1-1-2016

Improving User Experience In Information Retrieval Using Semantic Web And Other Technologies

Erfan Najmi
Wayne State University,

IMPROVING USER EXPERIENCE IN INFORMATION RETRIEVAL USING SEMANTIC WEB AND OTHER TECHNOLOGIES

by

ERFAN NAJMI

DISSERTATION

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

2016

MAJOR: COMPUTER SCIENCE

Approved By:

_____

Advisor Date

_____

_____

_____

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to my PhD advisor, **Dr. Zaki Malik**, for supporting me during these past years. I could not have asked for a better advisor and a friend, one that let me choose my path, help me along it and has always been there if I needed to talk to a friend. I really appreciate all the time he spent and all the patience he showed towards me. Secondly I would like to thank my committee members Dr. Fengwei Zhang, Dr. Alexander Kotov and Dr. Abdelmounaam Rezgui for the constructive feedback and help they provided.

I thank Dr. Khayyam Hashmi, for all these years I knew I could count on you; when I needed someone to listen and understand, when I needed a friend to have my back, and when I needed a brother to help me stand.

Lastly, this whole journey would not have been possible without the support of my family, in particular my parents. Mom and Dad this one is for you. While thousands of miles away, you have always been with me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: BACKGROUND AND INTRODUCTION

The need to find, access and extract information has been the motivation for many different fields of research in the past few years. The fields such as Machine Learning, Question Answering Systems, Semantic Web, etc. each tries to cover parts of the mentioned problem. Each of these fields have introduced many different tools and approaches which in many cases are multi-disciplinary, covering more than one of these fields to provide solution for one or more of them. On the other hand, the expansion of the Web with Web 2.0, gave researchers many new tools to extend approaches to help users extract and find information faster and easier. Currently, the size of e-commerce and online shopping, the extended use of search engines for different purposes and the amount of collaboration for creating content on the Web provides us with different possibilities and challenges which we address some of them here.

In this work our goal is to implement a set of approaches to create a faster experience for users to access the information they need. We begin with describing our approach for Question Answering using Semantic Web technologies. For this purpose we require large semantic knowledge bases which are in specific formats. While there are some examples of large knowledge bases which are available in SW formats, there are some publicly available knowledge bases which do not have this format. Our next step is to start by creating an upper ontology and then use that ontology to convert a large common sense knowledge base to RDF. We later provide a more detailed description of the conversion of ConceptNet knowledge base to RDF format using the mentioned upper ontology.

Another part of our focus is on product suggestion for customers using product reviews. One can look at this work as a similar approach to Question Answering, but users instead of asking general question are asking about the best product they

can buy based on their criteria. This work itself consists of different parts such as Sentiment Analysis and Review Usefulness Analysis. The result of these different sections is summarized to provide users with an in depth ranking of products, both from a general user perspective and based-on user specific priorities.

When talking about products, we have to mention the importance of microblogging Web sites such as Twitter and the effect they have on product sales and analysis. To analyze the vast amount of information available on these platforms, we create an approach to extract the related terms, and assign weights to them based on how much effect they have on the popularity of the product.

Finally, in a scenario that the user requires information about a concept we see a need to prioritize knowledge presented in different KBes based on general user opinions. The reasoning behind this proposition is that in many cases users do not want to see bulk of information related to a concept and they spend time to find the most interesting or popular fact. So we suggest a solution to provide the most needed piece of knowledge first. While we have many KBes presented in different formats, both in SW and other formats, there is no order in the knowledge. We propose that to convert this knowledge to wisdom we need an addition to this knowledge and discuss this approach in depth.

CHAPTER 2: RELATED WORK

2.1 Semantic Web Overview

World Wide Web (WWW) was created based-on the idea of interlinked web pages, providing human readable information to users. This human readability, while making it easier for users to read the information, is near impossible to be understood by machines. The growth of WWW presented a need to organize the information. First answers to this need was from search engines to create lists of desirable contents for users. While search engines have a long history on WWW, a structured idea of creating a machine readable backbone for Web was created and named Semantic Web (SW) in Berners-Lee et al. [2001].

The vision of SW is to provide a new approach to WWW to introduce intelligence and semantic, machine understandable meaning to it. To do so, there have been multiple technologies introduced over the years to help with this situation. In the following, we go over the main technologies, specifically the ones standardized by W3 consortium. The main goal of this section is not to describe the mentioned technologies completely, but rather have an introduction to give readers a basic familiarity with the terminology which can be seen in the next sections.

2.1.1 RDF

RDF (Resource Description Framework) (Lassila and Swick [1999]) is a model platform to represent information on the Web. Each record in RDF consists of three portions; Subject (Resource), Predicate (Property) and Object (Literal).

The subject identifies the resource the statement describes, the predicate is the property the statement wants to describe and the object is the value of the property. For example for the statement “Ora Lassila is the creator of the resource http://www.w3.org/Home/Lassila” the RDF triple is shown in Table 2.1 , visualized in Figure 2.1 and presented in XML/RDF format in the following:

Subject (Resource)	http://www.w3.org/Home/Lassila
Predicate (Property)	Creator
Object (literal)	“Ora Lassila”

Table 2.1: RDF sample triple

```
<rdf:RDF>
  <rdf:Description about="http://www.w3.org/Home/Lassila">
    <s:Creator>Ora Lassila</s:Creator>
  </rdf:Description>
</rdf:RDF>
```

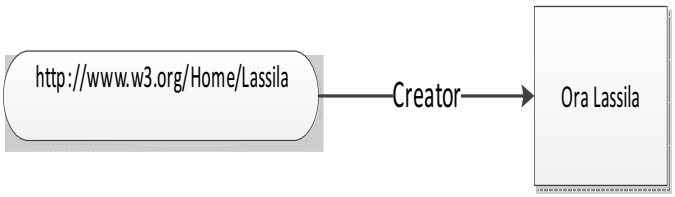

Figure 2.1: Sample RDF graph

2.1.2 Ontologies

Ontologies add semantic meaning to SW. While RDF triples by themselves can be meaningless for humans, the ontologies provide background and meaning to vocabulary used in the triples. Two major technologies to present ontologies in SW are RDFS (RDF Schema)(Brickley and Guha [2004]) and OWL (Web Ontology Language)(Dean et al. [2004]). Both these languages provide tools to describe RDF triples and add meaning to the vocabulary used in them, but the type limit and other constraints RDFS has mostly caused replacing OWL in RDFS place.

We can divide RDFS to three main portions. These portions include classes, dividing each object to a class of objects, such as rdfs:Resource, rdfs:literal; Properties include specifications of objects, e.g. rdfs:range and rdfs:type; and Utility properties which includes rdfs:seeAlso and rdfs:isDefinedBy.

On the other hand, OWL consists of 3 sub-languages of OWL Lite, OWL DL and OWL Full. OWL Lite is mainly created for users needing classification of objects and simple constraints. OWL DL has all the tools provided in OWL but implements Description Logic to limit the use of different constraints. OWL Full is created to have some compatibility to RDFS. Hence, it allows different annotations which under OWL DL and OWL Lite are not allowed. Further information regarding the specification of OWL can be found in Bechhofer et al. [2004].

A comprehensive tutorial and how-tos for both these technologies presented in Allemang and Hendler [2011]. Also there are many ontologies available online for free to use in research and other uses. The two main used are Dublin Core (Weibel et al. [1998]) and FOAF (Friend Of A Friend) (Brickley and Miller [2012]).

2.1.3 SPARQL

SPARQL (SPARQL Protocol And RDF Query Language)(PrudHommeaux et al. [2008]) is a query language over RDF data. SPARQL lets users query required and optional graph patterns along with their conjunctions and disjunctions. SPARQL also allows testing and constraining queries in a manner similar to RDBMS's.

Following code shows a simple SPARQL query which returns country capitals in Africa.

```
PREFIX abc: <nul://sparql/exampleOntology#> .
SELECT ?capital ?country
WHERE {
  ?x abc:cityname ?capital ;
     abc:isCapitalOf ?y.
  ?y abc:countryname ?country ;
     abc:isInContinent abc:Africa.
}
```

2.1.4 Other technologies

Following we go over some of the other technologies in use in SW, shortly describe the technology and the need for each.

- Microformats: The need of SW to create parallel content to add semantic to WWW has made its progress slow. Microformats are technologies to integrate semantic knowledge to general web pages created using current existing and dominant technologies (HTML and XHTML).The language created in accordance to main SW technologies such as RDF is RDFa (Adida and Birbeck [2008]). On the other hand, Web Hypertext Application Technology Working Group (WHATWG) created Microdata as an addition for HTML5 to integrate metadata into HTML web pages. There are other technologies for integrating metadata for specific contents such as hCalender, integrating event information, hCard, contact information.

- Semantic Search Engines: This term can point to two different understanding, first searching RDF or ontology documents and second make the search process semantically smarter. A good example of the first approach is Swoogle [1]. This search engine searches ontologies, documents, terms and data published on the Web. For this process, it employs a system of crawlers to discover RDF documents and HTML documents with embedded RDF content. Swoogle reasons about these documents and their constituent parts (e.g., terms and triples) and records and indexes meaningful metadata about them in its database. For the second approach which is currently more dominant, major search engines on the Web have approached it from different perspectives. The two current major search engines have implemented different approaches to have smoother

[1] http://swoogle.umbc.edu/

search process for natural language searches such as in question answering systems. On the other hand some smaller search engines such as Lexxe [2] which uses natural language processing to extract the information users are looking for. For example by searching keywords "color: camry" instead of searching for the keywords in Web page texts they look for different colors which have been mentioned alongside Toyota Camry. Other examples of semantic search engines include but not limited to *Yummly* [3] for food and recipes and *Kosmix* [4] (currently down).

2.2 Public Access Knowledge Bases

The use of information gathering in the logic based approach, specifically for computers, can be traced to "Advice Taker", a theoretical system introduced in 1963 in the book "Programs with common sense"(McCarthy [1963]). The main goal of this project was to create a platform which led the computers learn and reason from their experiences, similar to humans.

Currently, the oldest active work in information gathering from common sense is Cyc (Lenat et al. [1985]). While *Cyc* converted to commercialized product in 1994, there are free versions specifically created for researchers in *OpenCyc* and *ResearchCyc*. *ResearchCyc* provides access to the full *Cyc* knowledge-base under *ResearchCyc* licence. This version has more than 500000 concepts, nearly 5000000 facts and rules (called assertions in *Cyc* context) and more than 26000 relations. This release enables users to use *CycL*, the language generated for use of *Cyc*, and API to create any required application. Also an ontology exporter is available to export specified portions of the knowledge base to OWL. On the other hand, *OpenCyc*, which is freely available

[2] http://www.lexxe.com/
[3] http://www.yummly.com/
[4] http://archive.is/20120525042010/http://www.kosmix.com/

to everyone, consists of more than 239000 terms and 2000000 concepts which is last updated in 2012.

Another significant work which functions as a bridge between dictionaries, encyclopedias and common sense knowledge is *WordNet*(Fellbaum [1999]). *WordNet*, started in 1985, is currently in version 3.1. Also the 3.0 version is available in RDF which contains more than 155000 words. *WordNet*, other than providing synonyms, antonyms, meronym and holonym, provides different senses for each word, creating a hyper-linked network of words. Considering that *WordNet* does not support different languages, there have been works on a similar platform for European languages under *EuroWordNet*(Vossen [1998]).

A similar work to *WordNet* but comparably newer is *BabelNet*(Navigli and Ponzetto [2010]). *BabelNet* replaces *WordNet's* senses and synsets with Babel synset. BabelNet's latest release as of April 2014 is 2.0.1. There are a few points which makes BabelNet special compare to similar works. First, BabelNet consists of 50 languages (in October 2013). Second, it natively supports Lemon/RDF encodings. And finally it provides 7.7 millions images interlinked with the concepts in the data-set.

ThoughtTreasure(Mueller [1998]) was another work to gather common sense knowledge. The work on this approach started in 1993 and the support stopped in 2000 after the founder moved to IBM research to be part of the group which later developed Watson. Final release of *ThoughtTreasure* has more than 27000 concepts and 51000 assertions in English and French.

A specific case of common sense knowledge base is *YAGO*(Suchanek et al. [2007]) started in 2008. While gathering information from *Wikepedia*, *WordNet* and *Geonames* it has gathered more than 10 millions entities, 120 millions facts and 350000 classes in its current version (*YAGO2*) released in 2012. This knowledge base

has a comprehensive data on IsA taxonomy and schema which has been used as one of data sources for IBM Watson.

A different approach to information presentation is used in *SenticNet*(Cambria et al. [2010]) by providing pleasantness, attention, aptitude and polarity of phrases in [-1.1] range. The latest version of this knowledge-base, version 3.00, is currently in beta. We summarized the contribution and works in some of the knowledge bases specifically in common sense field in Table 2.2.

Approach	Year Founded	Year Stopped	Data	Avail. to download	Notes
MindPixel (Min [2005])	2000	2005	1.4M	no	
Evi	2005	2012	300M	no, app available	Bought by Amazon in 2012. Formerly True Knowlledge
Probase (Wu et al. [2012])			2.7M	no	Belongs to Microsoft, for internal use
ThoughtTreasure(Mueller [1998])	1993	2000	27K concepts, 51K assertions	no	
NELL (Carlson et al. [2010])	2010		50M	yes	Never Ending Language Learning (NELL)
Open Information Extraction (Banko et al. [2007])			15M	yes	From Clue Web 99 data-set
Freebase (Bollacker et al. [2008])	2007		2.5B+ facts, 43M+ topics	yes	RDF available. Google bought in 2010
UMBEL (UMB [2012])	2008	2012	25K (based on their web site)	yes	Subset of OpenCyc
Yago (Suchanek et al. [2007])	2008		10M entities, 120M facts, 350K classes	yes	
WordNet	1985		155K words in V3.00	yes	
BabelNet	2012		9M synsets, 50M word senses	yes	
ResearchCyc	2012		50K concepts, 5M facts	yes	
BabelNet	2012		9M synsets, 50M word senses	yes	
SenticNet	2010			yes	On Sentiment Analysis
OMCS (Singh et al. [2002])	1999		1M+	no	Used in Conceptnet

Table 2.2: Works in common sense knowledge retrieval

2.3 Question Answering Related Work

The first automated QA system goes back to 1961 with a paper from Green et al Green et al. [1961]. But it is just by the recent advancements in different fields of data processing that we have started having a working, open domain QA system. The current works in this field divides and combines different research topics consisting of (but not limited to) Information Retrieval, Statistical (Ittycheriah et al. [2002]), Semantic Web and Artificial Intelligence. In this section, we provide a brief overview of some of the related literature (more in line with our work).

Question Classification is the first step in a QA system. The goal is to classify questions and find the type of the answer. A number of different approaches have been proposed in this regard. For instance in Hovy et al. [2001] questions are categorized to 94 different categories based on patterns found in the user queries. Ravichandran and Hovy [2002] introduces a similar approach where patterns in questions are generated by using related patterns in documents gathered from web. Similarly, machine learning is used in Zhang and Lee [2003]. Information Retrieval (IR) algorithms are compared and show that the best question classification result comes from Support Vector Machine (SVM). A recent approach for the problem of question classification is presented in Ray and S. Singh [2010], where WordNet University [2012] and Wikipedia Wikipedia [2012] are used to find patterns in the query to classify the questions.

Most of the QA techniques for semantic web (Lopez et al. [2005a], Cimiano et al. [2006], Wang et al. [2007a], Lopez et al. [2009], Tablan et al. [2007]) have a major ontology dependency. Some of them try to reduce the effect of this problem by using automated ontology finder. First problem with this approach is the amount of required resources, which may be minimal, based on the specifics of the domain. The

second problem is the time complexity of the ontology finding and matching, which is usually very expensive to implement at runtime. Also some of these approaches need specific customization for different domains which makes it impossible for them to be used by general users.

In Kauffmann et al. [2006], an approach has been introduced which answers *Wh-* type (e.g., what, where, etc) questions based on a knowledge base. In this approach, for the disambiquity of questions, the interface asks users for the precise question from the KB. When a KB is chosen, the RDF triples are loaded into a Jena model, using the Pellet reasoner to infer all implicitly defined triples and WordNet to produce synonym-enhanced triples. Pattern matching is then performed by searching for triples that include one of the nouns or verbs in the query. Another work more focused on finding the answers is presented in Unger et al. [2012], includes finding the answer using SPARQL (SPARQL Protocol and RDF Query Language) and creates different patterns to find the answer in RDF files from the Web. Similarly, in Bernstein et al. [2005] and Damljanovic et al. [2010] the user query is translated to what exists in the KB, questions are suggested using the user interests and the knowledge base. Users, generally want to directly find their answers without struggling with inputting extra information, our approach suggests a way, using answer types as a top level ontology, to provide the final answer without collecting extra information from users.

2.4 Upper Ontologies and Related Tools

As mentioned previously to map common sense to a logic based computer comprehensible format we need two tools. As such, we can divide the related work to this approach to two portions. One part is the works related to *ConceptNet*, creating a common sense knowledge base using different methods. And the second part is the works on defining relations and creating upper ontologies. The other field of

work which worths mentioning, but as there hasn't been as many works in it we decided to not create a separate subsection for it, is the tools which has been used to create any of the works we mention in the following. These tools have been created to extract, gather and organize information from different resources. Each of these tools generally works with specific kind of resource. For example *Reverb* (Fader et al. [2011]) is focused on extracting information from web pages, *Pellet* (Parsia and Sirin [2004]) is a reasoner which can be used in Java or in softwares such as *Protege* (Noy et al. [2001]). *Protege* is a tool which provides the users with necessary instruments to create, understand and analyze ontologies and other SW resources.

An upper ontology is an ontology which describes different concepts in a general sense suitable for use in multiple fields. While the approach to upper ontologies has been controversial at best (Floridi [2008]), the need for an upper relationship management as backbone for any other ontologies and information presentation is generally acceptable and understandable. For this purpose, multiple organizations and research groups have generated different upper ontologies for different purposes. In the following, we shortly describe some of these works.

The *Basic Formal Ontology (BFO)* (Arp and Smith [2008]) is a small Upper ontology specifically designed for information retrieval, analysis and integration to scientific and other domains. The important consideration of this ontology is the lack of focus on physical and specific entities which makes it possible for the ontology to be used in many different fields. Majority of current practices of this ontology can be found in biomedical and security ontologies. An example application of *BFO* can be seen in the Ontology for Biomedical Investigations (OBI) Brinkman et al. [2010].

Open Robots Common Sense Ontology (ORO) (Lemaignan et al. [2010]) is an ontology created for use in AI and robotics with focus on properties. *ORO* has been implemented in Java and as such has dependency to Java Virtual Machine, Jena RDF

triple store, and Pellet for reasoning. The ontology has been maintained from 2008 to 2011. Considering the approach of this work, it is noteworthy that there are multiple parallel relations as data properties with binary range to create better environment for robotic data presentation.

Simple Knowledge Organization System (SKOS) (Miles and Bechhofer [2009]) is the W3 recommended approach to provide an easier migration path to convert data to RDF and other SW formats. It has been specifically created for conversion of thesauri, taxonomies, classification schemas and subject heading lists. Using the *SKOS* primer we can identify 5 main components of *SKOS* definition. First, "Concept" to present any unit of thought. Second, "Labels" to add description to concepts. To present the facts about concepts, a different component is used, which has logical difference with labels, called "Documentary Notes". To describe and understand concepts further, "Semantic Relations" is used to connect the concepts and clarify their meaning. Finally, "Concept Schemas" is used to present the used vocabulary to describe the concepts.

One of the suggested ontologies by W3 is *Dublin Core* (Weibel et al. [1998]). *Dublin Core* focuses on enabling ubiquitous access to cultural and scientific resources through galleries, libraries, archives and museums (GLAM). This goal is achieved by providing specific properties and classes suitable for this need such as language, license and publisher.

Talking about most acclaimed and used ontologies, *Friend Of A Friend (FOAF)* (Brickley and Miller [2012]) has been a dominant ontology to present people and organizations since the beginning of the SW movement. The goal of *FOAF* is to provide a standard vocabulary for generating and presenting personal and organizational information such as name, address, email address in the SW format as part of *WebID* (Sporny et al. [2011]) standards.

To name some of the other ontologies widely used in different context, we mention *SIOC (Socially Interconnected Online Communities)* to complement *FOAF* to describe the products of forums, blogs mailing lists and wikis; *GO (GoodRelations)* (Hepp [2008]) to describe products sold online; *Music Ontology* (Raimond et al. [2007]) to describe information related to music industry (not the music itself).

