Entrepreneur (0,02)	Education (0,02)			
ebertchicago	Chit-Chat (0,26)	Movies (0,22)	Daily Life (0,19)	Technology (0,07)	Quotations (0,07)	Blogger (0,03)	Politics (0,03)	News (0,03)	Entrepreneur (0,03)	Music (0,02)	Video Games (0,02)
eda49	Web (0,20)	Technology (0,17)	Daily Life (0,12)	Internet (0,11)	Chit-Chat (0,11)	Social Media (0,07)	Computer (0,07)	Graphic Design (0,07)	Blogger (0,04)	Politics (0,04)	Software (0,01)
EelcoVisser	Computer (0,26)	Technology (0,20)	Software (0,16)	Web (0,16)	Internet (0,10)	Daily Life (0,06)	Social Media (0,03)	Blogger (0,03)			
EVKwine	Food/Drinks (0,33)	Social Media (0,21)	Daily Life (0,21)	Technology (0,05)	Quotations (0,05)	Entrepreneur (0,05)	Chit-Chat (0,04)	Business (0,02)	Blogger (0,02)	Internet (0,02)	
hrheingold	Social Media (0,22)	Web (0,15)	Education (0,10)	Chit-Chat (0,09)	Technology (0,08)	News (0,07)	Politics (0,07)	Daily Life (0,07)	Blogger (0,03)	Computer (0,03)	Internet (0,03)
ilawton	Daily Life (0,40)	Chit-Chat (0,26)	Religion (0,15)	Blogger (0,08)	Quotations (0,08)	Environment (0,03)					
imhassan	Technology (0,31)	Internet (0,25)	Web (0,15)	Social Media (0,06)	Blogger (0,06)	Daily Life (0,06)	Computer (0,04)	Software (0,04)	Chit-Chat (0,04)		

Table B.1. User contributed categories (continued)

USER NAME	CATEGORIES										
kevinrose	Technology (0,28)	Daily Life (0,19)	Chit-Chat (0,18)	Social Media (0,09)	Blogger (0,06)	Internet (0,06)	Charity (0,04)	Politics (0,04)	Web (0,02)	Video Games (0,02)	Computer (0,01)
louisgray	Technology (0,23)	Blogger (0,17)	Internet (0,17)	Social Media (0,12)	Computer (0,10)	Web (0,10)	Chit-Chat (0,03)	Entrepreneur (0,03)	Daily Life (0,03)		
mandy_gryffin	Chit-Chat (0,41)	Daily Life (0,41)	Movies (0,16)	Education (0,02)							
mashable	Technology (0,27)	Social Media (0,19)	Internet (0,15)	Computer (0,12)	Web (0,12)	Blogger (0,08)	Movies (0,04)	News (0,04)	Entrepreneur (0,01)		
mattcherniss	Sports (0,36)	Movies (0,25)	Daily Life (0,24)	Chit-Chat (0,05)	Computer (0,03)	Entertainment (0,03)	Comics (0,03)				
mrdannyglover	Politics (0,37)	Movies (0,37)	Charity (0,08)	Music (0,08)	Environment (0,06)	News (0,04)					
philbowdle	Technology (0,30)	Daily Life (0,26)	Chit-Chat (0,22)	Religion (0,08)	Computer (0,04)	Social Media (0,03)	Blogger (0,03)	Music (0,01)	Web (0,01)	Internet (0,01)	
questlove	Music (0,41)	Chit-Chat (0,26)	Quotations (0,13)	Social Media (0,13)	Daily Life (0,07)						
RobertBluey	Politics (0,31)	Daily Life (0,16)	News (0,13)	Entrepreneur (0,13)	Chit-Chat (0,08)	Business (0,07)	Social Media (0,05)	Computer (0,03)	Blogger (0,03)		
ryanvooris	Daily Life (0,33)	Sports (0,28)	Chit-Chat (0,28)	Social Media (0,09)	Internet (0,02)						
ScottBourne	Photography (0,31)	Technology (0,22)	Computer (0,12)	Chit-Chat (0,12)	Social Media (0,09)	Web (0,08)	Blogger (0,03)	Entrepreneur (0,03)			
Steveology	Social Media (0,47)	Entrepreneur (0,13)	Business (0,13)	Chit-Chat (0,13)	Blogger (0,06)	Internet (0,05)	Web (0,05)				
timoreilly	Technology (0,29)	Computer (0,20)	Web (0,12)	Internet (0,10)	Entrepreneur (0,09)	Social Media (0,09)	Daily Life (0,05)	Blogger (0,04)	Health (0,02)		
uskudarli	Social Media (0,30)	Web (0,21)	Daily Life (0,17)	Internet (0,11)	Technology (0,06)	Computer (0,06)	Education (0,06)	Chit-Chat (0,01)			