While most mentioned works focus on general approach to infrastructural path to entity representation, there has been different tries to map human emotions to an ontology. The two major works in this field are *HEO (Human Emotion Ontology)* (Grassi [2009]) and *Smiley ontology* (Radulovic and Milikic [2009]). *HEO* mostly focuses on emotions and emotion representations while Smiley generally presents ways to express emoticons by different metrics and descriptions.

Discussing upper and generally used ontologies, it is important to mention *Schema.org* and the general acceptance of its different ontologies in SW community. Unfortunately the OWL version of the ontology on their web site is old and has not been maintained, but TopQuadrant has generated a new and reasoned version of the ontology in OWL format for public use. The ontology main classes are action (any action performed by and agent), creative work (any creative work in any field), data type (including basic data types such as integer), event (any event in any field), intangible (a class to cover many intangible "things" such as quantities), medical entity (any entity related to medical field such as tests studies and devices), organization (covering different organizations), person (covering any person, dead, alive or even fictional), place (entities that have fixed, physical extension) and product (anything which is made available for sale).

CHAPTER 3: QUESTION ANSWERING USING SEMANTIC WEB

3.1 About Question Answering Systems

Automated QA systems can facilitate using semantic technologies to better cater towards the needs of the users. Current QA systems use a "search engine approach" where a user goes through multiple pages and filters them to find the answers s/he needs, in other word, it depends on user intelligence to find the answer. We argue that if QA systems are supported through semantic technologies, it can use machine intelligence to return the desirable result and hence improve user efficiency. In recent years, there have been multiple attempts to add this functionality to search engines Ko et al. [2010]. Some search engines like Google have implemented a QA system for simple questions, that searches for answers by parsing the search records and then ranking the answer. On the other hand search engines like ask.com keeps a repository of questions and answers shaped from user queries and experts answers.

The first approach lacks the deductive power to answer multi-level questions. For example it can easily answer the question: "Who is the president of United States?" but it cannot answer questions with two or more levels of deduction like "Where did wife of Barack Obama graduate from?". It cannot conclude that the phrase "Wife of Barack Obama" refers to Michelle Obama so it has to look for where she has graduated from. Moreover this approach is computationally expensive, since it needs to crawl pages for specific answers and rank them at runtime. The second approach is limited by the number of questions and answers that are stored in the repository. Considering the growth rate of topics and questions on the Web, any repository cannot be expected to keep up with this growth.

In light of these issues, we need a solution that addresses two main concerns: First, it should be computationally inexpensive. Second, it should be able to keep up

with the growth of the Web and the ever-changing nature of information. The goal of this paper is to propose a new approach to enhance QA systems with semantics. We believe this approach can alleviate the current issues with QA systems. In terms of efficiency, searching for the answer in different scenarios, on the Web or in a knowledge base, is the most computationally expensive part of a question answering systems. We propose the use of RDF triples to reduce this efficiency bottleneck. By combining question types with RDF triples , we try to widen the grasp of our approach and make it ontology independent.

Question answering systems mostly consist of three main phases: Question Processing, Solution Access, and Solution Verification. In this paper we focus on the first phase i.e. question processing. Since we search for the answer in RDF files with the type and part of the query encoded in it, this greatly reduces the need for verification of the answer. The organization of this paper is as follows: In Section 2 we discuss the question classification problem. Section 3 presents our algorithm for translating the question to its RDF equal. In Section 4 we show how to find the answer in the RDF repository. Finally Section 5 presents our experimental results.

3.2 Question Classification

There are multiple ways to categorize questions. Many works try to be as specific as possible about getting the type of the answer, but since our main focus to find the answer is not a specific answer type, we can look at the problem from a more general perspective. There are many kinds of questions we can potentially consider, consisting of factoids, lists, definitions, hypothetical, causal, procedural and confirmation queries based on early Text Retrieval Conference (TREC) evaluation. We divide our classification procedure to two parts. For the first part we consider the factoids. This group is simpler to classify based on the question words in the query. In our work we parse the query for the limited set of the question words and if there is

one, it simplifies the solution as we describe in the next part. The second case of the classification is for the cases when we don't have a complete question as the query. The most common case for this category is when the user only uses the keywords. The ontology which we use, consists of the basic types, as we use in the question classification. These types has been gathered from WordNet, general types in similar works and our experiments. Also we have added functionality to automatically choose possible type candidates and suggest them to the users.

When in the process of finding the type we find a type as our first result which is not in our typeset, we keep it in a database. We also keep the frequency that we have found this type in this database. If the frequency of a type passes a predetermined threshold we suggest this type as a new type to the system administrator. The reason that we don't suggest this type to the end user is the need to keep a regular expression of the format of the type in our data set and the end user generally doesn't have the expertise to put it in the system.

For noun we have the following types: Time, Event, Food, Body, Plant, Substance, Artifact, Location, Person and Act. For verbs we use emotion, change, motion and consumption.

In the following, we provide details on these two classification categories.

3.2.1 Factoids classification

For category definition for factoids we extend the approach presented in Singhal et al. [2000] and expand it to meet our needs. The mentioned approach has the following basic types: people, locations, organizations, quantities, dates, and linear measures. We combine the *people* and *organization* to one type. Furthermore, we divide quantity type to *age*, *distance* and *quantity*.

In our question classification algorithm, we categorize questions into two main and 12 sub-categories. For the first category, we consider *True/False* questions. These

Table 3.3: Question types and subtypes

Question Word	Answer Type	Example
When	Time/ Date	When did Barack Obama graduate from university?
Where	Location/ Place	Where was Barack Obama born?
Who	Person/ Organization	Who is Barack Obama?
Why	Reason/ Text	Why did James Dean die?
Whom	Person/ Organization	Whom is Barack Obama married to?
Whose	Person/ Organization	Whose carpet was flying in stories?
How	Reason/ Text	How did James Dean die?
How many	Number/ Quantity	How many books are in the library of congress?
How long	Distance/ Quantity	How long is river Nile?
How far	Distance/ Quantity	How far is Detroit from New York?
How much	Number/ Quantity	How much is a stamp?
How old	Old/ Quantity	How old is Barack Obama?
How often	Frequency/ Quantity	how often is the world cup?

questions check the truth value of a statement. This type of questions don't have a question word at the beginning and we can always assign one of true or false values to them. As an example, the question "Is Barack Obama the president of United States?" starts with a *to be* verb and it's answer is currently true. As these questions have a boolean answer, we classify all of them together as *True/False* type questions. The second type of the questions are considerably more complicated to categorize. These questions begin with *Wh-* words and can have different kinds of answers ranging from information and text to a single date. In Table 1 we categorize such questions to more specific subjects: We categorize the *Wh-* questions into seven different categories. Six of these categories are simple *Wh-* words, the seventh one is *how* which itself divides into six categories.

One special case is with *How*, in our work we consider two different cases, if we have a noun right after how, we follow the procedure in the next section, otherwise we consider the type Reason/Text. In the case of *Which* and *What* question words, to find the type of the answer we follow the same procedure as the next section after removing the question word.

3.2.2 Keywords classification

We considered two different tools for this portion of our work. One possible approach we started with was Natural Language Parsers. After experimenting with different kinds of questions and different parsers we decided not to use NL parser for two main reasons. The first issue with these parsers is their time complexity. Their speed, compared to our whole process is considerably slower. The second issue with Natural Language Parsers happens when the user puts wrong grammar and/or just keywords. In this case the parsers cannot translate part of speeches correctly.

The second tool we used was N-grams. N-grams is considerably faster and better in understanding group of words. Also because N-grams use frequency of words in different contexts, for the new language phrases they return better results. In this paper we use Microsoft N-grams tool as our main decision maker to detect group of words.

The process of classification starts by defining group of words. We base our classification mainly on type of the noun. First, we look for the first group of words or the first noun. The type of the answer will thus be the type of the noun or group of words. We extract these types from WordNet. In case of group of words, there is a high possibility that we don't find the type in WordNet. In this cases, if there is a single noun in the group of words, we consider the type of that noun as the type. However if there is no noun or multiple noun, we use pattern matching to create the RDF query of the form "*Group of Words* is a". We look at the results and select the highest frequency "type" that we have in our types set.

3.3 Query Translation

The goal at this step is to convert the user query to an RDF tuple. The conversion simplifies our process toward creating the SPARQL to find the answer.

We divide our algorithm into two parts. The first part, the algorithm transfers the natural language query to a triple and in the second part we expand the algorithm further to incorporate complex question types.

We divide the possible user queries to 3 groups. In all the cases we use N-gram to find group of words. The other similarity in all the cases is the use of WordNet and English corpus to find the most frequent synonyms of the words. Our goal is to have verbs for predicate in the most possible cases. If what we find as predicate is not a verb we use the word definition and find the related verb from it as the predicate. To integrate all the created triples and queries with similar meaning together; we follow a two step procedure, first we look up the synonyms of the word from WordNet, at the second step we get the frequency of each of the synonyms from the Corpus of Contemporary American English (ENGLISH [2012]). This corpus data-set consists of 450 milions most common words in English with their frequencies. And then we store the triple with the new found term.

The first case of these three groups is when a user puts a complete query in the system. The user puts the complete question, with the correct grammar and dictation, e.g.,"When is Barack Obama Birthday?". In this situation the query starts with a question word, i.e., What, Where, etc. or a to be verb. We can use the question word to decide on the type of the answer. The same situation happens with the queries which start with a question word but do not have other parts of the complete sentence except keywords. The last case is when the user just inputs the keywords without question words and the non-keywords in the query. The problem which arises in this situation is the possibility that we don't have enough information to categorize the question. For example if the user query is stated as "Barack Obama born", where the motive is to look for the day that Barack Obama was born, we will not have enough information from the query to return a date. On the other hand if the user inputs a

query like “Barack Obama Birthday” we can conclude that date should be retrieved. If after the classification we can’t finalize what type of information we are looking for, we create a general query and return all the related content to the query to the user.

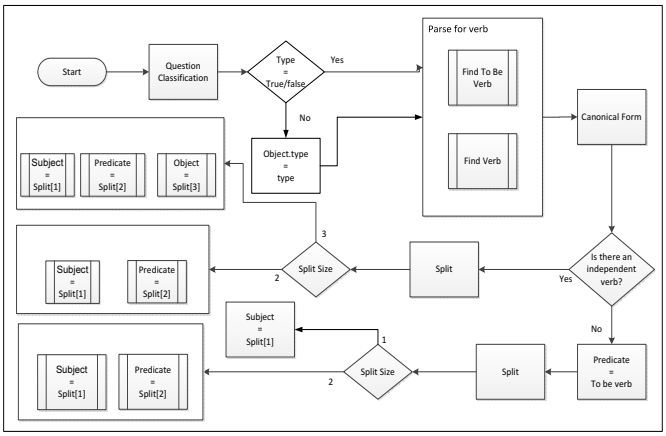

Figure 3.1: Query translation

Figure 3.1 shows the process of creating a triple out of the user query. The function `QuestionClassification()` that is invoked in the first step of the flowchart implements the classification we described in the previous section. We then assign the type we found to the object type if the type is not *true/false*. The next step finds both the *to be* verb and ,if there exists any, a normal verb in the query.

After detecting the verb in the question, we remove the *wh-* word and stop words from the query. The goal here is to create a type of a canonical form for our query. Since many of the stop words can change the meaning of the query, we have to be careful about removing all the considerable stop words, specially when there are connecting group of words. The distinction between these two kinds of stop words can be made by analyzing the query using N-grams and separating words which are in the middle of nouns from the other type which are not connecting two or more nouns together. For example if we have the group of words “The united states of

America", we remove *the* at the beginning and not *of* in the middle as it divides the two groups of nouns.

For the canonical form we perform the split operation in two different cases, one in case of a *to be* verb and one with normal verb. In both cases the next step is to run the split function. This function divides the query into groups of words based on the connection words and the relation of words together. The logic is similar to what we did to create the canonical form, i.e., find words which are not connected to each other and split them. In *to be* case, depending on the number of groups we may have *subject* and *object* or just the *subject*. In the latter case, we either have *subject* and *predicate* or we may have all three parts of the triple.

The group of words that comes before the verb can be grouped together as a primitive form for our subject. The verb itself, which is the remaining part of the split, will be the primitive form for the predicate (and we have the type of object from the previous section). The specific steps for the example query "Where was Barack Obama born?" are then:

1. From the result of the previous section we know that the type of the answer we are looking for is "Location/Place".
2. Looking for the verb in the question, which in this case is "born".
3. Removing the Wh-word and any extra words in the query. The result is "Barack Obama born".
4. Group words before the verb as subject which is "Barack Obama"
5. Group the verb as predicate which is "born".
6. The triple is ready: {Barack Obama, Born, ? (Location/Place)}

For the *True/False* type of questions we adopt a slightly different approach as we only want to check the truthfulness of the query. In this case, we have all three parts of the RDF triple in the query. The sequence of steps is very similar to the

previous algorithm. We start by looking for the verb in the query. If we find another verb except the *to be* one, we go to the next step. If not, the *to be* verb is our final predicate.

For the case that we have another verb except the *to be* one, the first step is to remove the stop words. Then whatever comes before the verb is the subject and whatever comes afterward is the object. When we have the second case, just one *to be* verb in the query, we divide the remaining part of the query into two groups of words. To find the connection between each group of words we look for transition words. When we have two nouns next to each other without a transition word in between, we can consider that part as the dividing point for the two words. The first group of words is the subject and the second group is the object of our triple. If we consider the *to be* verb as our predicate, we have all parts of the triple. We show how this procedure works on a sample question: "Was Barack Obama born in 1961?"

1. From the previous section we know that we are looking for a boolean answer for this question.
2. Looking for the verb in the question, which in this case is "born".
3. Removing any extra words in the query. The result is "Barack Obama born 1961".
4. Group words before the verb as subject which is "Barack Obama"
5. Group the verb as the predicate which is "born".
6. The last part of the triple is "1961" as object.
7. Now the triple is ready for the next section: {Barack Obama, Born, 1961}

3.3.1 Algorithm Expansion

Currently the radius of support for the RDF repositories is not vast enough to answer the need of the general users. This section tries to improvise a solution for this problem. There are a lot of cases that two questions together are pointing to the same subject and can answer each others. To clarify this point we consider two sample

questions “Who is Barack Obama?” and “Who is the president of United States?”. In our algorithm the answer of one of these questions is the subject of the other one. So, from previous section for the first question the triple is {Barack Obama, is, ? (Person/Organization)} and for the second one it is {President of United States, is, ? (Person/Organization)}. We analyzed a number of different questions and found that in many cases we can switch the place of object and subject in our triple and output both triples to the next section. With this approach we can create these two triples {? (Person/Organization), is, Barack Obama} and {? (Person/Organization), is, President of United States}. So, answer to one of the questions can solve the second question as well. Obviously we first look for an answer for the direct triple from the query and only if the answer cannot be found from it, we try the second query.

To make the triples more accessible and understandable we can consider the case where a query has a group of words pointing to an answer to another question, and as a result by answering that question we can make the original query shorter. We mentioned a sample of such questions in the introduction: “Where did wife of Barack Obama graduate from?”. In this case “Wife of Barack Obama” points to another question, by answering it we can make the original question simpler. There are two cases we consider regarding this issue. The first case we consider is if the group has the word *of* in it. We create a new triple that consists of word(s) after *of* as subject and the word(s) before as predicate. For this case, the type of the object is as the type of predicate we just chose. To find the type we do the same as the case of *which* question word before. The second case is when we have two questions associate with each other in one question e.g., “who was the president who died in office”. Our approach to these questions is by dividing them in two parts and answer them separately. After finding the answer we look for the same answers for both questions and consider it as final answer for the query.

3.4 Finding the answer

To find the answer for the query, we will run the result of previous sections in the RDF repository. To bridge the RDF triple from the last section and repositories we use SPARQL query language.

The result of previous step divides the triples in two different cases. The first case is with Wh- questions. In this case, we have two parts of the RDF triple and the type of the third part. For instance, for the query: "When was Barack Obama born?" the result of Section 4 shows that the answer type is "Time/Date". (The triple returned from the process in section 4 is {Barack Obama, born, ? (Time/Date)}). In this section, we create the following SPARQL query and run it in the RDF repository. Note that we get the basic type of the other parts of the triple from WordNet.

```
SELECT ?date
FROM <RDF repositorty.RDF>
WHERE
{ <http://name#_Barack_Obama> act:born ?date.
  ?date rdf:type type:date. }
```

The result will return the object of the triple which is the answer for us, or will return nothing which shows we don't have the answer for the query in the repositories. We return this part to the user as the final answer.

The second type of the question and results are "True,False" questions. The process from previous part showed the type of the answer as True or False and returned all three parts of the RDF triple from section 4. The goal here is to create as many RDF triples that are descriptive and short as possible.

We have to categorize all the connection words, connect them with one of the categories in our ontology. For example for "at" we need to connect it to location. Also the same applies for connection words between sentences, i.e., "Born in Honolulu,

Table 3.4: Natural language query conversion into RFD

Question Word	Correct Answer Ratio
True/False	0.85
What	0.73
When	0.87
Where	0.85
Which	0.67
Who	0.81
How	0.86
How long	0.89
How many	0.73

Hawaii, Obama is a graduate of Columbia University and Harvard Law School, where he was the president of the Harvard Law Review."

3.5 Experiments and Results

We conducted experiments with a small data set to verify the applicability of our approach. For the first set of experiments we used top 50 queries for each category(i.e. 50 for what, 50 for when etc.) of questions as retrieved from Google and used our algorithm to convert them into RDF triples. Table 3.4 shows the the results of our experiment.

We can see from the table above that our algorithm performs consistently good for all types of questions. However, *Which* type of questions are a little tricky to convert since they mostly have complex nouns and complex verbs. Similarly, for the *How many* type questions we mostly have a complex verb, such that one of them is mentioned as a part of the other one. Currently our approach cannot handle such relationships. Hence we have a low conversion rate for these types of questions.

For the second set of experiments we considered question subsets from Dataset [2004]. We compare the answers generated by our approach with the first answer retrieved from Google query. We took a sample data set from Data [2012]. We used the RDF triple generated for the query translation to find its answers both in the RDF files as well as search query in Google search engine. We used wikipedia as the information source for semantic serach for the experiments. We used 20 random

Table 3.5: Question types and subtypes

Question Word	Google Answer Ratio	SW Answer Ratio
What	0.57	0.52
When	0.85	0.90
Where	0.83	0.95
Which	0.50	0.15
Who	0.77	0.66
How	0.50	0.50
How old	0.36	0.56
How long	0.98	0.98
How many	0.50	0.50

questions of each type (as defined in table1) and repeated the experiment 10 times. Then we averaged out the numbers and the results of these experiments are presented below. Table 3.5 shows the results of our experiments.

We can see from the results that our techniques performs fairly good in comparison to the Google search results. Our approach performs better than Google for *When, Where and How old* types of question where as Google does a better job at *What, Which and Who* types of question. Where as we see similar results for both Google and our approach for *How, How long and How many* types of questions.

We can see from the results that the best solution was found for quantitative type questions. It is attributed to the fact that if an answer is found it is very probable that it will be a correct answer. Since a no-match found would be termed as an invalid answer. The text based answers had the minimum ratio for successful answers. The reason for the low score lies in the fact that our technique works of exact matching and the fact that for these types of questions we may have multiple answers and all of them could be valid/true. This makes it is very difficult to figure out the best answer for text based questions and makes it harder to put much confidence behind a possible answer candidate. Similarly we can see that number based answers i.e. when, how far etc did show better results. These results give us better insight into the semantic QA process. These limited sets of experiments show the applicability of our approach and serve as a proof of concepts for our solution.

Our proof of concept experiment run highlighted some of the problem points of our approach. The first one belongs to the semantic meaning of the combination of query words, for example the question “what is the biggest hit of Insane Clown Posse” we have to interpret the term ”biggest hit”. Now there could be multiple interpretations of the term biggest hit e.g. the biggest hit in terms of revenue or popularity or number of records sold etc. The missing information could be guessed using the heuristic measures base on frequency of words i.e. biggest hit is mostly associated with the number of albums sold. However this is not the semantic meaning of this combination of words hence this type of questions are difficult to answer. The second issue is when a question has a domain specific multi stage answer e.g. if we ask the question ”who discovered prions” in this case there is no single subject answer for this question since discovery of prion is attributed to three different stages. During the 1960s radiation biologist Tikvah Alper and mathematician John Stanley Griffith developed the hypothesis, Francis Crick recognized the potential importance of the Griffith protein-only hypothesis for scrapie propagation in his book and finally in 1982, Stanley B. Prusiner of the University of California, San Francisco announced that his team had purified the hypothetical infectious prion. Hence we can see a lot of domain specific knowledge is needed to construct answers for this kind of single questions. The third issue is with the questions of the type ”who was the lead singer of nirvana”. Our current approach of using N-grams will rank the N-gram lead singer as the most appropriate search tuple since it has a very healthy frequency. Although lead singer could be translated into singer by a human who knows the semantic rule that if there is only one singer in the band, then he/she should be the lead singer. But these types of domain specific rules do not exist for question answer systems. One of the approaches used to overcome this problem is to use the root of N-grams

being searched however this is computationally expensive and does not work in all cases.

CHAPTER 4: CONCEPTONTO

4.1 Introduction

In the beginning, the goal of World Wide Web was creating the most comfortable presentation of information similar to books and catalogs. This approach was later followed by number of technologies, such as HTML and CSS, which made the transition of information to human friendly presentation possible. The exponential expansion of World Wide Web introduced a new predicament to extraction of information from the Web. The advancement of search engines such as Yahoo at the time was a sign of this need. Overtime, researchers found out the main issue with the current form of the Web is the lack of understanding on part of machines on "common sense and knowledge" of humans. There have been two general approaches to solve this problem. One short term solution has been to use different algorithms on information retrieval and machine learning to retrieve the necessary data via a fringe understanding of information for machines. On the other hand, the conversion of information to a bridge format which is both understandable for machines and humans is a long term approach which has been chosen by many of the researchers in the field. Semantic Web (SW) is a general term used for many of technologies have been created for this purpose. The key stone for these technologies in this regard have been RDF and OWL for representing information.

The Semantic Web (SW) while comprehensive, needs a deep understanding of human common sense knowledge to understand basic information which seems primitive for most humans. As the base of human communication is relations between concepts, the first step to form this understanding is to create an ontology which maps basic human relations to different concepts while understanding some basic requirements of those relations. The field of Common sense conversion and common

sense knowledge bases follows different directions to map the relations and common knowledge to create the data-sets. Some works follow the manual information gathering approaches such as asking for data from users, and some other works are more focused on creating instruments to gather required information automatically.

Our analysis of different works in common sense knowledge retrieval and presentation has showed that *ConceptNet* (Liu and Singh [2004]) is one of the more comprehensive and extended knowledge bases available for public use. Open Mind Common Sense knowledge-base was founded in 1999 based on simple information gathering approach from normal Web users to generate simple triples with over 30 basic relations. Later on, this work was expanded by the addition of *WordNet* and *Wikipedia*. The generality of the presented knowledge, and the simplicity of the relations and the information makes *ConceptNet* a formidable data-set for generation and extraction of relations.

In this paper we introduce the process and steps of creation of *ConceptOnto* to map the mentioned relations to their equivalent in OWL. The relations are based on the default relations of *ConceptNet* and addition of some equal or useful relations which we consider useful for an upper ontology with the focus on common sense. The ontology in OWL format is available for general use on our Web site [5]. In *ConceptOnto* our focus is readability for humans, maximizing the functionality while saving the generality of the ontology. Our goal in here is to present a through explanation of concepts and terms in our ontology so a general user can start implementing and using this ontology to create new SW data representations as needed. Throughout this presentation we have tried to consider the major points, however, we encourage and welcome any suggestion to improve the ontology.

[5] http://score.cs.wayne.edu/ConceptOnto.owl

The rest of this paper is organized as follows. First, we explain some of the terms used in this paper which need explicit explanation or needs further description based on the context. In Section 4.3 we describe the process and details of *ConceptOnto*. Section 4.4 presents some of the fields which we believe can benefit from our work.

4.2 Term Definition

In different sections of this work there are a few concepts which need clarification. In the following we describe some of the terms which we believe are more helpful to follow the process and the logic behind parts of the process.

- Open World Logic: In the context of common sense knowledge, open world logic means that every statement can be true unless the opposite is known as a fact. For example unless we specify that the location kitchen cannot be the same location as garage and car is in garage, then the reasoner cannot point out that the car is not in kitchen.

- Common Sense Knowledge Base: Representation of the knowledge that most people generally possess, in a way understandable for intelligence programs which can use natural language or make inference about the world.

- Transitive Relation: A relation between two items is transitive when we can conclude that if a is connected to b with this relation, and b is connected to c, then a is connected to c. For example "table isLocatedNear chair", and "chair isLocatedNear TV", then we can conclude that "table isLocatedNear TV".

  $\forall a, b, c \in X : (aRb, bRc) \Rightarrow aRc$

- Symmetric Relation: A symmetric relation means that this relation holds for both sides, if a related to b, then b is also related to a. For example "abnor-

mal isSimilarTo exceptional", then we can conclude "exceptional isSimilarTo abnormal".

$\forall a, b \in X : aRb \Rightarrow bRa$

- Reflexive Relation: A reflexive relation means that any item with this relation is related to itself. For example "bird isRelatedTo bird". While on the first look many of the relations, such as isRelatedTo in this case, do not make sense as a reflexive relation, we can see in multiple real world scenarios that defining them as reflexive increases the functionality and usability of our ontology. To clarify this point consider the triple "Bird isRelatedTo Bird". Using this instance in a set of items we can see the relation of two instance of birds, such as peacock and duck compared other animals or other items.

4.3 Methodology

Our ontology consists of two main parts which we describe separately in the next two sections. In the first section we introduce and define the classes implemented in our ontology and some of the uses in different relations. The second section focuses on properties in *ConcpetOnto*. We define the different properties, their relation with each other and corresponding relation properties.

4.3.1 Classes

To identify base classes to be implemented in our ontology (considering that the goal is to create a general purpose relation ontology) we returned to the basic entities of *ConceptNet*. These classes consist of four general purpose items. We provide a brief explanation of these classes in the following.

Noun Phrase (NP) is by far the most common entity in *ConceptNet*. These phrases normally consist of one or more main noun as the root and one or more other

parts to clarify the noun. For example for the relation isDefinedAs, for any triple, the object and subject are *NP*.

Verb Phrase (VP) is any phrase which has a verb as its root. These verbs can be precedented or succeeded by any other word. An example of a *VP* is the property "isCapableOf" with the triple "bike isCapableOf moving_forward", moving_forward is a verb phrase with the root of the verb moving followed by the adjective "Forward".

Adjective Phrase (AP) is the general term uses specifically in range and domain of properties such as "hasProperty". Any *AP* can consist of a set of words which in turn can have multiple adjective, noun or even verbs. For example in triple "bike hasProperty common_In_Asia", common_In_Asia while starts with an adjective, follows by a noun to complete the concept.

Terms are the last class of entities in *ConceptNet*. Terms are general phrases used in "isDerivedFrom" or "isTranslationOf" which shows the relation between two phrases when one derives from the other one or the phrases are equal in different languages. For example in the triple "begin isDerivedFrom start" subject and object are verbs while in "earth_science isDerivedFrom earth" the object is a NP and the subject is a noun. This generality of concept in Terms make all the other classes subclass of this class.

To compare classes in *ConceptNet* with classes in *WordNet* (as an example of a similar work) we like to mention a few key differences. First, the classes in Conceptnet are comparably very general. For example for NP, any phrase with a noun root belongs to this class. Second, in works like WordNet there is a high focus on linguistic analysis of terms, while considering that in Conceptnet the main focus in on common sense, the relations are based on common sense which can have ambiguities meaning depend on general user perspectives and understanding of different concepts and relations. Compare to classes in *SKOS* Miles and Bechhofer [2009] , *SKOS* is focused on units of

thoughts and concepts (using equally in SKOS) while completely ignore any linguistics of concepts. Another generally used upper ontology to discuss is BFO Arp and Smith [2008]. Classes in BFO are divided by their temporal identities. If a concept is independent of temporal properties, then it classifies as continuant. On the other hand if a concept is dependant to a temporal variable it classifies as occurrent. In our ontology there is no focus to identity of concepts time wise. GFO Herre et al. [2006] is another upper level ontology with focus on sets as its entities. It separates entities to two, items belonging to sets (based on ZFC Barnes and Mack [1975] set definition) and items which do not belong to sets. *Relation Ontology (RO)* Smith et al. [2005] is possibly the closest ontology to our approach regarding the properties, but on the subject of classes, the focus is mainly on sets and synonyms. Also as this ontology still evolving, in the newer versions there is an obsolete class for the classes that has been replaced. *UMBEL* UMB [2012] has a different approach to class definition. While separating concept as a unit of thought, it defines superclass which consists of mostly disjoint classes for other entities such as people, food and diseases. This through classification makes it easier to introduce new concepts, but most of these classes can be used in simple triples such as "Pizza isA food" with further explanation.

4.3.2 Properties

In this paper we use properties in place of relationships as been used in SW context. While the majority of the properties are directly dictated from ConceptNet relations, We make several modifications to improve the functionality and mobility of the ontology for making it compatible to open world scenarios. We have to remember that properties in ConceptNet are not definite as in the general ontology relations. Even if any of the properties are not true in a logical sense, it is possible it "makes sense" which is the definition of common sense and the main point of difference to any other information gathering approach. Another point to consider toward

different relations is the cultural difference between different languages in ConceptNet. To clarify this point consider the property "desires" in the following triple which is translation of Korean: "cockroach desires slippers" which in English does not make sense as slippers are not the desire of cockroach.

The first major modification is concerns to naming. As the general approach to property creation in ontologies, we try to modify the relations by adding the two keywords "is" and "has" based on objective or subjective meaning of relations. This naming methodology defines the difference between having a specific property versus the entity resides in another entity. This is why if a property begins with "is" the reverse begins with "has" and vice versa. While we haven't found the use of "is" and "has" in this extent in similar works, this methodology is suggested in main OWL tutorials Horridge et al. [2004].

The second change is the use of reverse properties. Considering the open world logic, generating reverse properties, while *ConceptNet* originally does not include most of them, let us analyze more possible scenarios of events and concepts. Another effect of reverse properties is the generalization benefit. The data from *ConceptNet* is provided from general users perspective toward knowledge, so the information are concerns with the general approach to common sense which in most cases considers reverse relation an obvious logical conclusion. On the other hand, in SW logic, unless you directly point to the fact to what reverse function means and if is true or not, we cannot indirectly include the reverse functions. It is important to note the same logic for the transitive, symmetric and reflexive properties. Finally, there are only three properties which have subproperty/superproperty relation. The properties LastSubeventOf and FirstSubeventOf are sub properties of SubeventOf. In the context of OWL properties this means that every instance of the two mentioned properties is an instance of SubeventOF.

An important property we need explicit attention to is the "IsA" relationship. RDFS:SubclassOf has the same specification and meaning as IsA in general case. We believe implementing IsA in the ontology has two main benefits. First the domain and range of IsA relation in *ConceptNet* is NP. While the RDFS:SubclassOf does not have this limitation, in general we believe the functionality is used for noun phrases more than any other class of words. The second benefit of an explicit IsA property is better readability for human eyes which makes the presentation of the information easier in different cases. For the same case in RO Smith et al. [2005] ontology the preference has been on using RDFS:SubclassOf instead of implementing IsA.

After analyzing different data from both *ConceptNet*, and other knowledge bases and ontologies we decide to add some other properties which makes the conversion and addition of different sources easier. The first set of properties is in regard to creation and demise of any entity. This addition which in both cases are data properties (different from object properties which are native of *ConceptNet*) have a range of literal which is dates in this case. It is important to emphasize that these properties can be used for humans, in the concept of born and death, buildings, in the concept of being built and demolished, or even cities, in the concept of the first settlers to the last citizens. While we can create sub-properties for each of these cases, we believe while having one relation for all simplifies things, it also creates a unified way to present different concepts, while they are differentiable by their other relations such as IsA properties. Another addition to *ConceptOnto* is existential relation between concepts. These properties are modified version of properties implemented in *General Formal Ontology (GFO)* Herre et al. [2006] as depend_on and necessary_for. We have changed the names to isDependOn and isNecessaryFor and changed the domain and range of it from Item in *GFO* to Term in *ConceptOnto*.

As mentioned previously, RO is the closest ontology to our work mainly because of the closeness of properties implemented. To consider the similarities of the two ontologies we use the EquivalentObjectProperties to define the equivalency of the properties such as isDerivedFrom from our ontology to derived_from in RO to unify and make the process of mapping different ontology and resources faster and easier.

Finally to check, expand and extract implicit relations in *ConceptOnto* we tried two different reasoners to find the best addition to our work. After through analyze of FaCT++ Tsarkov and Horrocks [2006] and HermiT Shearer et al. [2008], we used HermiT to make the final modifications to our ontology such as using the equivalent relations for expanding the inverse relations.

Table 6.11 shows the properties implemented in *ConceptOnto*. The first column shows the name of the relation as in the ontology. Second column shows the specific properties of the relation which consist of Transitive, Symmetric and Reflexive. If the inverse of the relation has also been implemented in the ontology, its name can be found in the third column of the table. The fourth column is the original name of the relation as available in *ConceptNet*. If the relation has an equivalent in one of the discussed ontologies in previous sections, its name is available in the fifth column. And finally, the sixth column of the table shows the domain and range of the relation of *ConceptOnto*. For example, the first row of the table is describing the property isSimilarTo which is both transitive and symmetric. This property does not have an inverse and originally presented as SimilarTo in *ConceptNet*. An equivalent of this property is presented in *Relation Ontology (RO)* as SimilarTo and finally the property goes from Noun Phrase (as its domain) to Noun Phrase (as its range).

4.4 Use Cases

While the main benefit of a general use upper ontology is to represent information, this approach can be useful in many different fields of research and practical

Property	Properties	Inverse	Original Property	Equivalent	Domain → Range
isSimilarTo	T, S		SimilarTo	RO:similarTo	NP → NP
isAtLocation			AtLocation		NP → NP
isCapableOf			CapableOf		NP → VP
isCreatedBy			CreatedBy		VP → NP
isDefinedAs			DefinedAs		
isDerivedFrom		isDerivedInto	DerivedFrom	RO:derives_from	
isLocatedNear	T, S, R		LocateedNear		NP → NP
isMadeOf			MadeOf		NP → NP
isPartOf	T	hasPart	PartOf	RO:part_of	
hasPrerequisite		isPrerequisite			NP,VP → VP,NP
hasProperty		isPropertyOf	HasProperty		NP → AP
causes		causedBy	Causes	RO:causes	
reveivesAction		givesAction	ReceivesAction		NP → VP
isTranslationOf	T, S				Term → Term
hasSubevent		isSubeventOf	HasSubevent		VP → NP,VP
hasFirstSubevent		isFirstSubeventOf	HasFirstSubevent		VP → NP,VP
hasLastSubevent		isLastSubeventOf	HasLastSubevent		
hasSynonym	T, S				Term → Term
hasAntonym	S		Antonym		Term → Term
isA			isA		NP → NP
isMotivatedByGoal			MotivatedByGoal		
desires			Desires		
hasA			HasA		NP → NP
isRelatedTo	T, R		RelatedTo		NP → NP
isSymbolOF		hasSymbolOf	SymbolOF		NP → NP
isUSedFor			UsedFor		NP → VP
isDependOn	T	isNecessaryFor		GFO:depend_on	Term → Term
isBornOn					NP → literal
isDestroyedOn					NP → literal

Table 4.6: ConceptOnto properties specification.

use. One of the main issues in any of the fields related to information retrieval is the accessibility of information both computationally and time wise. A field which we believe can have immense benefit from an easier data representation is "Question Answering". As we presented an approach to use RDF triples in this field Najmi et al. [2013], we have discussed that converting a question to a triple useable in SPARQL PrudHommeaux et al. [2008] is far less complicated than find the information online in any of the current SW repositories, considering the size of the current knowledge in required format. We believe the corner stone to our approach and similar approaches Lopez et al. [2005b] Mann [2002] Lopez et al. [2007] in this field is a thorough ontology which presents the possibility to convert different available and to be available knowledge bases to RDF.

Another research area which has gathered a lot of attention in the past few years is sentiment analysis. While there have been a few work specifically focused on Sentiment Analysis (as mentioned in Section 7.2) but on a higher perspective, bridging the gap between human understanding of emotions compared to machine understanding, is a more sophisticated topic which needs further research. The presentation of emotions in SW formats can be a good start in this direction. To do so, new ontologies with deeper relations to present different situations and scenarios can help this cause. While we do not claim that *ConceptOnto*, at this state, is providing the tools to present emotions, we believe it provides the necessary tools to present different emotions presentation in form of words.

While in the past few years search engine technologies had many major advances, the real technology behind these engines has stayed the same. Innovations such as *PageRank* Page et al. [1999] from Google has changed the perspective on finding valuable resources on the Web, but finding related contents to user inquiries is still mostly based on content similarities and closeness of concepts together. A

major issue with improving the quality of search engines is the nature of information on World Wide Web. In lack of traditional databases, a replacement tool which has the potential to present the information in more machine understandable way is by use of Semantic Web technologies. There are two approaches have been introduced to implement the aforementioned solution. The use of current infrastructure of WWW, use of HTML and similar taggings, has been the chosen approach for works in MicroFormats Khare and Çelik [2006], RDFa Adida and Birbeck [2008] and similar approaches. On the other hand, the general transformation of information presentation in SW formats is a harder approach which requires remake of the infrastructure for specific SW technologies. We believe the first step to achieve this goal is to provide an ontology to map general knowledge to SW format. *ConceptOnto*, as an ontology based on common sense which has been created by the purpose of representing general understanding of natural language, can be a useful tool for conversion and retrieval of this information.

CHAPTER 5: CONCEPTRDF

5.1 Introduction

Semantic Web paradigm, has gained a considerable amount of traction in recent years. On the other hand, the expansion of internet, without an agreed upon mechanism to organize and retrieve information has introduced different approaches for organizing and retrieving information. To simplify the retrieval process from the textual body of the Web, many research groups have introduced different knowledge bases which have gained popularity based-on the extent of information presented in them, the simplicity of information retrieval from them and the maintenance and support which has been provided for them.

An important issue which prohibits the flow of information to different knowledge bases is lack of consensus in their syntaxes and the diversity of their information presentation. One of the solutions for the first part, as many research groups have worked on, is to create a unified upper level ontology, which provides simplistic syntaxes and follows clear rules for presenting relationships and entities in any knowledge field. The other approach is to convert different knowledge bases to semantic Web format using different ontologies to add more functionalities which can make use of these knowledge bases in semantic Web field. In this paper we follow the second approach to present our methodology and steps for converting ConceptNet Liu and Singh [2004] Speer and Havasi [2013], one of the biggest common sense knowledge bases available, to RDF/XML Klyne and Carroll [2006] format. We believe that RDF model, as the cornerstone of Semantic Web, combined with other related technologies (such as SPARQL PrudHommeaux et al. [2008] for querying over RDF, OWL McGuinness et al. [2004] to create and unify ontologies) can be the technology to be used to integrate all the data available in different knowledge bases and create

a unified source of information. It is noteworthy that while we present RDF/XML format, it is a straightforward process to convert it to any of the other RDF formats such as TTL.

ConceptNet is one of the major knowledge-bases which has gained popularity due to its extensive knowledge and periodic updates. In addition to the mentioned benefits, it is also available for public use in CSV and JSON Crockford [2006] format and through its Web site and API. Converting this knowledge base to RDF has its unique challenges which arise from the complexity of its edges and deep hierarchy of the presented information in it. The feasibility and benefits of this conversion have been discussed previously in Grassi and Piazza [2011], but here we present the actual conversion steps and limitations. The work presented in this paper is an expansion of our previous work in ConceptOnto Najmi et al. [2014], in which we described the steps to create an upper ontology based on relations in ConceptNet.

In the following sections, first in section 5.2, we describe the structure of ConceptNet both in JSON and in CSV format and analyze different parts of information presented in it. The process and the methodology of the conversion is presented in section 5.3. This section also provides the main limitations of converting the data from JSON or CSV to RDF and provides solution for them to some extent (Subsection 5.3.1). We believe this conversion is beneficial for different use cases for researchers and normal users, which we have described in section 5.4.