APPENDIX C: RELATED CATEGORIES

Table C.1. Related categories

User Contributed Category	SweetTweet Category
Blogger	American Blogs
	Blog hosting services
	Political Weblogs
Charity	Charities
	International non-profit organizations
	Non-governmental organizations
	Non-profit organizations based in the US
Chit-Chat	Internet Slang
	Slang
	Texting Codes
Computer	Computer File Formats
	Computer Hardware
	Library and Information Science
	Tablet Computer
Daily Life	Childhood
	Love
Graphic Design	Cascade Style Sheets
	Graphic Design
	Graphics File Formats
	Vector graphics editors
Business	Canadian Business People
	Community Organizing
	Companies based on San Francisco, California
	Companies established in 1998
	Disabilities
	Economic Problems
	International Organizations
	Internet Marketing
	Marketing
	Privately held companies of th US
Education	Academic Publishing
	Columbia University
	Education in the UK
	Education in the US
	Educational Organizations
	Educational software
	Learning

Table C.1. Related categories (continued)

User Contributed Category	SweetTweet Category
Environment	Climate Change
	Climate Change Organizations
	Ecology
	Energy Development
	Energy in the US
	Energy Policy
	Environmental Economics
	International environmental organizations
	Renewable Energy Economy
Food/Drinks	Historical Foods
	Wine Regions of Germany
Internet	Cloud Clients
	Google
	Google Services
	Internet Advertising and Promotion
	Internet Culture
	Internet Forums
	Internet Hoaxes
	Internet Marketing
	Internet Memes
	Internet properties established in 2005
	Internet Protocols
	Internet Terminology
	Search Engine Optimization
Literature	American Novels
	American poetry collections
	Historical Novels
	Science fiction novels
Movies	African-American Film
	Best picture Academy Award winners
	English Film Actors
	Film and Video Technology
	Film Related Lists
	Film Review Web Sites
	Film Soundtracks
	Films based on novels
	Films directed by actors
	Films set in NewYork City
	Films shot anamorphically
	List of animated television series episodes
	Online Movie Databases

Table C.1. Related categories (continued)

User Contributed Category	SweetTweet Category
Music	American Indie Rock Groups
	American Record Labels
	British Jazz musicians
	English Male Singers
	Independent record labels
	Music Software
	Number-1 Singles in Switzerland
	Pop Ballads
	The Beatles Songs
	The Cure members
News	News websites
Photography/Art	American Painters
	Contemporary Art
	Digital Art
	Digital Photography
	Modernism
	Photography by Genre
	Photography Magazines
	Photography Organizations
	Photography Techniques
	Postmodern Art
Politics	Anti-communism
	Books about Barack Obama
	British Female MPs
	CIA Operations
	Culture Jamming
	Demography
	Human Rights
	Independence Referendum
	Labour Party UK
	LGBT wings of political parties
	Liberal Parties
	Liberal-Labour Politicians UK
	Members of the UK Parliament
	Political Corruption
	Political Organizations
	Political Slogans
	Political Terms
	Politicians from Liverpool
	Population
	Presidency of Barack Obama
	September 11 attacks
	UK MPs 2005

Table C.1. Related categories (continued)

User Contributed Category	SweetTweet Category
Politics	UN Security Council Mandates
	United States Supreme Court Cases
Quotations	Phrases
Religion	Christianity in Philippines
	Holy week
	Types of churches
Social Media	Online Social Networking
	Social Classes
	Social Groups
	Social Information Processing
	Social Media
	Twitter
	Web 2.0
Software	Cross-platform Software
	Free cross-platform software
	Free software programmed in C
	Iphone OS
	Java Platform
	Linux Software
	Lists of Software Extensions
	Mac OSX Software
	Network Related Software
	Public Domain Software
	Python Programming Language
	Software
	Software Architecture
	Software Companies of the US
	Software Comparisons
	Windows Software
Sports	Association Football Defenders
	Association Football Terminology
	Australian Rules Football
	Baseball terminology
	Women's NBA
Technology	ITunes
	Multi-Touch
	Smartphones
	Technical Communication
	Technology in Society
	Touch Screen Portable Media Players
	Wi-Fi Devices
Video Games	Puzzle Video Games
	Video Game Culture