5.2 ConceptNet structure

One way to see ConceptNet is as a graph in which each concept or assertion is a node and edges (in graph context and not in the context of ConceptNet, as described later) are relationships connecting them. There are two output formats available to download ConceptNet; one is a normal CSV file, outputting one line for each relation, separated with comma and tab. The second format is JSON which on

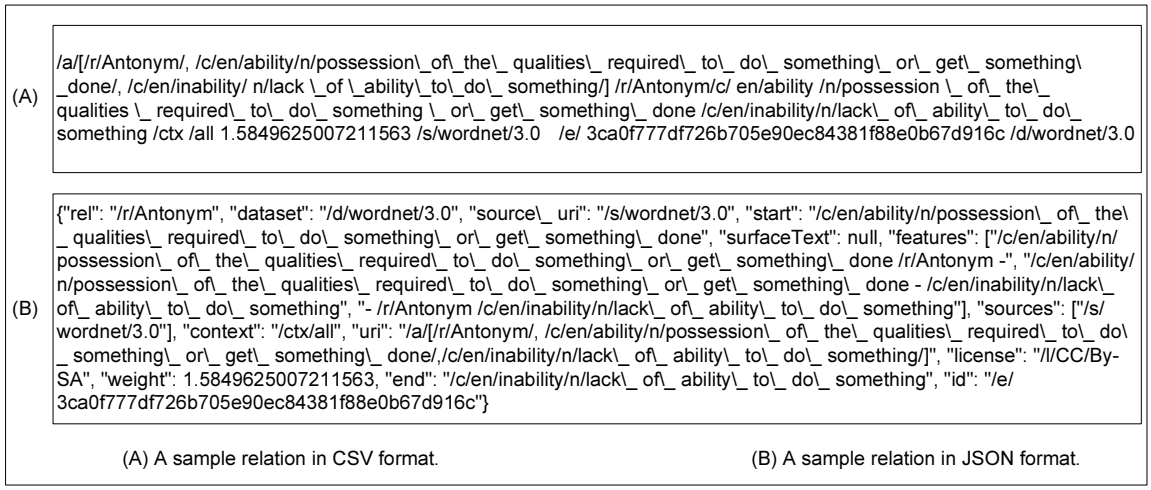

Figure 5.1: Sample line from ConceptNet data-set

the perspective of information presented is the same as CSV but with better human readability. Considering that the information presented in these two formats are practically the same, we describe the data-set with disregard to the presented format.

The main components of each line of information is a triple of subject, object and predicate. While in most cases the subject and the object are concepts, there are other cases which need clarification. In the following we first analyze a general concept relationship line in the data-set and later on we focus on special cases of information presentation in ConceptNet. The line (as shown in Figure 5.1.A) in the CSV file begins with the triple presented inside an assertion tag. The tag marks in ConceptNet URL are separated with a /. Each line in the ConceptNet data-set presents an edge. Each edge, identified by its ID following */e/* tag, consists of different parts as follows. Assertions, shown as */a/*, are the general knowledge presented in ConceptNet by marking the relation name, its beginning and the end of it. Every concept in the data-set follows a */c/* tag, presenting a term or a phrase. Relations are presented using */r/*. It is important to note that these relations are language independent and the concepts on both sides of them can be in different languages.

/d/ marks the data-set the edge has been extracted from, defined in ConceptNet context as a large source of knowledge which can be downloaded as a unit. Similar to the */d/* tag, there is */s/* to present the source of the information presented in the assertion. Currently ConceptNet has four source of knowledge. The first source is *contributor*, meaning an individual which has added the knowledge to the data-set. *Activity*, a knowledge collection task presented to a user to collect the knowledge (such as a game). *Rule*, an automatic rule to extract knowledge from different resources to the desirable format. And *site*, a knowledge base extracted from a Web site. The ctx tag shows the context of the relationship, e.g. */ctx/all*. Finally */and/* and */or/* marks the conjunctions and disjunctions of sources.

Concepts in ConceptNet can have up to four parts. As mentioned previously the first portion of any concept URI is */c/*. This tag normally follows by a two letter language mark, e.g. *en* for English, *ja* for Japanese. The third part is the concept itself which has been normalized by lemmatizer available in *conceptnet5.language* package as part of ConceptNet code repository [6]. The concept then follows by a letter marking the part of speech tagging of the word (e.g. *v* for verb or *n* for noun) and the last part of the URI is the sense of the concept, if available (generally available for the knowledge extracted from WordNet).

Description of the edge presented in Figure 5.1 in plain English would be "ability, as a noun, with the meaning of possession of the qualities required to do something or get something done, is antonym of inability, as a noun, meaning lack of ability to do something. The context of this edge is all (currently all the contexts are "all" in ConceptNet version 5, but we store the context in the case of compatibility with future versions). The source and the data-set of this knowledge is WordNet version 3.0. This edge has an ID of 3ca0f7...b67d916c and the weight of this edge

[6]Latest version at https://github.com/commonsense/conceptnet5

is 1.584...11563." The same line of information in JSON is presented in Figure 5.1. The JSON presentation stores the same information while providing better readability. JSON format guaranties a presentation which is both suitable for machines and humans. The down side of this presentation is the space consumption of all the extra tags and labels which makes processing this information for any text analyzer more difficult. A partial view of the final result, without the modifications implemented in later sections of this work is as follows.

```
<rdf:Description
rdf:about=
    "http://conceptnet5.media.mit.edu/
    web/c/en/ability">
    <COnto:hasDataset>wordnet</COnto:hasDataset>
    <COnto:hasSource>wordnet</COnto:hasSource>
    <COnto:hasContext>all</COnto:hasContext>
    <COnto:hasPOS>n</COnto:hasPOS>
    <COnto:hasSense>
    lack_of_ability_to_do_something
    </COnto:hasSense>
    <COnto:hasAntonym rdf:resource=
    "http://conceptnet5.media.mit.edu/
    web/c/en/inability"/>
</rdf:Description>
```

5.3 ConceptRDF conversion process

As mentioned in Section 5.2, ConceptNet data-set is available in JSON and CSV format. The JSON format has extra labels and tags to increase its readability for humans, which makes it larger in volume which in turn makes it computationally more expensive. For this reason, we decided to process the CSV files to generate the

RDF files. The process of converting the files to RDF is straightforward. This process consists of reading the files line by line, create tokens from them, parse the tokens and extract the information. The following algorithm shows the algorithm used for this process.

```
read file\;read line\;
separate by comma\;
\While{Tokens available}{
    find $/r/$;
    extract the relation as predicate;
    find the first concept;
    read the subject;
    If(subject){
        If (part of speech available)
            extract subject pos;
        If {sense available}
            extract sense\;
    If {object}{
        If (part of speech available)
            extract object pos;
        If {sense available}
            extract sense;
    extract context;
    extract data-set;
    extract source;
    extract weight;
    extract edge ID;
}
```

However, as much as this process is thorough, there are issues need addressing which we discuss in the next section. The result of this conversion, alongside the ConceptOnto ontology, is available for public use on our Web site [7] [8].

5.3.1 Limitations

The issued for converting ConceptNet to RDF are two folded. The first issue is with the logical disambiguation between formats like JSON and CSV to RDF. This issue is more fundamental and has more importance compared to the other problem. The second issue is more up to case by case basis related to specific relations which have further complexity than a normal triples.

Regarding the first problem, in the ConceptNet official blog it mentions [9] the main reason for preference of JSON over RDF:

> ConceptNet is not RDF
>
> I have sometimes been asked, given that ConceptNet is fundamentally a graph, why it isn't published in an RDF-based format. RDF is a very general representation of graph data, and yet it doesn't quite cover the information that ConceptNet needs to convey.
>
> Much of the information in ConceptNet is expressed as properties of its edges. In RDF, edges simply exist; they don't have properties. Additionally, all edges in RDF have to be considered incontrovertibly true, regardless of what source they came from, because they don't preserve any information about their sources.
>
> If you want to be able to talk about an edge, you need to reify it by turning it into a node and connecting it with a different kind of edge. This representation of ConceptNet would be difficult to create and even more difficult to work

[7] http://score.cs.wayne.edu/ConceptOnto.owl
[8] http://score.cs.wayne.edu/result
[9] https://github.com/commonsense/conceptnet5/wiki/Linked-Data-and-the-Semantic-Web

> with. Instead, representing the edges of ConceptNet as JSON structures (see JSON streams) makes the information in it easily accessible in a variety of programming languages.

The main point in this discussion is the complexity of presenting the diversity of information in edges in RDF form. This information can be divided into three parts, respectively subject, predicate and object. In creating an RDF triple for the main triple in each line, we can expand the information presented for the subject by creating multiple triples for the same subject. The problem arises for creating triples with objects of the original triple as subject. This limitation can be solved by assigning IDs to objects and create triples describing them separately. While this solution answers the problem in hand, it complicates the data-set to the extent that retrieving required information (using SPARQL, as described in section 5.4, or any other method) will be lengthly and complicated. To this extent, and based on the fact that the main piece of information presented for the object is the meaning of them (extracted from WordNet), we have decided against this approach.

Another approach to solve this problem is to separate the properties which are related to the subject of the triple from the ones related to the object of the triple. In the case of ConceptNet, we can do this by addition of one relation, to add a similar property to hasSense; namely replacing hasSense relation with subjectHasSense and objectHasSense. As a side note, it is necessary to mention that the only case that this addition is useful for is in converting the information extracted from WordNet, but because of the extendability of this knowledge-base (such as adding BabelNet synsets to the data-set) this is a useful addition and future approach. While this approach has better presentability for humans, considering the way SPARQL queries are presented, it complicates the creation and readability of the queries. We try to clarify this point further by an example. For the edge we presented in section 5.2

(ability hasAntonym inability), if we try to extract the sense of inability we run the following SPARQL query, in which the pivot point of the query is still the subject (ability) while we are looking for information on the object (inability).

```
SELECT ?sense
WHERE {
 :ability conto:objectHasSense ?sense;
 :ability conto:hasAntonym :inability
}
```

In this instance, and similar relations with transitive property, we can use this property to run the query for the opposite direction of the relation and run the same query for the object which simplifies the process to some extent; but for any other relation this complexity still exists. Another main limitation for converting JSON to RDF format (the second issue as discussed earlier) is the TranslationOf (isTranslationOf in ConceptOnto) property. In the context of ConceptNet most instances of this property have different triples as subject and object. The issue with this proposition is RDF does not have the capacity to inquire triples inside other triples. For example the following triple has two triples as its subject and object (dog IsA animal in English and Japanese).

(犬 IsA 動物) TranslationOf (dog IsA animal)

Analysis of instances of TranslationOf shows that the triples implemented in both object and subject of this relation are both presented in other lines. As any triple which exists in ConceptNet has an ID (edgeID), we can use these IDs to create new triple consist of subject edge ID, the relation isTranslationOf and the object triple edge ID. Using this approach the mentioned relationship changes to the following (edge IDs have been shortened for convenience of formatting)

```
/e/e1...5c1 isTranslationOf /e/5c6...0ea9
```

Another approach for this issue in RDF model is to use RDF reification. Reification is usually used to create statement describing another statement. In this case, we believe reification is a better option considering that having all the details in regard of the statement in one place is more self explanatory and makes the retrieval of the information easier. While this approach mostly solves the mentioned problem, there is still a lot of discussion Nguyen et al. [2014] on the use of reification and its complexity, which we consider to pursue for our future works.

5.4 Use cases

There are multiple use cases for Semantic Web data representation such as in Question Answering systems, Sentiment Analysis or any similar research topic in which the research tries to make sense of common sense knowledge which is hard to represent in any other format except an interconnected web of information. In this section we present two examples of the possible use cases of ConceptRDF.

For the first example we follow the directions in our previous work Najmi et al. [2013] to answer a simple question using ConceptRDF. In that work we mentioned the need for a large RDF knowledge base as a pre-requirement for our approach. While the approach has an acceptable performance on a simulated data-set, we believe that with the extended ConceptNet knowledge base and a possible addition of other similar data-sets it can improve exponentially. We try to provide an example which not only shows the usefulness of our approach, but also shows the limitation of current status of ConceptNet for the purpose of QA system. The question we consider is "Who is Bill Clinton?". The translation of this question in SPARQL is shown in the following.

```
SELECT ?object ?weight
WHERE {
 Bill_Clinton conto:IsA ?object;
```

```
 Bill_Clinton conto:hasWeight ?weight
} ORDER BY DESC(?weight)
```

It is noteworthy that this query orders the results by their edge weight, provide the results with the highest confidence as the first answer. Also we can see that lack of temporal information, in many cases can cause wrong information to be retrieved from the data. In this case the result of the query is the triple "*Bill_Clinton* isA *president_of_unite_state*" which with temporal consideration is not true (Bill Clinton is a former president of the Unites States would be a better answer). Finally the normalization to the concepts ConceptNet has changed United States to unite state which is not the same answer the user is looking for.

In the field of sentiment analysis, the knowledge presented in ConceptNet can be used as an intermediary information resource. Because the ConceptNet is a knowledge base based on common sense, it is rich on information regarding emotions, their states and causes. A logical follow through of the emotions can start by using IsA relation, extract the emotions by using the following query, then by using causes relation, extract the words and concepts which can be used to find relations between emotions, events, feelings or any other concept.

```
SELECT ?emotionNoun ?cause
WHERE {{
SELECT DISTINCT ?emotionNoun WHERE {
        ?subject conto:IsA emotion;
      }}
    OPTIONAL {
     ?cause conto:causes ?emotionNoun
      }
}
```

The result of this query returns emotions such as love, happiness and fear as result. Also in search of emotion without limitation of IsA relation, if there is an intelligence system it can find other details regarding emotions, for example people have emotions (people HasA emotion) or computers do not have emotions (computer NotHasA emotion).

CHAPTER 6: PRODUCT RANKING USING CUSTOMER REVIEWS

6.1 Introduction

Currently more than 85% of customers prefer online shopping to in-store shopping[10]. Major reasons for this preference are the convenience of online shopping compared to in-store shopping, and the reduced cost of storing and maintaining inventories for online retailers. Every year, the value of e-commerce trade increases exponentially. Subsequently, more categories of products are opening to customers via online shopping. The increasing use of Internet as a medium of shopping, provides an opportunity for users to express their opinions regarding their experience with products. These feedbacks show themselves on different factors on the Web as sales records, product ranks and reviews. Ghose and Ipeirotis [2006] argue that reviews, their assessment of the products and their quality are effective factors that impact the sale of the products. As the different rankings of products are useful for some buyers in deciding what to buy, there are many customers who need more in-depth insight about different products. The motivation of this work is to facilitate decision making for users by creating a new rank for each product using a combination of product reviews, review ranks and the products brand rank.

Figure 6.1 shows a sample review, based on user helpfulness votes, for the TV category. Using this sample and other highly voted reviews, we identify key points of online reviews (regarding their content and structure) as follows.

1. Best reviews consist of both positive and negative aspects of a product. In these kinds of reviews, it is not always possible to assign a positive or negative value to the complete review.

[10]http://www.safehomeproducts.com/shp2/news/news20071211.aspx

2. A single word in any position of a sentence can completely change the meaning and subjectivity of it, e.g., the word "but" at the beginning of the second sentence voids the negative weight of the first sentence.

3. For different features of the product the reviewers may use different terms. For example, the third paragraph of the sample review (Figure 6.1) talks about picture quality, but in the first sentence the word brightness refers to the same feature using a different terminology.

4. In some cases, while the reviewer's opinion is positive in general, the review may present negative opinions at first, but later expand the discussion by providing the reasoning on why the negative points are not valid.

Our proposed approach (CAPRA: a Comprehensive Approach to Product RAnking) starts by gathering the reviews in specific product categories. For each category, we select 10 products or more with similar major features. Major features are selected after performing various analyses on reviews, and product descriptions. For example, for TVs we look at 10 products with the same screen size. For each product, we store the products' specification in the database. The specifications consist of product aspects, manufacturer, product description, and its sales rank. Sales rank is later used in comparing the results of our product rank approach with its actual sale. Next, by using the iFrame address we go through the pages of reviews and store them in the review database. Thereafter, we use *part of speech tagging* and *stemming* on the reviews. The result will be useful for sentiment analysis and review ranking. As we have different aspects for each product, different customers may have different preferences. Finding these different aspects and assigning different weights to them is what comes in the next step. In this step, we also link sentences to aspects based on the words contained in these sentences. Next we omit the unrelated pieces of

information from the reviews by filtering sentences that do not correspond to the product or its aspects (features). For the next step, we assign sentiment values to these sentences consisting of Negative, Positive and Neutral. The step-wise results are used in obtaining a final product rank (both in the general case and according to user specific preferences). Note that based on user preferences, we can prioritize the product aspects as well. Finally the user is presented with succinct, and understandable search results which assist in finding faster, personalized, and more accurate products.

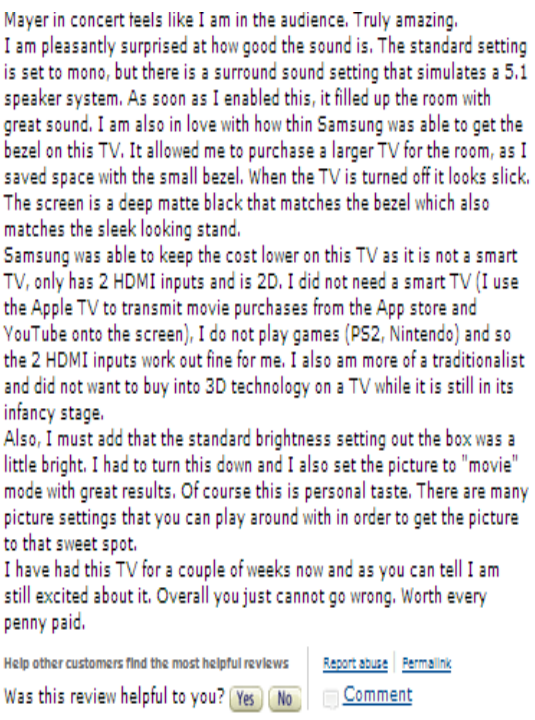
Mayer in concert feels like I am in the audience. Truly amazing.
I am pleasantly surprised at how good the sound is. The standard setting is set to mono, but there is a surround sound setting that simulates a 5.1 speaker system. As soon as I enabled this, it filled up the room with great sound. I am also in love with how thin Samsung was able to get the bezel on this TV. It allowed me to purchase a larger TV for the room, as I saved space with the small bezel. When the TV is turned off it looks slick. The screen is a deep matte black that matches the bezel which also matches the sleek looking stand.
Samsung was able to keep the cost lower on this TV as it is not a smart TV, only has 2 HDMI inputs and is 2D. I did not need a smart TV (I use the Apple TV to transmit movie purchases from the App store and YouTube onto the screen), I do not play games (PS2, Nintendo) and so the 2 HDMI inputs work out fine for me. I also am more of a traditionalist and did not want to buy into 3D technology on a TV while it is still in its infancy stage.
Also, I must add that the standard brightness setting out the box was a little bright. I had to turn this down and I also set the picture to "movie" mode with great results. Of course this is personal taste. There are many picture settings that you can play around with in order to get the picture to that sweet spot.
I have had this TV for a couple of weeks now and as you can tell I am still excited about it. Overall you just cannot go wrong. Worth every penny paid.
Help other customers find the most helpful reviews
Report abuse | Permalink
Was this review helpful to you? Yes No
Comment

Figure 6.1: Sample Review

Figure 6.2 shows the general architecture of CAPRA. Our main contributions are: 1) Creating and using "Brand Rank" as a preliminary rank for new product releases. 2) We provide both aspect ranks and average product ranks, based on general user opinions and users specific needs, and finally 3) To the best of our knowledge this is the first work that combines all the mentioned research fields, and creates a unified product rank.

The rest of the paper is organized as follows. In section 6.2 we present some aspects of our data-set preparation process. Sentiment Analysis (Section 6.3) describes our approach for analysing the reviews and assigning negative, positive or neutral polarity to them. In Section 6.4 we identify different aspects of the products and introduce our approach to rank them. Section 6.5 outlines two different approaches to brand ranking. In Section 6.6 we describe our approach to product usefulness

analysis. Section 6.7 summarizes all the previous sections to create a unified product rank. Experiments and results, Section 7.5, shows the result of our experiments and comparison of CAPRA to some of the similar works. Also in each section we first present the related work to that specific field to familiarize the reader to some of the previous work in that respective field.

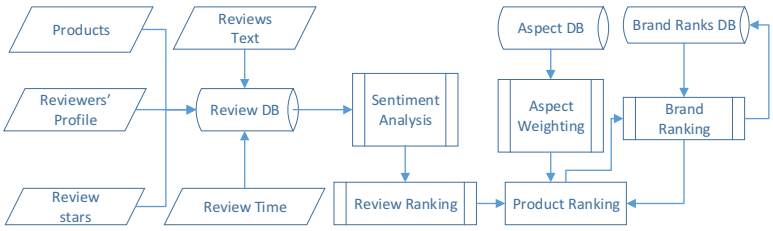

Figure 6.2: Product Ranking Process

6.2 Data-set Preparation

After analyzing different resources (considering the main criteria we are looking for in the reviews; mainly descriptiveness of reviews, range of reviews from good to bad and different measures to review user experiences with products) we decided to use data-set gathered from Amazon. The positive points about the Amazon data-set are:

1. Number of reviews: On average, we have a large number of reviews for each product we considered in our data-set.

2. The Star system: The star rank each reviewer gives to the products shows the overall opinion of the reviewer of the product.

3. Review usefulness ranks: Users, when deciding what to buy go through the reviews in Amazon and sometimes vote on their usefulness. These ranks are useful in defining a baseline for finding the most useful reviews.

4. Products sales rank: Amazon provides us with a sales rank number showing the sale record of products in each category.

5. Reviewer public profiles: Determining the reviewer's history can assist in determining their interests, previous reviews, etc.

6. Number of replies to each review: Some of the reviews have replies from other users or sometimes from the producers of a product.

While providing general users feedback on products, there are some concerns regarding Amazon reviews which we need to identify. First, for some of the reviews, portions of them do not address the product and mostly talks about the conditions for buying the product, e.g., the occasion or the time of the event. Second, the timeline of the reviews start at the release of the product and continue till the product becomes discontinued, so the user experiences are not the same over time. The third problem we have to consider is that Web sites like Amazon etc. allow different sellers to sell the same product or have different colors of the same products sold separately as a unique product. This issue not only causes duplicate products, but also provides a situation in which users repeat their reviews for different products. For the purpose of this paper, we have to remove these duplicate products and reviews. There are two general solutions for this problem. One approach runs similarity metrics on reviews using factors like bi-grams and the second approach uses reviews TF-IDF to compare their similarity. In this work, we consider reviews' bi-grams and if we find more than 80% similarity among them, we consider these products or the reviews to be duplicates and we discard them.

Another point to mention is that we generally divide the products into two general categories, *content driven* and *use driven* products. When deciding to buy content driven products, users generally focus more on the content compared to phys-

ical attributes of the products. This approach changes users' expectations from both the products and their reviews. Hence we need to develop different processes for extracting useful information from reviews for these two types of products. The *content driven* products consist of "Books", "Music" and "Video games" etc., while the *use driven* products are the physical products for everyday use like TVs or cameras. In this paper, while the focus is on *use driven* products, in different sections we point out some difference among these two categories. For instance, analyzing *content driven* products requires modifications to each set of features we have considered in this work. We leave further work in this regard to our future work.

The process of gathering the reviews from Amazon starts by finding the products. Then, as Amazon doesn't provide users with the review texts, we scrape the Web pages to gather their text, date, reviewer info and star value of each review. The next steps focus on processing the reviews to prepare for different analysis on them. The changes consist of tokenizing, part of speech tagging and stemming. For this purpose, we use Stanford Natural Language Processing (Stanford [2005]) Toolkit which in our assessment provides satisfactory results with acceptable efficiency (Ge and Song [2010]).

The last addition to the data-set is the time token. Time of the reviews is an important factor both in comparison to the other reviews and to the release time of the products. Considering that over time newer products in the same category come out, many users review the products based on their experience with newer products. Also, in many cases, when a product has a known problem, in the newer version (of the same product), while keeping the same general specifications, companies fix the issue. While we can not provision this issue in the current work, it is an important issue to consider for future versions of our work.

Finally, because of the diversity of product categories and their reviews, we limit our product categories to HDTVs and cameras. Further discussion about our data-set and experiments is deferred until Section 7.5. We tried to gather products with review numbers in different ranges (high number of reviews, more than 100, average number, between 10 and 70, and low number of reviews, under 10) in order to consider a broader range of products.

6.3 Sentiment Analysis

Sentiment Analysis (SA) is the process of assigning polarity and sentiment values to words, sentences and the whole body of text. In recent years there has been a lot of work regarding the subject of Sentiment Analysis. Generally, works in this field have two main approaches. First, some works focus on assigning a positive or negative sentiment to a body of text, as a whole (examples include Pang and Lee [2004]; Taboada et al. [2011]; Turney [2002], etc.). While this approach can be useful for general text, reviews are more complicated (as shown in Section 6.1). In contrast, the second approach covers the text on a sentence-by-sentence basis (Narayanan et al. [2009], Zhang et al. [2011a]). Our premise is that, separate sections of reviews talk about positive and negative points of a product in accordance with each other, so it is logical to not consider the text as a whole and treat each sentence as a separate body of text regarding the sentiment. Moreover, for the later parts of the process, we need to consider different features of the product separately. As different portions of a review may address different aspects with different sentiments, hence we define a different approach for analysis. We divide reviews into its different aspects, and by summing up the sentiment values of each aspect, we gather the opinion of each review on the subject.

On the technical front, the SA problem is also divided into two major classes. In the first approach (Lexicon Based (Taboada et al. [2011],Ding et al. [2008])) the

focus is on creating lexicons of words, and assessing their "polarity". Polarity is defined as the orientation of the word, sentence or body of the text regarding its sentiment. Some works also store the information regarding part-of-speech taggings to be more specific about different scenarios. While this approach can be sufficient for direct and simple sentences, with the addition of complexities of natural language, it has difficulties understanding the polarity. To clarify this problem we present two examples in the following.

1. *The case of "But clause"*: In most cases, the keyword "but" voids the first half of the sentence and the "but clause" can be translated alone. Similarly, there are other cases which either negate the first portion of the sentence or put more emphasis on it, e.g., "This camera size is big, but with its good design, it can easily be handled.", "I not only like the picture quality of this camera, but also its size","This camera doesn't have a VGA port, but with internal WiFi you won't even need it". We can clearly see that identifying the differences between the sentiments of these sentences is not possible by only using the lexicon approach.

2. *The case of "Negation"*: In some cases, negation can make a positive polarity negative with the same weight. In other cases, however, it can change the polarity of the sentence but with less weight than the positive sentence. Moreover, in other cases negation can be used with intensifiers, which makes the behavior of the sentence unpredictable; i.e. it can decrease the weight of the polarity or completely change it with different weights. For instance, "Nobody says this is a good camera", "This camera is not very great", "In short, it is not a good camera". Similar to the previous case, differentiating between the polarities of such sentences is not possible only with the lexicon-based approach.

The main difference in lexicon based approaches is how they treat cases similar to the above mentioned examples. The general approach is to use pattern recognition

to analyze these cases. While the rules and patterns introduced in these works increases their accuracy to some extent, further complexities of natural language have promoted the introduction of a second approach. The text classification approach (Turney [2002],Narayanan et al. [2009]) uses classification methods to analyze and classify sentences' polarity as a whole. Normally these approaches make use of a lexicon (in some cases to be used as seed to expand and in cases as one of the classification features). Similar to these approaches to SA there are others which focus on snippets or aspect based SA (Sauper et al. [2011]). The literature shows that classification approaches, specifically in more complicated texts and when implemented to specific domains has a better performance compared to lexicon based approaches.

Before going into details of our text classification approach, we expand on the complexity of reviews in the following. Our goal is to provide a more in depth analysis of reviews regarding the complexity of natural language. To understand the complexities of product reviews, we used manual annotation to classify a set of reviews in different categories of products. The aim is to find out if using a simple analyzer would suffice the needs of our sentiment analysis. Table 6.7 shows the results of this annotation. We separated the sentences into three main classes: Neutral, Positive and Negative. Moreover, we divided the positive class to three sub classes. The sentences can be (1) Simple positive; using simple terms to show positive opinion, e.g., "This product is amazing". (2) Negation; to negate a negative in the sentence, e.g., "This functionality is not bad at all". (3) Complex; which depend on the readers' knowledge to infer positive or negative meaning of a sentence, e.g., "This is like going from Blackberry to Iphone". The results shown in Table 6.7 depict that around 89% We summarize the main issues as follows:

1. In a few cases, a negative sentence was followed by a neutral sentence. This neutral sentence was an answer to the point in the previous sentence and made it

of positive sentences are simple. Thus, we can conclude that using simple analysis and negation in our classifier, we can achieve an accuracy level close to 90%. Our analysis of online reviews reveals both structural and semantic complexities that are inherent to natural language processing.

Sentence Classes	%age	Positive Classes	%age
Positive	27.2	Simple	88.2
Negative	14.08	Negated	4.2
Neutral	58.6	Complex	7.5

Table 6.7: Manual annotation result of a sample data-set

positive or vice versa. For example, "Unfortunately this product has just one HDMI port. But if you use a gaming console, that's enough."

2. In one paragraph each sentence has neutral meaning separately but the overall theme in the paragraph has general positive or negative meaning.

3. The complication of sentences can range from a simple idioms, to comparison of two unrelated products, to an expression which does not have semantic meaning at first glance. For example "Whites are white and blacks are black" while looking completely neutral, in the TV category, this is essentially a positive attribute of the picture, and means that colors are alive and natural.

Thus, we can safely conclude that in the best case, if the system identifies all the complex sentences, clearly analyzing them would be semantically near impossible. This is mainly due to the lack of semantic knowledge on our side to consider all the different terms of the natural language. So even if we use a more complex approach to this problem, the final accuracy would not drastically improve compared to the simple approach. Also the semantic knowledge in different categories are at least slightly different from each other which, without modification, can affect the result negatively and void the cost of the process.

In light of the above discussion, we divide the SA features into:

- Structural features: These features focus on the structure of sentences, e.g, negation.
- Semantic features: Some words have gained additional and different meanings over time. This group of features focus on this concept, e.g., smileys.
- Polarity features: Polarity features of words and their "pre" and "post" contexts.
- Numerical features: Numbers mentioned in the sentence, e.g., 23MP.
- Review features: Review features which affect the sentiment value of sentences, e.g., number of stars of the review.

Considering the polarity of words in sentences, we differentiate between "modifiers"; words which modify the polarity of a sentence, e.g., even though, and "intensifiers"; words which intensify the polarity of sentences, e.g., very. Although most of these words are grammatically adverbs and adjectives, we have to expand the list to other parts of speech (POS) as well. For example, *nothing* as a noun is generally used as a modifier. Thorough analysis and discussion of the use of modifiers and intensifiers can be found in Polanyi and Zaenen [2006]. In the field of SA there are works focusing on these words as the primary approach and follow a manual annotation of words to assign values to them (Kennedy and Inkpen [2006]; Nadali et al. [2010]). Most of the more comprehensive approaches using this method use different compilations of the work presented in Quirk and Crystal [1985]. For this work we follow a similar approach, i.e., assigning manual weights to these words which are more used in reviews compared to other parts of literature. The resulting data-set consists of 76 words following Table 6.8's structure. We start by annotating different phrases,

Word	Weight	POS
Very	2	adj
Barely	0.5	adv
Not	-1	adv
Nothing	-1.5	noun

Table 6.8: Shifters Table Sample

from term level to whole sentence, as positive or negative. To prepare a lexicon of subjective terms we expand the SentiWordNet corpus Esuli and Sebastiani [2006], to make better sense of the phrases. SentiWordNet, itself, expands WordNet Miller [1995]; Miller et al. [1990] by assigning negative and positive values to words between 0 and 1 respectively. In general, words

can be *positive*, *negative*, *both* or *neutral*. An example of positive words is 'good' as in "This camera has a good picture quality". Negative words like 'negative' as in "The most negative aspect of this camera is its body size". A word which has both polarities like 'funny' as positive in "That is a very funny movie" or as negative in "The button looks funny on the TV". A neutral word is a word which does not have a specific polarity. This category consists of all the nouns or aspects of products. In our data set we make use of the pre assigned negative or positive number of a word. The neutral words are the words which have 0 negative or positive polarity. For the other three categories we assign a threshold as show in Equation 6.1 to assign positive and negative polarity to those words.

$$W_{pol} = \begin{cases} Positive & \text{if } W_p - W_n > \theta \\ Negative & \text{if } W_n - W_p > \theta \\ Neutral & \text{if } |W_p - W_n| < \theta \end{cases} \tag{6.1}$$

Where Wp is the positive polarity of the word, Wn is the negative polarity and θ is our assigned threshold. Wpol holds the final polarity of the word. For example, by assigning θ as .15, for the word "living" with the positivity of .5 and negativity of .125 will result in assigning positive sentiment to the word.

Word Features	In subject	Sentence Features
Words' letter case	Modification Features	Strong/weaksubj in current sentence
Word Part-of-speech	Proceeded by adverb	Strong/weaksubj in previous sentence
Word context	Proceeded by intensifier	Strong/weaksubj in next sentence
Prior polarity	is intensifier	Cardinal numbers in sentence
Reliability class	Modifies strongsubj	Pronoun in sentence
Review Features	Modifies weaksubj	Modal in sentence
Product Category	Modified by strongsubj	Adjectives in sentence
Review star value	Modified by weaksubj	Adverbs in sentence
Structure Features	Proceeded by adjective	Product aspects in sentence
In copular		Shifters in sentence
In passive		

Table 6.9: Neutral sentence classifier features

Our approach to SA consists of two phases. In the first phase, we solve the problem of non-neutral terms that appear in neutral sentences. As Table 6.7 shows we have around 59% neutral sentences in our corpus which if not identified, because of the non-neutral terms in them, can effect the general positive and negative weights of reviews. The base classifier classifies the sentences based on the class of terms which can assign a sentiment other than neutral to unrelated sentences. To address this issue, we will use the result of the next section to remove unrelated sentences from our data-set based on the aspects in each sentence. In short, we separate the non-neutral sentences from neutral ones. The first portion, neutrality classifier, considers 27 features. These features are shown in Table 6.9.

The second step for the approach is polarity classification, considering that we have already removed the neutral sentences. This classifier focuses on three classes of features: Word feature, polarity features and sentence features. These features are shown in Table 6.10. Some of the features we used in this section have been used previously in related literature (e.g., Wilson et al. [2009]). While these approaches

are similar, we have tailored different parts of the approach to better suit our needs. A comparison with existing approaches is presented in Section 7.5.

Word Features
word's pre defined polarity: positive, negative, both, neutral
Polarity Features
negated: binary
negated subject: binary
modifies polarity: positive, negative, neutral, both
modified by polarity: positive, negative, neutral, both
conj polarity: positive, negative, neutral, both
general polarity shifter: positive, negative, very positive, very negative
Sentence Features
sentence main aspect
emoticons in the text

Table 6.10: Polarity classifier features

6.4 Product Aspect Analyzer

Aspect Analyzing (AA) is defined as extracting and analyzing products aspects and features. The subject of AA/"Topic Detection" has gained little attention in the literature, and most of the works function at the document level (Stoyanov and Cardie [2008]; Wang et al. [2007b]), as opposed to sentence level (focus of this work). NIST sponsored "Topic Detection and Tracking"(TDT)[11] research track is one of the very few research tracks specifically targeted to this subject, i.e., focused on providing tools for English language speakers to access, correlate, and interpret multilingual sources of real-time information. In recent years, other than the general topic detection approaches (Joy and Leela [2013]), more focus has been given to specialized topic detection in specific fields, e.g. health care, etc. (Lu et al. [2013]).

Topic detection at the sentence level is normally used in works which need to analyze documents at a deeper level than only the general subject of documents

[11]http://www.itl.nist.gov/iad/mig//tests/tdt/

like review analysis. Sentence level topic detection, or "Aspect Analyzing", while harder in some aspects (limit of information in a single sentence compared to the whole body of text), is less complicated from other points of view (no need to post process and can judge each sentence independently). The main difference between sentence level and document level subject analysis is that in sentence level analysis, we have a limited set of words and sentences, and there is no given list of topics that we can map the sentences to. The former stops us from following the most common practices in this field (which is using classification (Wiener et al. [1995])). Similarly, for lack of topics, we need to make a list of aspects related to different categories of the products. Moreover, each user has different priorities while looking at and/or buying a product. While these priorities can be substantially different, most customers in different categories of products are looking for specific features in their product. Thus, to analyze the reviews and break the sentences based on different categories, we need to gather the different aspects for each specific category. Therefore, instead of ranking a product as a whole, we break the product according to its different aspects. One probable solution to extract aspects is to parse the reviews to find the group of nouns and consider them as aspects of the product, based on their frequency (Hu and Liu [2004]). While this approach finds all the product aspects, it also adds considerable noise in the process, which usually does not reflect the products or shoppers' opinions about them. For example, "I got this product from XYZ store, and as you know it's very expensive in there". If we follow the mentioned solution (of grouping nouns), the approach will consider XYZ as an aspect of the product, which can have a high frequency, if the store is a big distributor of the product. And based on Sentiment Analysis (Section 6.3), this sentence has negative polarity (the word *expensive* is negative and *very* is an intensifier which increases the negative polarity). To avoid this problem, in our work we also consider the product description as another source

of aspects, as this body of text normally describes the important aspects of products focusing on the aspects that companies consider vital for their sales.

To extract aspects from the aforementioned resources, we use Term Frequency (TF) on groups of nouns using pattern matching. Part of these patterns are complete sentences with specific structures which in all cases have an accompanying adjective, and in some other cases we have numeric lists where each item is just a group or a single word. Note that our pattern list is not exhaustive, and by expanding the data-set (and increasing the frequency of each aspect), we can extract all the common aspects from reviews. In each category of products there are different words which directly or indirectly point to the same main aspect of a product. To consider these similar aspects as one and decrease the redundancy, for each pair of extracted aspects we measure their similarity. The solution we chose for this purpose makes use of adjectives in sentences. Our analysis shows it is common that in each category same adjectives are used to describe similar aspects. We ran a small experiment to prove this point by analyzing a small set of sentences from reviews of the same category. The results show that in 74% of cases this theory is correct.

Following from the above mentioned point, we keep the adjectives from different aspects and compare them together. If 85% of similar adjectives are used in comparison of two aspects, we consider the two aspects the same and store them. The result is a list of products which considers the aspect similarities. We have to note that even though our process is designed specifically to extract aspects, and compared to similar works in the literature, our process of extracting all group of nouns is better since it has less noise but we still need to improve the computational complexity of the process and the list of aspects needs to be refined. While the weighting process described later in this section will create a sorted list to be presented to the users; which in return acts a natural filter for aspects. To increase the precision of the aspect

list we use expert opinions. This is specifically possible because we have limited our data-set to two categories. Nevertheless, for our future work we intend to reduce the noise, and eliminate the need for expert opinions. Since each category of products has similar aspects, we create an XML file for each category. A sample XML file for the camera category is shown in the following.

```
<Category>
    Camera
</Category>
<Aspect>
    <Key term>
        Resolution
    </Key term>
    <Equal Terms>
        size, Picture size
    </Equal Terms>
    <Regex>
        [1-52]MP
        [1-52]Megapixel
        [1-52] Megapixel
    </Regex>
</Aspect>
```

In addition, there are three aspects that we consider for all the products i.e. "Delivery Time", "Packaging" and "Customer Support". The aspects files also store the synonyms for each aspect and the terms which can be used to describe these aspects. For example, for the feature "Refresh Rate" for a TV, the term in the title is described by a number followed by Hz. We store the regular expression of the term and match the pattern with numbers. Another part of the xml file stores the related terms to each aspect, e.g., for "Refresh Rate" we store terms "motion","blur" and "picture quality". Each product in a given category should be compared to other products in the same category with similar base aspects. For example, in the process of purchasing a TV one can consider the size and the technology of the TV as its

main features, so the final decision will be made based on these criteria, compared to other products with similar features. We can consider another user whose main criteria for buying a TV is its size and the price range. Our goal is to find a set of criteria for each category of products which are the most important for general users. After thorough analysis of different categories we believe the main criteria of each product can be found in the product title and its price. In this respect, it is noteworthy that the price range for each set of products can be different based on the product category. To consider this difference we create the margin automatically using the population standard deviation between product prices in a given category. For example if we have three products in one category, with respective prices of 300, 400 and 450, the margin used in this category is 62.

Finally, after extracting the aspects, using main priorities of the users, we assign different weights to different aspects. For example, picture quality in a TV is more important than its applications' execution speed. To compute the weight of each aspect we use Equation 6.2. Also, based on our experiments we divided the weights of the negative results for each category from the positive ones and give them different weights respectively. For example, if for aspect one of a category we have 6 positive and 4 negative polarity in the reviews and we have 15 positive and 10 negative polarity in all the aspects of the category, the result of the equation for this example is calculated using the following equation. After calculating all the weights, we normalize the weighs so the summation of all weights in each category equals to 1.

$$\alpha \times 6/15 + (1 - \alpha) \times 4/10$$

$$Aw = \alpha \times PA/P + (1 - \alpha) \times NA/N \tag{6.2}$$

where Aw is the weight of each aspect, PA and NA are all the positive and negative repeats of the aspects, P and N are all the positive and negative sentiments of all the aspects respectively.

6.5 Brand Ranking

Brand ranking or brand popularity analysis is an old research topic dating back to the late 50s (Pessemier [1959]). Most of the works to date show that brand rank is more related to brand loyalty and brand popularity than brand quality (Dawar and Parker [1994]; Traylor [1981]). There are two main reasons for this. First, from a business perspective, what is important is how companies sell their products and product quality rank is not as effective on sale records as other ranks. Second, the access to information to measure brand value from a quality perspective, compared to brand popularity and loyalty, is hard to come by. While these approaches with focus on business ventures compared to end users are comprehensive, we have access to user reviews as a base of user opinions about brands and we try to create brand ranks for end users compared to businesses. We gather the brand ranks based on product quality from reviews to create a unified brand rank for each brand in each category.