Table C.1. Related categories (continued)

User Contributed Category	SweetTweet Category
Video Games	Video game developers
	Video game franchises
	Video game gameplay
	Video Game Magazines
	Video game sequels
	Video games with expansion packs
	Word War II First Person Shooters
Web	Amazon.com
	Real-Time Web
	Semantic Web
	Semantics
	Video on demand services
	Viral Videos
	World Wide Web
	YouTube Videos

APPENDIX D: EVALUATION RESULTS

Table D.1 shows the comparison results of SweetTweet category list and user contributed category list.

Table D.1. Evaluation results

Username	Exact	Subsume	Related	Unrelated
adamcroft	0	0.3	0.6	0.1
algore	0	0.2	0.7	0.1
aplusk	0	0	0.2	0.8
CaliLewis	0	0.3	0.5	0.2
CharlieMars	0	0	0.1	0.9
ChrisPirillo	0	0.3	0.4	0.3
chrisspooner	0.1	0	0.3	0.6
CynthiaWare	0	0.3	0.2	0.5
dougfox	0	0.1	0.2	0.7
ebertchicago	0	0.1	0.4	0.5
eda49	0.1	0.4	0.1	0.4
EelcoVisser	0	0.6	0.4	0
EVKwine	0	0.2	0.1	0.7
hrheingold	0	0.4	0.1	0.5
ilawton	0	0	0	1
imhassan	0	0.2	0.8	0
kevinrose	0	0.4	0.2	0.4
louisgray	0	0.7	0.3	0
mandy_griffin	0	0	0.3	0.7

Table D.1. Evaluation results (continued)

Username	Exact	Subsume	Related	Unrelated
mashable	0	0.4	0.2	0.4
mattcherniss	0	0.2	0	0.8
mrdannyglover	0	0.3	0.3	0.4
philbowdle	0	0.1	0.3	0.6
questlove	0	0.1	0.4	0.5
RobertBluey	0	0	1	0
ryanvooris	0	0	0.2	0.8
ScottBourne	0	0.6	0	0.4
Steveology	0.1	0.5	0.3	0.1
timoreilly	0	0.5	0.2	0.3
uskudarli	0	0.5	0.2	0.3
AVERAGE (%)	**1%**	**26%**	**31%**	**43%**

REFERENCES

1. O'Reilly, T., "What Is Web 2.0", http://oreilly.com/web2/archive/what-is-web-20.html, 2009.

2. Java, "Java Programming Language", www.java.com, 2010.

3. W3 Semantic Web, "W3 Semantic Web Activity", http://www.w3.org/2001/sw/, 2010.

4. RDF, "RDF - Semantic Web Standards", http://www.w3.org/RDF/, 2009.

5. LinkedData, "Connect Distributed Data across the Web", http://linkeddata.org/, 2009.

6. Twitter, "A social networking and microblogging service", http://twitter.com/, 2010.

7. Spivack, N., "Radar Networks", http://www.radarnetworks.com/, 2007.

8. IBM DeveloperWorks, "IBM DeveloperWorks Interviews: Tim Berners-Lee", http://www.ibm.com/developerworks/podcast/dwi/cm-int082206txt.html/, 2009.

9. RailsOnWave.com, "Ruby on Rails, Web 2.0, Ajax tutorials", http://www. railsonwave.it/railsonwave/2007/1/2/web-2-0-map/, 2008.