Whenever a new product is released it takes a while for user reviews to start showing up. It may be possible to find some reviews in blogs, etc. but in e-commerce Web sites in general (like Amazon), as there is no user experience, there are no reviews for the product. In such a scenario, the knowledge base regarding this specific product is very small. While we don't have much information about the product, based on the history of the producer we can predict the popularity and quality of the product (which our analysis also proves). For example, in case of Samsung TVs, the star value of the products are 47% four and half stars, 38% four stars and around 10% three and half stars. As this example shows most of the product reviews on same brands

assign similar star values to the products. As more reviews come out, the weight of the brand rank decreases till we have enough reviews to completely nullify its effect.

Some studies have shown that brand quality can be generally measured regardless of the category of product from consumer's perspective (Aaker and Keller [1990]). This assumption (as shown in Aaker and Keller [1990]) has two main requirements regarding the extension of the brand to different categories. One condition is the concept of "fit" category which means the previous product line of the brand has similarities to the new line. The second key point is the difficulty of extension for the company which directly relates to how similar the product categories are. These studies show that if the new category of products are too similar to the previous one, the consumers do not assign a high quality to the products. Assuming validity of these points (in all categories), and since we do not have valid similarity metrics between categories, and that we want to rank brands from a machine perspective, we separate brand ranks for different categories. For example, it is possible that a brand which produces Cameras also produces TVs, but as the quality of the products compared to other brands in the same category can be different, the ratings of these two should also be different. In the general case, the brand rank weight for content oriented products sets to zero. In the following, we discuss two approaches to calculate brand ranks. The first approach uses star and review ranks to calculate the brand rank, and the second approach makes use of PageRank (Brin and Page [1998]) to calculate the brand ranks.

Since we do not have any product ranks in CAPRA in the beginning, we start by using the average stars for each category brand from Amazon. When we rank each product, the average rank of the product will consequently change the rank of the brands. Equation 6.3 shows CAPRA's brand rating process. The first portion of the equation creates the ratio of non-ranked products in the category and multiplies it

by their star rank. This gives us an average value for the rank of the products which have not been ranked based on the reviews. The second part of the equation first finds a ratio of ranked products and multiply it to their rank to calculate the rank of the brand of already ranked products. The final score is normalized summation of these two scores, in the range of [0,1].

$$Br_c = \frac{\frac{(TP_c - TRP_c)}{TP_c} \times avg(SV_(nRPCB) + TRP_c}{TP_c \times avg(RB_c)} \tag{6.3}$$

where Br is the brand rank of products, c is the category of the product, TP_c is the total number of products in the category c, TRP_c is the total number of ranked products in the category c, SV is the star value of the product, and nRPCB is the number of products from this specific brand which have not yet been ranked.

The second approach to the problem of ranking products makes use of the Page-rank algorithm (Brin and Page [1998]) to rank brands. PageRank, in its original form, uses links between pages to approximate the value of each page. Formally, the page-rank equation in described as: "We assume page A has pages T1...Tn which point to it (i.e., are citations). The parameter d is a damping factor which can be set between 0 and 1. We usually set d to 0.85. Also C(A) is defined as the number of links going out of page A."

$$PR_(A) = (1 - d) + d(\frac{PR(T_1)}{C(T_1)} + ... + \frac{PR(T_n)}{C(T_n)}) \tag{6.4}$$

In CAPRA, we implement the Page-Rank algorithm similarly to Equation 6.4. We replace page A with brand A and pages pointing to it are replaced with reviews which mention the brand A. C(A) in our case will also be replaced with the number of brands mentioned in reviews of products from brand A. The performance of this

approach is mainly related to how many of the reviewers consider mentioning other products as important for the value of their review. Based on our analysis, current reviews on Amazon do not have that many relation points to other reviews. The reviews are written mainly by the end users of the products which do not have the experience of using similar products. Hence, using this approach for brand ranking is not as useful as the previous approach. For our future work we intend to expand to reviews from the Web, and find reviews from experts in each field (which can provide information and comparison on similar products from different brands).

6.6 Review Usefulness Analysis

The 'usefulness' of a review may vary from one user to another. The concept of usefulness of reviews is effected by the reader's perspective of the product and his/her approach on product selection. Also the writing of the review, its tone, the words a writer uses, and all the other details which are not measurable using machine learned approaches may alter the usefulness. This is why even if we create a complete classifying function with 100% accuracy from one person's perspective, there is no guarantee that someone else will have the same point of view and accepts a review as useful. Thus, we meditate on general usefulness of the reviews.

We believe that just considering the usefulness votes of users for each review is not accurate enough to be considered the only factor regarding the usefulness of the reviews as mentioned in Liu et al. [2007]:

- Imbalance Vote Bias: Users have a tendency to value other reviews as helpful even in cases when they are not really helpful.
- Winner Circle Bias: Users generally vote reviews which already have positive votes as helpful compared to reviews which have not gathered as much positive feedback from other users.

- Early Bird Bias: Early reviews of products generally get more positive votes as they have more views compared to more recent reviews.

In addition to the above mentioned problems, we have found that some reviews are not actually related to the product itself, but are based on user experiences. Such judgmental reviews usually have little or no effect on end-users, and just effect the rank of the products. For example, late delivery can be a reason for a reviewer to give one star to a product, while it does not say anything about the product itself. Researches show that users normally review a product when they are extremely happy or extremely angry with a product (Hu et al. [2009]). As a big number of reviews follows this "J shape" graph; meaning the highest number of reviews are either one or one and half stars or four and half or five stars (extremely dissatisfied or extremely satisfied), we cannot assume that the star values can completely be trusted as the review value. Thus, in the following we show how our system ranks the reviews and find a more accurate ranking.

To rank a product we first need to assign scores to the reviews. As mentioned previously, most existing works assign a positive or negative value to indicate reviews' helpfulness. We use Machine Learning Regression to assign a score to reviews. Another approach to this issue would be machine learned ranking. Use of ranking, while at the final step would create a clear ranking of reviews, needs repetitive ranking with addition of more reviews to the data-set, which in turn increases complexity and redundancy of the approach.

Support Vector Regression (SVR) is a widely used regression method for analyze helpfulness of reviews. While we do not differ between helpfulness and usefulness on higher levels for users, we propose that helpfulness analysis is trying to measure how helpful reviews are from end users' perspective. On the other hand, usefulness

analysis targets how useful reviews are for machine analysis and product ranking. Kim et al. [2006] present an approach to analyze reviews' helpfulness using SVR. The contribution of this work is not only the use of regression but also analyzing different set of features which have the best performance in this field. Their analysis shows that the best set of features consists of unigram, length, and star values of the reviews. Considering the differences we mentioned between our works, review usefulness analysis, the aforementioned work, and review helpfulness analysis, we create a separate set of features which we believe are more related and applicable for our purpose. For the kernel, Radial Basis Function (RBF), and other settings of SVR machine, we follow Kim et al. [2006] as new settings require more thorough analysis and focus on this research topic. For the purpose of training our regression, we use a training set of assigned values gathered from manually scored reviews. We use 3 set of scores from 3 different users trained to focus on important aspects reviews targeted for machine readability.

We consider two categories of features, a set of features which shows readers' point of view on how useful reviews are, for example usefulness votes of the review. The other set of features are the ones which effect the usefulness of the reviews measurable by machine, i.e. sum of aspects. Analyzing different reviews which have the highest usefulness from different Web-sites plus the related works in this field shows that the following factors are the most decisive on review usefulness:

- Length: The number of words in objective sentences is a good measure on usefulness of the reviews. A longer review usually provides more information to the users and talk about more aspects of the products (either positive or negative).

- Reviewer average rate: Each reviewer has a history of other reviews. This history can be considered as history of the reviewer's reviews usefulness based on users' votes on previous reviews.

- Sum of sentiments: This feature is the total number of sentiments that has been discussed in the review (Following Equation 6.5).

$$SoS = \frac{(Ss_P + Ss_N)}{SS} \tag{6.5}$$

  where SoS is sum of sentiments, Ssp and Ssn are sum of positive and negative sentiments respectively, and SS is sum of all sentences in the review.

- Star value: The star rank that a reviewer assigns to a product can show how useful the review is. More extreme ranks, specially one star, can show that the reviewer is biased towards the product.

- Sum of aspects: We take into consideration the result of our Aspect Analysis for each review; computed as the ratio of total number of aspects in the review to the total number of aspects for the product category.

- Time of the review with respect to the release date and current date (Following Equation 6.6).

$$TT = e^{\frac{T_r - T_l}{T_c - T_l}} \tag{6.6}$$

  where Tl, Tr and Tc are the release time of the product, the time of the review and the current time respectively.

- Spelling mistakes: We measured the number of spelling mistakes within each review using Google spell corrector [12], and we normalized the number by dividing it to the length of the review (in characters).

- Review replies: Some reviews based on their popularity have replies. This feature stores the number of replies to the review.

- Usefulness votes: The usefulness votes based on the other users' opinions.

- Reviewer's badges: For some reviewers, Amazon assigns badges, based-on their history or their performance as a reviewer. These badges include *#1 Reviewer*, *Top 10 Reviewer*, *Top 50 Reviewer*, *Top 500 Reviewer*, *Top 1000 Reviewer*, *Hall Of Fame Reviewer*, *Real Name Author*, *Artist*, *Manufacturer*, *Vine Voice*, etc. There are a few more badges which are not effective on user reviews quality such as *2008 Holiday Team*. We store each of these badges as a boolean value in the data-set.

- Verified Purchase: These reviews are done by users who have bought the items from Amazon. This item uses as a factor which shows the validity of the review considering the user has really bought and used the item.

For all the features, we also run a simple standard transformation to normalize and scale them to [-1,1] values (as suggested in Hsu et al. Hsu et al. [2003] to improve the performance of the SVM).

6.7 Product Ranking

The last step of our work focuses on ranking each product among similar products in its category. This entails analyzing product reviews, breaking them down

[12] A simple Java interface for the API available in https://code.google.com/p/google-api-spelling-java/

and creating a ranked list of products based-on different aspects. In this regard, some works (e.g. Zhang et al. [2010]) create a manual list of product aspects which are of importance for users. Then text mining techniques run on the reviews to identify subjective and comparative sentences. With this information, a graph of product aspect rankings is created. While this work is similar to our approach it has some key differences. First, the mentioned work (Zhang et al. [2010]) mainly focuses on comparative sentences to compare the products and rank them. In real world data-sets the number of comparative sentences is highly limited (an average less than 1 comparative sentence per review based on our analysis) which decreases the performance of this approach immensely. In contrast, we analyze any sentence available in the reviews and specifically focus on non-neutral sentences for further analysis. Second, we add brand ranking as one of the main features for product ranking which is very effective for new products or any product which does not have as many reviews as the other products in the category. Third, unlike Zhang et al. [2010] we analyze the review usefulness to filter out reviews which are not informing or useful for users. Comparison of the approach implemented in Zhang et al. [2010] to our work is presented in Section 7.5. Other works and different approaches to this problem are proposed in Feng et al. [2009]; Tian et al. [2009]; Zhang et al. [2011b]. We consider the score of each product as a combination of the brand score (from Section 6.5), plus the score gathered from the reviews (from Section 6.6). The weight of these two variables can differ from zero to one. This number may change, based on: (i) The number of products with the same brand (in our data-set), and (ii) The number of reviews we have for this product, and the summation of word count of them. These factors assure us that even for products with no review (especially when they are new) we have a partial rate to make the product comparable to other reviews. Equation 6.8 shows these two factors' effects on final product rank. We consider the

average number of reviews for a product in the same group (not category). Same group means products which can be considered comparable as discussed in section 6.4 . When the number of reviews of the product is more or equal to the average number it completely voids the brand rank of the product.

$$\alpha = \begin{cases} \frac{N_r}{avg(N_c)} & \text{if } N_r < avg(N_c) \\ 1 & \text{if } N_r > avg(N_c) \end{cases} \tag{6.7}$$

$$Pr = (1 - \alpha)B_r + \alpha R_r \tag{6.8}$$

Where Nr is number of reviews for the product, Nc is number of reviews for products in the category, Br is brand rank of the product, and Rr is the reviews' rank of the product. The rank from the reviews follows Equation 6.9. The first portion of the equation normalizes the result for products and categories with difference in number of reviews or aspects. The remaining part of the equation sums the aspects of each review to finalize the review rank.

$$R_r = \frac{1}{m \times n} \sum_{k=1}^{n} (\Sigma_{i=1}^{m} (Ar_{i,k} \times Rr_k)) \tag{6.9}$$

The result of this equation, Product Rank (PR), provides a product score. This score then will be sorted compare to other PRs. This product list is the final result of our approach which can be shown to the end user as response to their search query.

6.8 Experiments and Results

The data-set used for our experimental contains two main categories (TVs and Cameras). We limit our data-set to these two categories considering the number of products, their reviews, and secondly, the processing limitations. One should note that as the process of analyzing reviews, sentiment analysis, aspect extraction and

review ranking can be done off-line, the work load directly related to the end user is only limited to creation of the list of ranked products. We use a total of 197 products and 56368 reviews. Our original plan for the experiments was based on 200 products which later reduced to 197 after removing the identical products. For these 197 products, we removed more than 110 reviews as they were marked identical due to having more than 80% similarity to other reviews. A detailed specification of the data-set is shown in Table 6.11. The interface of the application provides the users with search options.

# of products	197	# of products in camera	99
# of reviews	56368	Min # of reviews per product	0
# of products in TV	98	Max # of review per product	1174

Table 6.11: Data-set overview

The search input consists of product categories, product aspects and product price range. Product category is generally the main specifications of the products which are desirable for users, such as “45in TV”. Price range provides user with the option of selecting minimum and maximum price which is acceptable for the final products and the last component of the input screen. As many users have different priorities for their desired aspects, product aspects search option provides users with the option to select the aspect priorities which better suits their needs. If the user does not select the important aspects, we use our default aspect weights to generate the search result. The output of the system provides 5 recommended products alongside the result of similar search on Amazon, omitting the products that are not in our data-set. For our experiments, we have recorded the users’ choices.

The challenges for extracting reviews from Amazon arises from the limitation to the API. By the end of 2011 Amazon stopped providing API users with product reviews. Therefore to gather the reviews we had to parse the iFrame provided from

Amazon and extract the reviews, their writers, number of helpful votes and their star values. As the size of the products and accordingly their reviews increases, we store the information in a database. Figure 6.3 shows the ER diagram of the database.

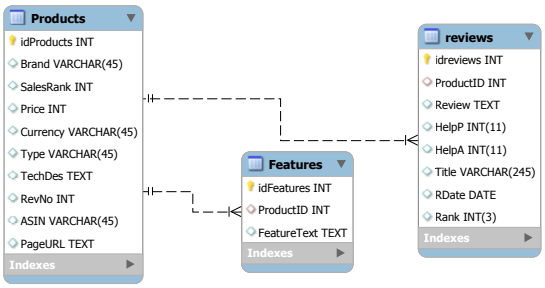

Figure 6.3: Products database ER diagram

6.8.1 Sentiment Analysis Experiment Result

The first experiment demonstrates the performance of the SA classifier. For both parts of our approach we used SVM (Support Vector Machines) with linear kernel. While content-oriented product ranking is not the focus of this paper (and hence not complete) we ran the experiment on a small set of these products as well. As the results show in Table 6.12, the neutrality classifier does not have the same performance on content-driven products. Apart from having a smaller training set, the increase in the number of neutral sentences (as the reviewers are more descriptive about the products), alongside using the same set of features, makes removing the neutral sentences harder. Neutral sentences effects the performance of both neutrality and polarity classifiers. As mentioned earlier our selected features are not as proficient for content-driven products as they are for the other products.

	Recall	Precision	F-measure
Neutrality	45.1	53.3	48.8
Polarity	53.2	55.8	54.4

Table 6.12: Content oriented products Sentiment Analysis

The result in the other two categories (TVs and Cameras) are presented in Table 6.13, are not that different from each other. The little difference between these two categories results from the different approach of reviewers toward the products. In general, for our future work we intend to expand the categories, adding more specific features and creating an automated approach for selecting features based on the specifics of the categories.

	TV			Camera		
	Recall	Precision	F-measure	Recall	Precision	F-measure
Neutrality	59.2	73.5	65.5	57.3	69.7	62.8
Polarity	72.5	78.4	75.3	72.3	81.4	76.5

Table 6.13: Sentiment Analysis experiment results

We also provide a comparison with the Wilson et al. [2009] system in the following, since it is closely related to the proposed approach. Here, we present a comparison of the result of implementing CAPRA and the mentioned work on the same data-set. As the results (Tables 6.14) show our work and feature set has a better performance comparably. For implementation of the approach presented in Wilson et al. [2009], we followed the implementation suggestion with the best performance for neutrality and polarity classifiers in Wilson et al. [2009] using respectively the TiMBL (Daelemans et al. [2003]) tool and BoosTexter (Schapire and Singer [2000]).

	Neutrality			Polarity		
	Recall	Precision	F-measure	Recall	Precision	F-measure
Wilson et al. [2009]	48.5	58.6	54.6	64.7	67.2	65.9
CAPRA	61.6	66.9	64.1	71.6	79.5	75.3

Table 6.14: Neutrality and polarity classifier performance comparison

Finally we show a comparison of our classifier to another classification approach (Sauper et al. [2011]). The mentioned approach works with a probabilistic topic model (mostly Dirichlet distribution) on snippets of yelp reviews, but we expand

this approach to whole body of texts (to make it possible to compare the performance of the two approaches). Table 6.15 shows the result of this comparison. As the result shows CAPRA outperforms this approach when applied to our data-set. We believe there are two main reasons for this result. First, the mentioned approach does not consider neutral sentences in its process and second, considering the initial implementation of this system was focused on text snippets, when applied to whole reviews the system does not perform as expected.

	Recall	Precision	F-measure
CAPRA	71.6	79.5	75.3
Sauper et al. Sauper et al. [2011]	67.6	64.2	65.8

Table 6.15: Polarity classifier result comparison

6.8.2 Product Aspect Analyze Experimental Result

We ran our AA approach (defined in Section 6.4) for the mentioned category of products. In Table 6.16 we present our experiment result for both TV and camera category in detail. The results show the performance of aspect analyzer in extracting the aspects by providing the number of aspects for each category. The second presented information is in regards to how many aspects have been decided as synonyms and removed from the main list of aspects. Finally, we present how many of the aspects were selected using expert opinions, which is the final list presented to the end users.

	TV	Camera
Result of pattern matching	69	46
# of Aspects after similarity check	44	27
# of aspects after expert opinion	28	18

Table 6.16: Result of aspect analysis in TV and camera categories

Another part of aspect analyzer is assigning weights to different aspects. For the purpose of our experiment we assigned α as 0.6 to give more weight to the positive

reviews of the aspects. Table 6.17 presents a sample of aspects weights in the TV category.

Aspects	Weights	Aspects	Weights
Picture quality	.14	Remote backlight	.004
Sound quality	.08	Look	.03
Weight	.01	Application usability	.04

Table 6.17: Sample of aspect weights in the TV category

6.8.3 Review Helpfulness and Product Ranking Experiment Result

In this section we present the result of our experiments for the final product ranking on a small number of products and reviews using both our approach and Zhang et al. [2010]. Furthermore, after providing results of product ranking using steps from previous sections, we present the result of the same experiment while employing a different approach to review usefulness analysis and compare the results with CAPRA. To analyze the performance of the approach we use standard recall, precision and F-measure on the following events. For true positive class (TP) we consider when a user buys a product and is satisfied with it, false positive (FP) is when a user buys a suggested product but is dissatisfied with it, true negative (TN) is when we cannot find an appropriate product and the user agrees based on the normal returned results of the search and finally false negative is when the user is not satisfied with our recommendation, but find the desirable result from Amazon normal search. We expand the experiments with attention to 1, 3 and 5 products. Table 6.18 shows the result of this experiment.

While this experiment was a simulation of the real word scenario, we had a limited number of products. This limitation effected the evaluation specifically in higher number of suggestions as the number of products in each category is very

# of products	TP	FP	TN	FN	Precision	Recall	F-measure
1	58	27	8	7	68.24%	89.23%	77.33%
3	71	16	8	5	81.61%	93.42%	87.12%
5	86	4	8	2	95.56%	97.73%	96.63%

Table 6.18: Experiment result; 1, 3, 5 product recommendations

limited compared to the real world scenario. This effect shows itself strongly in high recall as we have been very selective on the products to be added to the database.

To compare the functionality of our work to real world data, we consider the product sales rank as the gold standard for each category. Table 6.19 shows the result of correlation comparison between CAPRA and the gold standard separately for TV

	Pearson Correlation	Spearman Correlation
TV	56.5	69.2
Camera	56.8	71.1

Table 6.19: Correlation result for CAPRA compared to gold standard

and camera categories using Pearson and Spearman correlation metrics. While the results are satisfactory, we have to remember the goal of CAPRA is to provide better product suggestion and this product is not necessarily the best selling product in the category. Also our approach is specifically tailored for dividing each product category to sub-categories and not to rank all products in each category together.

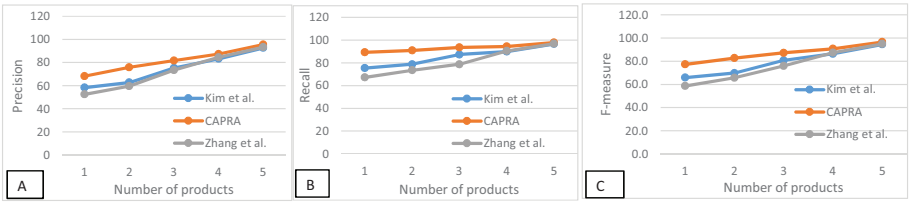

Figure 6.4: CAPRA performance, compared to similar works

The closest work similar to CAPRA presented here for review usefulness analysis is Kim et al. [2006]. While similar, there are key differences in the selected feature-set. The main difference is, with the focus of helpfulness in the mentioned

work, the gold standard is defined as Amazon helpfulness votes of users. For our work, we consider helpfulness votes as one of the features for SVR. The reason (as mentioned in section 6.6) for this decision is that we consider the usefulness of each review for ranking products and not for informative purposes for the end-users. To present a comparison between the two sets of features we run the usefulness analysis using the features from Kim et al. and complete the product ranking using its result.

We can see that for the standard definitions of "recall", "precision", and "F-measure", CAPRA shows better performance (presented in Figure 6.4 A, B and C), in retrieving relevant products to user queries specially in smaller number of returned products. In comparison, the experiment shows that Zhang et al. [2010], as an example of more simplistic approach to product ranking, is not performing as well as our work specially in case of recommending less number of products, but as the number of suggested products increases, the performance difference decreases. Another effect of more suggested products, recognizing the limited number of products in the data-set, the performance of three approaches exponentially becomes closer together. In real world scenarios, with the increase in the number of products in each category, the performance would differentiate more and CAPRA would show considerably better results. Considering all the mentioned points we can safely conclude that our results are satisfactory, and with small modifications and extensions (as part of the future work) can be used in real world scenarios.

CHAPTER 4: PRODUCT WEIGHTED TAXONOMY CREATION

7.1 Introduction

The world of consumer products has seen tremendous changes in the recent years. One can find the root of these changes in two main innovations. First, the advancements in technology have created a world of excitement for users. Nowadays, consumers can look into new products every few months and always find a different approach to the products they have been using in their everyday life. The second innovation is World Wide Web in general and social media specifically which provides users with an environment in which they can freely provide their feedback and reviews shortly after a new release, or analyze and read other people experiences and opinions.

Social Web has introduced a new horizon for gadget lovers. The vast amount of information on different social medias such as Twitter and Facebook, has make it possible to access a large corpora of knowledge just via a few click.

On the other hand, big corporations have new opportunities to advertise their products using word of mouth. While this approach is beneficial for companies, it is arguably even more beneficial for end users, which can filter and analyze the products, tapping into "wisdom of crowds" Surowiecki [2005]. For this purpose we have to consider a pre-requisite which can effect the outcome of the filtering process. Parsing and scraping through the vast amount of knowledge in the aforementioned sources is a time consuming process which can be affected with various factors such as the interests of the user, the filtering process of the social media, etc.

Twitter, as one of the major social medias, follows a special format, which by its nature, forces users to provide a totality of their opinions in the shortest length possible. This restriction, has create a culture which promotes direct, to the point sentences, in many cases following by a URL of a related link.

Furthermore, the fortuity of a product depends on many other factors, other than its aspects, which previously where not as important, or were non-existent. First, most of the current successful gadgets come from a successful line of products, which the popularity of the older versions predetermines the popularity of the new product to some extent. Second, the popularity and opinion of the users of different products for the maker of the gadget, effects how people react to a new product from the same company. Other than these factors, there are many other circumstances which effect the popularity of a product.

In this paper we propose an approach to extract related terms and concepts to a specific product. Furthermore, we have to differentiate the concepts in two ways. First, how important the effect of the newly found term is to our product, and second, is the extracted concept effects the product positively or negatively. The final result of our approach is a weighted taxonomy consist of the product itself, related terms to it, and a weight, presenting how positive or negative the effect of terms on the product is. An example of the extracted graph for Dell XPS laptop is presented in Figure 7.1.

The structure of this paper is as follows. In Section 7.2 we discuss some of the related works to this paper. In section 7.3 we propose our filtering process. Section 7.4 presents the steps we take to create the taxonomy and assign weights to the relationships. We present our experimental results and setup in Section 7.5.

7.2 Related Works

Almost always the first step in information retrieval works on Twitter start with filtering and cleaning tweets. Depending on the task at hand, this process can differ. For example, in case of sentiment analysis, it is common practice to remove tweets which are not include at least one word with non-neutral sentiment Nakov et al. [2013, 2016].

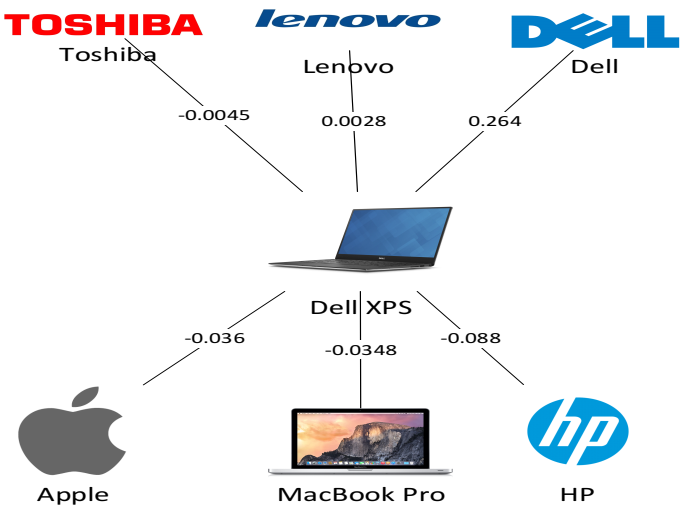

Figure 7.1: Dell XPS Taxonomy

Taxonomy extraction, compared to other sections of this work, is a more recent research topic. The cause of this phenomena can be traced back to two reasons. First, the complexity of extracting terms and relations from free text which needed extensive advances in NLP techniques and strong computational power which was not possible until recent years. And second, based on the requirements of the system, the taxonomy and its structure will differ (for example hypernym-hyponym relations compared to other ontological and semantic relations) and it is not possible to create a generic taxonomy and use it for all purposes.

A recent addition to SemEval competition since 2015 has been taxonomy extraction Bordea et al. [2015]. They generally divide the taxonomy extraction task to three parts: term extraction, relation discovery, and taxonomy construction. To understand the related work to taxonomy extraction better, we also need to divide it to these sub-parts and discuss them separately.

In the context of term extraction, many of the related works goal is to create a set of words which best represents the documents and differentiates between

them. An early example of this approach can be found in Salton et al. [1975]. The ranking of the words in this approach is based on its degree of separation from other unique words in the data-set. This work is one of the statistical approaches to term extraction. Other examples of statistical approaches can be found in some of the more recent works such as Carpena et al. [2009]; Herrera and Pury [2008]. The other approaches to term extraction generally differentiate based on the machine learning approach they use: Unsupervised, semi-supervised and supervised. Formulating the term extraction task as a supervised problem was first done in Turney [2000]. The idea behind this approach is that all the terms in a textual body are potential candidates but the ones should be selected which a human user would select them. Turney Turney [2000] uses a genetic algorithm for the mentioned approach. Other supervised approaches includes (but not limited to) Frank et al. [1999]; Hulth [2003]; Song et al. [2003]. On the other hand, the unsupervised approaches are more diversified. To name some of the approaches, there are simplified works which just select noun phrases in the text Barker and Cornacchia [2000], or perform clustering on the extracted noun phrases Bracewell et al. [2005]. Graph extraction Litvak et al. [2011], non-extensive statistical mechanics Mehri and Darooneh [2011] and TF-RR (Term Frequency-Realized Relation) Gazendam et al. [2010] are some of the other methods used in this context. Finally, the semi-supervised approaches generally use the other pieces of information presented in the document to extract the key phrases. For example in Li et al. [2010] the authors use the notion that the title of an article is the most informative piece of text, and key phrases can be selected based on their semantic similarity to the title.

Relation discovery is a harder task compared to term extraction. An old and still feasible approach to this problem is using lexico-syntactic patterns. Works following this approach consider some pre-created patterns and use those patterns to

relate the text corpora to pre-defined relations Hearst [1992]. A more recent addition to this approach is expanding it by adding learning mechanisms to improvise and extract new patterns as the system is used over time Etzioni et al. [2004]. Considering co-occurrence of terms and concepts in different scenarios is another approach to relation discovery Sanderson and Croft [1999]. Sub-string inclusion and it's use for key-phrase extraction is another approach toward relation discovery. The last addition to various approaches used for this purpose is using semantic relations to discover the relationship between different concepts Navigli and Ponzetto [2010]. There are also works which use Semantic Web resources for the same purpose Lee et al. [2011].

Taxonomy construction in many works equals to taxonomy extraction. There are works which focus on different portions of the task of taxonomy extraction, but many of the works in this field, regardless to their specific task, include the construction of the taxonomy. This task in general combines the result of the previous steps and create an acyclic graph consisting of concepts as vertices and relations as edges Kozareva and Hovy [2010]; Liu et al. [2012].

7.3 Filtering process

There are a few do's and don't's we have to consider for the process of filtering the tweets for our purpose. These factors include the language of the tweets, the frequency of Named Entities (NE from here on) in the tweets and processing the duplicated ones.

For Name Entity Recognition (NER) we tried multiple tools and approaches. For most cases the tools have problems identifying current instances of named entities. At the final phase of our experiment the candidates were NLTK [13], CoreNLP [14] and SpaCy [15]. Interestingly, while CoreNLP and NLTK are the common standards of

[13] http://www.nltk.org/
[14] http://stanfordnlp.github.io/CoreNLP/
[15] https://spacy.io/

academia, SpaCy provides best results by far for the NER task. The complexity of NER on Twitter results from the short length and the different grammatical structures of the tweets.

In regard to the first mentioned point, we have to divide our attention to the two phases of our approach. While we go into more details of each phase in Section 7.4, in short, our work consists of first extracting related concepts, and second extracting their relations. For the purpose of the first portion, the language of the tweets is not of importance considering most of the concepts are known NEs which are not different based on the language of the tweets. On the other hand we have two options for the second portion of our work. We can try to translate the relations, or we can remove the tweets which are not in English. Based on our analysis we decide to remove the tweets for two reasons. First, there are less than 5% of tweets which are not in English in our data-set (we go into more details of the data-set in Section 7.5) and second, considering the high number of tweets in the data-set, the filtered tweets based on their language are minuscule.

We decided to remove tweets which have just one NP. The reasoning for this decision is simple and straight forward. We commerce the search on twitter based on a product which itself is a NP. So if the tweet does not include more than one NP in it, it means that there is not another concept which we can use on our taxonomy creation. It is noteworthy that in many other lines of research, such as sentiment analysis, this approach is not logical and beneficial to the research.

On the third point, our approach is clear and has been used in many other information retrieval approaches using twitter. In our work, we remove the tweets with more than 70% similarity and keep the longer ones and the number which presents the frequency of these tweets. Also we consider the re-tweets as duplicates and treat them similarly.

7.4 Taxonomy Extraction

In recent years the process of releasing a product has gained broader meaning than just releasing its aspects. In this section our goal is two-folded. First, we need to identify and categorize the related term to the selected product, and then to analyze and assign weights (positive or negative) to the extracted concepts.

The resulted taxonomy of our work is an acyclic weighted graph G(V,E), where the vertices are presented as V and $|V| = |relatedterms + 1|$. Considering the nature of this graph, it is also possible to present it as a tree of height two. The root of the tree is the main product, and the leaves are the related concepts, terms or products. In most cases if we expand the graph, find the relations and terms related to the leaves, the graph will change to a cyclic graph in which each of the terms are connected to the root (and some of the other terms) as the condition of having a relation, not necessarily having the same relation, is symmetric between the concepts.

One of the important related concept to a product is its predecessors (if any). If a device has previous versions with the same concept but with significant improvement or changes, then we can use the general sentiment of users toward the old product and a base line for the new product. This is more prevalent for products which are (1) very popular, and (2) update on a yearly schedule. These two concepts have direct relation with each other and having positive sentiment toward an older product has shown to effect the new product positively. In a similar context, the competitors of a product are also important when analyzing a new product announcement. A positive sentiment toward a competitor can negatively effect the popularity of the new product. These are normally the products which consumers have to decide between them when making their final shopping decision. It is noteworthy that based on our analysis in

may cases Twitter users do not include the competing product name but the brand of the product for comparison.

Our previous research Erfan Najmi et al. [2015] has shown that brand name is an important factor on deciding on popularity of products, specially at the time of their release. A positive image from a company can greatly benefit or harm a new product. Other than the brand name itself, a new high level manager in the company or other effective people in the company can also effect the popularity of a product.

Recently, to announce new top of the line products companies like Apple and Microsoft make the announcement in special events which sets the expectations for the mentioned product. The event itself, the key note speaker and other circumstances related to it are effective factors which can influence the popularity of a product. The last points to consider are (1) the products released from the same company which are similar in concept to the new product, (2) The software running on the device.

For example, in case of Iphone 6s we can identify the important concepts as (1) Apple is the brand name of the product, (2) the direct predecessor of Iphone 6s is Iphone 6, (3) the main competitor if Iphone 6s is Galaxy S7, (4) James Cook, CEO of Apple, was the key note speaker of the event, (5) a product with a similar concept to Iphone is Ipad, (6) and finally, the OS of the device is IOS. Note that this list is not exhaustive and many other terms and concepts can be added to it. This list is based on common sense and our understanding of the product, but our experiments have shown a large similarity between our findings based on actual tweets and common sense.

We divide the taxonomy extraction process to three main section. After filtering and removing unwanted tweets, the next task is to extract related terms and concept to our search term. Second part of the work is to find the relation of the terms with each other. What we want to extract from the relation is not precisely what is

the relation, but more importantly, if the general effect of the relation is positive or negative. The last part includes assigning weight to the relation extracted from the previous step. This weight shows the effect of the extracted term to the main product. The final taxonomy is a weighted graph consisting of the main concept that we have searched for, the related concepts extracted previously and their relation which are weighted between -1 and 1. In the following we will describe these parts in detail.

7.4.1 Term Extraction

Choosing a suitable corpora for the key-phrase extraction is the keystone for the success of any approach. The important factors of the corpora can be listed as follows:

- Length: Generally the concern in term extraction is as the length increases, the number of phrase candidates increases which makes it harder to filter and select the more suitable one. For example a scientific paper normally has at least 10 key-phrases Hasan and Ng [2010]. In case of Twitter, we are analyzing one of the shortest text corpora possible, which changes the problem from removing and filtering key phrases, to identifying them in a larger number of tweets.

- Structural Consistency: In normal text documents, articles or scientific papers, there are specific portions which have more importance and value in term extraction. For example in a scientific paper, the abstract and introduction have better key-phrase candidates than the other sections Kim et al. [2013]. In Twitter, each message consists of one or two sentences. There are no difference where a phrase appears considering there are not that many different locations in a tweet.

- Topic Change: In long documents, the key phrases and topic of discussion changes periodically. Considering the length of tweets, it is obvious that this does not happen on regular basis on Twitter.

- Topic Correlation: This observation also is important in longer texts which is important to understand the topics, and hence find the correlation of the topics. Like previous point, this is not applicable to Twitter.

The basis of our taxonomy extraction commence with searching for specific NPs on Twitter. These NPs, for our work, are products which have been recently announced and released. The length of tweets permits us to approach the problem of related term with simple base of Term Frequency (TF) and part of speech tagging.

The length limit of tweets enforces a more straight to the point format which while can be tricky to analyze for semantics, most of the times keeps the NPs which we use in our taxonomy. To extract NPs, after running POS tagging from TweetNLP [16], we filter the tweets with one NP and extract the NPs from the remaining tweets. We then remove the main NP which we commit the search for and count the frequency of the other NPs.

The NPs with highest frequency can be considered as candidates for the taxonomy. The final point we have to consider is to find and remove equal NPs. By equal NPs we mean the terms and concepts that are semantically the same but have various representations. For this purpose we consider character n-grams (as used in Mcnamee and Mayfield [2004]; Stamatatos [2009]) and find words similarities and create a list of equal terms for the concepts we find in this section.

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

7.4.2 Relation Discovery and Weighting

At first glance the problem of assigning weights to the relations of different concepts looks straightforward. We can analyze the sentiment of the relations, by extracting the words specified to the relation (verbs in most cases) we can analyze the pre-specified sentiment value of them and assign the weights based on these values. In reality the weight assignment process is more complicated. We provide further details on these points in this section. Considering this complexity, we divide the work in this section to two parts. First, we discuss how to assign a sign to the relation of the two concepts, and later on we propose a simple approach to weight assignment.

For each product, the related concepts extracted in the previous section can have various relations with each other. While one can name multiple relations between the concepts and the product, the general effect of the concept is positive or negative from the totality of relations. In many tweets, a simple grammatical structure of the sentence can mean different sentiments based on different terms and concepts. In short, the positive sentiment of the relation does not prove or disprove that the relation is positive or negative, So the sentiment value of the relations is disjoint from the sentiment of the relation on the product. Another approach to better describe this problem is as follows: if the relation between a product and a concept is direct, meaning positive change of one causes improvement to the other and vice versa, then the sign of the relation is positive and if the effect of change in one concept has opposite result on the other we assign negative sign to their relation.

If we consider the terms related to our concept as ST when $RC \in ST$, to find the sign of the relationship of the main concept, C, to the new term, RC, we create a new set of related concepts, SRC, following the same steps from the previous section for RC. The new set normally includes the main product, as well as other

concepts more related to the new concept. These two sets have some similarities, but when the two concepts belong to two different family of concepts, the similarity will be minimal.So to find the relation, we compare the similarity of ST and SRC. Our analysis shows that the size of the set resulting from the intersection of these two sets, $ST \cap SRC$, can be used to make an informed decision and conclude the similarity or dissimilarity of the concepts, and in turn extract the sign of the relationships.

For example, in case of Iphone 6s as C and Galaxy S7 as RC, ST includes terms such as Galaxy S7, IOS, Apple, Ipad, etc. and SRC includes Samsung, Android, Note 7, Iphone 6s, etc. The intersection of these two sets would be an empty set which suggests that these two concepts do not have a direct relation and as such have negative relation with each other. On the other hand, for the concept IOS as RC, the set includes Apple, Iphone 6S, Ipad, etc. The intersection of SRC and ST includes most terms of the two sets, which suggests a positive correlation between the two concepts, and as result positive sign of the relation.

The second approach we follow is similar to works in KnowItAll Etzioni et al. [2004]. We create a seed bank of patterns and their signs. While we map these patterns to the related tweets, we find the relationships, and use the terms in those tweets to find other tweets with both NPs present. We then use the new tweets to learn new patterns and extend the data-set. In this approach we start by creating the seed bank for specific category of products. It is essential to understand that the seed bank does not need to be exhaustive, but our heuristic approach is able to extend the limited terms to cover more aspects of related terms to the products when applied on large data-set. Hence, with this approach, it is important to gather the tweets related to each concept and product for a longer period of time, and for higher number of products in each category. But when the sufficient number of tweets, products and

patterns gathered, the same pattern repository can be applied to any product in the same category. Pseudo-code of this approach is presented in the following.

```
Find Sign(noun phrase NP, Tweet set TS
          , rule set RS)
    for each t in TS
        if pattern(t) in RS
            sign-np = sign-np
                      + sign(pattern(t))
        else sign-np is known
            add pattern(t) to RS with sign-np
        else
            remove t from TS
            if t's flag is true
                discard t
            else
                make t's flag true
                add t to end of TS
```

There are some major differences between our approach and KnowItAll. First, KnowItAll makes use of search engines and PMI-IR Turney [2001] metric. In our work we focus on Twitter and use Tweets for all our calculations. The second difference is the use of RDBMS for storing the gathered information. In our work for the same purpose and considering the size of data, our implementation and our project pipeline, we use Pickle [17] to store and retrieve the information.