10. Berners-Lee, T. and M. Fischetti, *Chapter 12, Weaving the Web*, Harper SanFrancisco, ISBN 9780062515872, 1999.

11. W3 XML, "Extensible Markup Language", http://www.w3.org/XML/**,** 2009.

12. W3 OWL, "OWL Web Ontology Language Overview", http://www.w3.org/TR/owl-features/, 2009.

13. SPARQL, "SPARQL Query Language for RDF", http://www.w3.org/TR/rdf-sparql-query/, 2009.

14. SemanticWeb.org, "Semantic Web Stack", http://semanticweb.org/images/3/37/ Semantic-web-stack.png/, 2010.

15. DBpedia, "A community effort to extract structured information from Wikipedia", http://dbpedia.org/, 2009.

16. The Open Archives Initiative, "ORE User Guide Primer", http://www.openarchives. org/ore/1.0/primer/, 2010.

17. W3 RDF/XML, "RDF/XML Syntax Specification(Revised)", http://www.w3.org/TR /REC-rdf-syntax/, 2009.

18. W3 Notation3, "Notation3 (N3): A readable RDF Syntax", http://www.w3.org/ Design Issues/Notation3/, 2009.

19. W3 RDF Intro, "Introduction to RDF Metadata", http://www.w3.org/TR/NOTE-rdf-simple-intro/, 2009.

20. W3 Linked Data, "Linked Data Design Issues", http://www.w3.org/DesignIssues/ LinkedData.html/, 2009.

21. Tumblr, "A blogging platform", http://www.tumblr.com/, 2009.

22. Jaiku, "A social networking, micro-blogging and lifestreaming service", http://www. jaiku.com/, 2009.

23. Identi.ca, "A micro-blogging service", http://identi.ca/, 2009.

24. Web 2.0 Summit 2009: Evan Williams and John Battelle, *A Conversation with Evan Williams*, O'Reilly Media, October 21, 2009.

25. Twitter, “Twitter Search”, http://search.twitter.com/, 2009.

26. Twitter Blog, “Twitter Suggested Users List”, http://blog.twitter.com/2009/03/suggested-users.html/, 2010.

27. Wefollow, “Twitter Directory and Search”, http://wefollow.com/, 2010.

28. Twitterholic, “Top Twitter User Rankings & Stats”, http://twitterholic.com/, 2010.

29. Hotho, A., R. Jaschke, C. Schmitz and G. Stumme, “Information retrieval in folksonomies: search and ranking”, *Proceedings of the 3rd European Semantic Web Conference*, http://www.kde.cs.uni-kassel.de/stumme/papers/2006/hotho2006information.pdf, 2006.

30. Gruber, T., “Collective knowledge systems: Where the Social Web meets the Semantic Web”, *Web Semant.* 6, 1 (Feb. 2008), 4-13, 2008.

31. Kim, H. L., S. Scerri, J. G. Breslin, S. Decker and H. G. Kim, “The state of the art in tag ontologies: a semantic model for tagging and folksonomies”, *Proceedings of the 2008 international Conference on Dublin Core and Metadata Applications* (Berlin, Germany, September 22 - 28, 2008), International Conference on Dublin Core and Metadata Applications. Dublin Core Metadata Initiative, 128-137, 2008.

32. Halpin, H., V. Robu, and H. Shepard, “The Dynamics and Semantics of Collaborative Tagging”, *Proceedings of the 1st Semantic Authoring and Annotation Workshop (SAAW06)*, 2006.

33. Bojrs, U., J. G. Breslin, A. Finn, and S. Decker, “Using the Semantic Web for linking and reusing data across Web 2.0 communities”, *Web Semant.* 6, 1 (Feb. 2008), 21-28, 2008.

34. SIOC, “Semantically-Interlinked Online Communities”, http://sioc-project.org/, 2010.

35. Mika, P., “Ontologies are us: A unified model of social networks and semantics”, *Web Semant.* 5, 1 (Mar. 2007), 5-15, 2007.

36. Mika, P., “Social Networks and the Semantic Web”, *Proceedings of the 2004 IEEE/WIC/ACM international Conference on Web intelligence* (September 20 - 24, 2004). Web Intelligence. IEEE Computer Society, Washington, DC, 285-291, 2004.

37. RT@google : Tweets and update and search, oh my!, http://googleblog.blogspot.com/2009/10/rt-google-tweets-and-updates-and-search.html, 2009.

38. Android, “Android at Google I/O”, http://www.android.com/, 2010.

39. Twitter4j, “A Java Library for the Twitter API”, http://twitter4j.org/, 2009.

40. Spring, “Spring Source Community”, http://www.springsource.org/, 2009.

41. MySQL, “MySQL :: The world’s most popular open source database”, http://www.mysql.com/, 2010.

42. Sun Developer Network, “Java Server Pages”, http://java.sun.com/products/jsp/, 2010.

43. Sun Developer Network, “Java Servlet Technology”, http://java.sun.com/products/servlet/, 2010.

44. Apache Tomcat, “Open Source Servlet Container”, http://tomcat.apache.org/, 2010.

45. IBM alphaWorks, “alphaWorks: Word Cloud Generator Overview”, http://www. alphaworks.ibm.com/tech/wordclo ud/, 2010.