As mentioned previously, we store some defaults relations in our data-set. An example of such patterns is "C is superlative adverb NP RC" and "C is comparative

[17] https://docs.python.org/2/library/pickle.html

adverb NP RC" in which C is the main concept, NP is any optional noun phrase and RC is the related concept in question. As shown in these two patterns, some patterns are very similar but cover completely different concepts. An instance of the first pattern is "Iphone 6S is the best product of Apple" and an example of the second pattern is "Iphone 6S is better than Galaxy S7", in which the first pattern has positive relation and the second one entails a negative relation.

The main consideration for calculating the weight value is that for each product the event of announcement and releasing a product happens once in its life time. This point is important because, while products can be similar to each other based on their category or their company, the terms which are related to them is always different. Over the years the competitors of products change, the people related to product change their position, and events for each product happens once. This means that we cannot create the taxonomy once and use it for all the similar products. Based on this logic, the process of extracting weight of relations should be done independently for each product and all the concepts related to it.

While in our work we need to weight the relation of the terms and our concept, based on our previous discussion, regarding the insignificance of every single relation between the concepts, and the importance of the relation as a whole, we assign the weights of the relations based on the concepts. After considering various metrics and approaches on term weighting on Twitter (such as Lee et al. [2011]) we chose to use the classic TF.IDF (Term Frequency, Inverse Document Frequency) metric with small modification of replacing documents with hourly Tweets (as presented in Equations 7.1,7.2,7.3. The simplicity of TF.IDF, plus its statistical significance Hiemstra [2000], and the fact that it is directly reflective of user interests on concepts are some of the

benefits of this approach compared to similar metrics.

$$\frac{tf_(i) = n_{i,j}}{N} \tag{7.1}$$

$$N = \sum_{k} n_k, j \tag{7.2}$$

$$idf_i = log\frac{D}{d_i} \tag{7.3}$$

Where $n_{i,j}$ is the number of times word i occurs in document j (including the duplicates) and N is the number of words in document j (excluding the duplicates). In idf_i, d_i is the number of documents which includes the term i and D is the total number of documents in the data-set (both including the duplicates). As mentioned previously, we define documents as all the tweets related to one concept in an hour window.

7.5 Experimental Results

The approach we presented in this work is focused on new products, specially when they are trending on Twitter. We limit this time window from the time of announcement of a new product to the time of its release. Considering this limitation, at the time of preparing this approach we consider two products in two different categories and target Tweets related to them.

All parts of the approach are implemented in Python 2.7. For compatibility issues with different tools used in different sections of the work, we did not use newer versions of Python (e.g. Python 3.5). After inputting the product to the system, we use the Twitter streaming API to get as many Tweets as possible periodically. The result is 281,776 Tweets for (or related to) the Galaxy S7 smart-phone and 124,112 Tweets for (or related to) the Dell XPS laptop. Related topics for the Galaxy S7 include Apple, iPhone 6S, Galaxy S6, Samsung, Android and Nexus. Related topics for the Dell XPS include Apple, Dell, Toshiba, Macbook Pro, Lenovo, and HP. Out of

Table 7.20: Data-set overview

	All tweets	**Non-English**	**Duplicates**	**Final data-set**
Dell XPS 13	124112	11050	52712	60350
Samsung Galaxy S7	287262	21751	71643	193868
Total	411374	32801	124355	248732

Table 7.21: NER approaches comparison

	Precision	**Recall**	**F1**
CoreNLP	0.46	0.44	0.449
NLTK	0.51	0.51	0.51
SpaCy	0.74	0.69	0.714

411,374 total Tweets (for both topics), 32801 were in different languages other than English (however, some foreign tweets made it through). Also there were 48843 direct duplicates and 75512 re-tweets. As discussed in Section 7.3 we do not completely discard these duplicates, but keep their frequency and add it to the weighting equation in the previous section. Details of the data-set for the two products is presented in Table 8.24.

As mentioned in Section 7.3 for the NER task we considered three approaches: **SpaCy**, **CoreNLP** and **NLTK**. Table 7.21 shows the result of this task. While we do not focus on time complexity of these different approaches, it is noteworthy that **SpaCy** not only has the best performance, but also provided the minimum running time between these three systems.

Finally, in the two taxonomies created, there are 12 terms in total, 6 for Galaxy S7 and 6 for Dell XPS. The average weight of relations for Galaxy S7 is 0.099, with the most positive weight being 0.48 for the term “Samsung” and the most negative weight being -0.29 for the term “Apple”; the detailed weight for concepts related to Galaxy

S7 is presented in Table 7.22. For "Dell XPS" the average weight equals -0.035, with the most positive weight being 0.264 for "Dell" and the most negative weight being -0.348 for "Macbook Pro"; thorough description of terms and their weights for "Dell XPS" is presented in Table 7.23.

Table 7.22: Samsung Galaxy S7 related topics

Concept	**# of Tweets**	**Weight**
iPhone 6S	76603	-0.29
Galaxy S6	3470	0.43
Samsung	83888	0.48
Apple	17409	-0.038
Nexus	7012	-0.030
Android	5486	0.042
Total	193868	0.594

Table 7.23: Dell XPS related topics

Concept	**# of Tweets**	**Weight**
Apple	7497	-0.036
Dell	27664	0.264
Toshiba	616	-0.0045
Lenovo	921	-0.0028
HP	5472	-0.088
Macbook Pro	18180	-0.348
Total	60350	-0.2153

CHAPTER 7: KNOWLEDGE HIERARCHY

8.1 Introduction

Richard Saul Wurman expresses in his book titled Information Anxiety, that "Data is fairly worthless to most of us; it is the product of research or creation (such as writing), but it is not an adequate product for communicating. To have informational value, it must be organized, transformed, and presented in a way that gives it meaning." Wurman [1989]. In the past few years the "big data" paradigm has gained considerable attention and the general direction of every information scientist has been on applying different approaches to make a better sense and understanding of the vast amount of information on the Web Manyika et al. [2011].

The World Wide Web (WWW) in its early form provided the backbone to present data which subsequently helped converting this data to information. The information or knowledge hierarchy; the pyramid of Data, Information, Knowledge and Wisdom (Figure 8.1) (DIKW) has been used over the years to pursue this advancement of information systems Rowley [2007]. We can differentiate data and information in the perspective that information is data in a form which is useful Hartley [2008]. However, the one way flow of data, i.e. from content creators to users, in the earlier versions of the Web, made it difficult to fine-tune the acquired information to advance it further on the DIKW hierarchy.

The introduction of Web 2.0 showed the natural need to extend the functionalities of WWW further. The stream of information has transitioned to being bidirectional (users and data providers), and made it easier for users to participate and expand the information and convert it to knowledge. While this knowledge is more and more accessible for humans by using technologies such as Wikis and blogosphere, there is a key step missing from this equation. The lack of translated

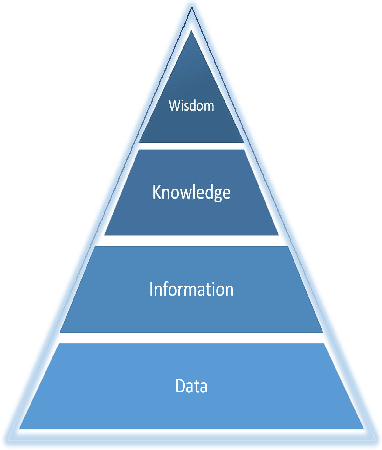

Figure 8.1: The knowledge hierarchy

knowledge in a way which is understandable by machines has been a big shortcoming of Web 2.0 which makes it difficult for search engines to index this knowledge and improve its accessibility.

To close the above mentioned gap, the concept of Semantic Web Berners-Lee et al. [2001] has been introduced and pursued by numerous researchers. The languages and protocols of Semantic Web are mainly focused on providing an intermediary step to make the same knowledge available and indexable for machines. By the use of various technologies, such as different fields of AI and Semantic Web, machines now have an unlimited knowledge of different subjects. For a user with a specific concept in mind, using various approaches to extract that information and convert it to presentable knowledge, in Semantic Web formats such as RDF/XML and TTL to more generic formats such as CSV and TSV, is now possible.

The top layer in the DIKW pyramid and hence the next logical step for the advancement of the Web is to convert the acquired knowledge to wisdom. R.L Ackoff in his address to "International Society for General Systems Research" Ackoff [1989] defines this keystone as "Wisdom adds value, which requires the mental function we call judgement". In other words, the knowledge is available in different formats, but

analyzing, understanding and categorizing it requires extra attention to convert it to wisdom. We believe that this absence of "value" is the reason that the works in AI, in regards to subjects such as Turing test Turing [1948], are not getting to fruition. We describe this point further with an example in the following. Consider a user query to a knowledge base to provide information about apple, the fruit. In case of an apple, the first information a person would care about, other than it being a fruit, is that it comes from a tree, it is edible and normally it is in red, green or yellow. The other pieces of information such as the apple tree is a deciduous tree in the rose family or that it is the most common grown species in genus Malus come later with less importance for a generic user. Since the knowledge base has no prior knowledge of the importance and value of different facts, i.e. from its *core* level to *exterior* details, currently, there is no simplistic way to differentiate the importance of this information on different levels. Hence, the main difficulty in converting the vast knowledge available on the Sematic Web to wisdom is best described as: "Computer based knowledge systems require higher-order mental faculties, but lower to apply knowledge to generate it. In general, they do not develop knowledge, but apply the knowledge developed by people." Ackoff [1989]. In this paper our goal is to identify the depth of this issue and provide some possible solutions and approaches for transforming the available knowledge of the Web in general and the Semantic Web, in particular, to wisdom.

As pointed out by Allen Turing: "If we are trying to produce an intelligent machine, and are following the human model as closely as we can, we should begin with a machine with very little capacity to carry out elaborate operations or to react in a disciplined manner to orders (taking the form of interference). Then by applying appropriate interface, mimicking education, we should hope to modify the machine until it could be relied on to produce definite reactions to certain commands." Turing

[1968]. There are two key points in this passage we want to focus on. First, the simulation of human model, and second, mimicking education. In this context, we combine the human education with cognitive development and follow the natural flow of knowledge learning which is the answer to *how* humans acquire knowledge. In general, we believe that the knowledge should be presented as similar as possible to the process of educating an infant, from the core parts of the knowledge, to more in-depth, less common facts of information.

Finally, we provide fuzzy values in the hierarchy of knowledge. Importance of each fact is not an absolute value which needs to be zero or one. These values can be continuous in a specific range, where the values assigned to each fact should be calculated with two points in mind. First, machines do not have a sense of morality. The knowledge presented in a knowledge base can be considered inappropriate or completely cultural dependant by many users. Filtering the knowledge on different levels to provide a chance for users to order facts can move the less trustworthy facts to a point which a non-specific query regarding a concept does not present those statements to users. Second, while many knowledge bases provide confidence level on the truth value of the facts, the advancement on automatic knowledge extraction have made this process complicated and to some extent overzealous. An automated approach extracts every bit of knowledge without considering the depth and meaning behind human interactions and writings. Our approach is a process in which the confidence level integrates in the presentation process, which is beneficial both in acquisition and delivery of knowledge.

To sum up the above mentioned points, and to categorize our contribution, in this paper we present an addition of a value to each fact in the existing knowledge bases using three different approaches. These approaches assign a fuzzy score to each fact automatically or semi-automatically which generates a wisdom hierarchy. This

hierarchy and the accompanying values is then can be integrated to the knowledge bases using different methods such as RDF reification.

The remainder of this article is organized as follows. In Section 7.2 we review some of the works related to knowledge bases and future of the Web. Section 8.2 provides an overview of our three presented approaches, each creating hierarchy separately, and discusses their details. We present some of the possible use cases of our approach and further discussion on motivation behind this work in Section 8.3. We discuss some of the main dilemmas in creating and implementing our approaches in Section 8.4. In Section 8.5 we present results of our experiments on all the approaches and a comparison of their results, and finally in

8.2 Approaches

The main obstacle for machines to follow the natural knowledge acquisition of humans is that the flow of data in case of an infant is usually a stable learning process which consists of acquiring knowledge gradually over time. In case of a machine, all the information is available at a given time for each concept, and there is minimal to no flow in the knowledge base. The goal in this work is to simulate the natural flow of information presentation over the available knowledge bases.

Each of the approaches discussed in the following taps into the wisdom which is already available for public use on the Web. In the book "Wisdom of crowds" Surowiecki [2005] the authors point out that "under the right circumstances, groups are remarkably intelligent, and are often smarter than the smartest people in them.". We follow the same logic in all the approaches and try to create an average opinion for each fact and its importance/ value.

The foundation of all the approaches discussed in this section is based on two main points. First, for each concept and set of facts related to it, the goal is to create a numerical hierarchy of facts. Figure 8.2 shows a graphical presentation of

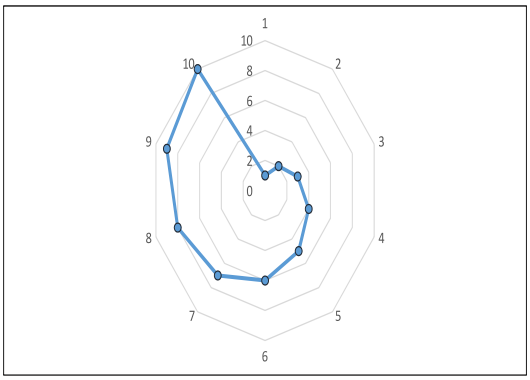

Figure 8.2: Graphical representation of a concept wisdom hierarchy

this approach. Each data point in the figure is a fact related to the same concept. Considering the center as the closest fact to the concept itself, the proximity of each fact to the center of the circle shows it value (importance or interest) in the hierarchy. Second, we consider the natural flow of gaining knowledge for a person as the gold standard. The numerical values are a representation of when in the process of learning the facts related to a concept, those facts should be presented to a person.

In the following, we present the problem with describing the prerequisite and the goal of the process, then provide further details of these approaches.

- Prerequisite: For any concept C we have a list of facts $(f_{1C}, f_{2C}, ..., f_{nC})$. For each fact we select the main list of keywords $(w_{1f_{1c}}, ..., w_{nf_{1c}})$. In most cases such as knowledge bases like ConceptNet, the normalization of words has been processed during the creation of the data-set. We assume the issue of named entity recognition solved (further information can be found in Nadeau and Sekine [2007]), and consider that for each concept we can differentiate it from other entities with the same name (apple the fruit vs Apple the company).

- Goal: For the concept C the goal is to create an ordered list of facts. For each fact there is a numeric value which shows how favorable that piece of information is for the user ($v_{f_{1C}}, v_{f_{2C}}$). The same number is used for ordering the list.

Considering that the knowledge bases we consider for our work mostly follow the RDF model, we consider the facts as triples. For each fact we use RDF reification to add the new value to them. For instance, for the concept of apple, as mentioned in Section 8.1, we can list the following facts: Apple (1)is a fruit, it (2)is edible, normally (3)sweet. it is from (4)Rosaceae family and (5)Plantae kingdom. For the sake of this example, we assign the numbers using common sense (we provide some automated approaches in the following sub-sections, that all map to the common sense in various ways). Also we introduce a new predicate "hasWHvalue" which maps a triple to a numerical value (wisdom hierarchy value). The following shows the complete RDF/XML description of the mentioned facts.

```
<rdf:Description
rdf:about="http://www.example.com/Apple">
  <ex:isA RDF:ID:'s1'>Fruit</ex:isA>
  <ex:hasProperty RDF:ID:'s2'>Edible
  </ex:hasProperty>
  <ex:hasProperty RDF:ID:'s3'>Sweet
  </ex:hasProperty>
  <ex:belongsTo RDF:ID:'s4'>Rosaceae Family
  </ex:hasProperty>
  <ex:belongsTo RDF:ID:'s5'>Plantae Kingdom
  </ex:hasProperty>
</rdf:Description>
<!-- Proposed Addition -->
<rdf:Description
```

```
rdf:resource="#s1">
  <ex:hasKHvalue>0.67</ex:hasKHvalue>
</rdf:Description>
<rdf:Description
rdf:resource="#s2">
  <ex:hasKHvalue>0.48</ex:hasKHvalue>
</rdf:Description>
<rdf:Description
rdf:resource="#s3">
  <ex:hasKHvalue>0.01</ex:hasKHvalue>
</rdf:Description>
<rdf:Description
rdf:resource="#s4">
  <ex:hasKHvalue>0.008</ex:hasKHvalue>
</rdf:Description>
```

In the following sub-sections we provide details of the approaches for creating a wisdom hierarchy. The first section discusses PMI and our modifications to it to create SPMI, the second section discusses the details of two games we improvise to create the hierarchy, and the last sub-section provides details of an approach based on the order of appearance of words in natural language textual bodies.

8.2.1 Singular Pointwise Mutual Information

Pointwise Mutual Information Church and Hanks [1990] or PMI is a generic metric used to find association between concepts. Similar uses of PMI to our approach can be found in different information retrieval works Popescu and Etzioni [2007]; Turney and Littman [2003]. The root of number of *hits*, i.e. number of returned results from search engines, was first introduced in Turney [2001] for finding synonyms in English language in TOEFL like questions. This work presents four variants of

PMI-IR (Pointwise Mutual Information-Information Retrieval)with various degree of complexity. The first equation (equation 8.1) simply considers when the two keywords co-occur (using the "AND" search function) divided by the number of hits for the new query in question. The second variation gives more importance to the keywords being close to each other in the textual body of Web pages (equation 8.2) using the "NEAR" search function. In the third case, the equation gives different scores to concepts related in meaning and to antonyms using the "Not" keyword (equation 8.3). The last and the most complex score (equation 8.4) adds the context of the keywords into consideration. This score adds the context keyword to all the queries and effectively differentiates between different senses of the keywords. In all the equations C stands for the concept in question, f_i for the fact we want to calculate the PMI-IR for, and k for the keyword of the context in discussion.

$$PMI - IR_1 = \frac{hits(C_{AND}\, f_i)}{hits(f_i)} \tag{8.1}$$

$$PMI - IR_2 = \frac{hits(C_{NEAR}\, f_i)}{hits(f_i)} \tag{8.2}$$

$$PMI - IR_3 = \frac{((C_{NEAR}\, f_i)_{AND\, NOT}((CORf_i)_{NEAR}\text{“}not''))}{hits(f_{iAND\, NOT}(f_{iNEAR}\text{“}not''))} \tag{8.3}$$

$$PMI - IR_4 = \frac{hits((C_{NEAR}f_i)_{NEAR}\, k_{AND\, NOT}\, ((C_{OR}f_i)_{NEAR}\text{“}not''))}{hits(f_{iAND}\, k\, _{AND\, NOT}\, (f_{iNEAR}\text{“}not''))} \tag{8.4}$$

In the following, we review the other approaches that utilize PMI-IR in various fields. KnowItAll Yates [2004] is an approach to extract common sense knowledge from Web. In this work different queries run in a search engine to find related facts to different concepts. Product review analysis and feature extraction has been discussed in Turney [2002] by calculating PMI-IR for products and words with strong sentiment values. Similarly, the semantic orientation of words has been discussed in Turney

and Littman [2003]. In this work, other than considering the negative and positive sentiment of words, the degree of strength of the word is also considered. Similarly in Popescu and Etzioni [2007], authors discuss the use of PMI-IR in extracting product aspects from the Web by searching for the products and different possible aspects. Finally social network extraction has been discussed in Matsuo et al. [2007] to find relations between persons, detect groups of persons, and obtain keywords for a person.

In our work, we measure the PMI of the concept C with every fact (f_i) related to the concept. Equation 8.5 is the modified version of PMI-IR we use for this purpose. To calculate each score we use the number of hits from submitting a query to Google (as suggested in Thelwall [2008] for hit count estimates compared to other search engines). Each query consists of the main concept and the keyword of the fact (for example "apple edible"). It is plausible to expand this approach by considering synonyms of the keywords. It is specially beneficial for the facts in which there are multiple words to describe the same fact. For example in case of "apple is edible", for **edible**, one can use synonyms such as tasty or eatable which are easier to use for common users. On the other hand, search engines, such as Google, have a built in function to replace query keywords with highly similar synonyms. An example of this functionality is the query **Cow** which automatically searches for keyword **Cattle** to increase number of desirable results. Our used PMI metric, SPMI-IR (shorten to SPMI from here on) has a major difference with the original metric. In the original, the number of hits of the combination query (combination of original concept and the additional fact) is divided by number of hits for the fact. In our version, the first part of the formula is the same while for the second part the division is done on the number of hits for the original concept, effectively making the metric pivot around the concept instead of the fact. Our decision for this change is two-fold. First, using the hits for the fact instead shows the symmetry of the fact to the concept as

much as the relation of the concept to the fact. Second, the division to the original concept normalizes all the numbers and make them comparable to each other. It is noteworthy that we are not using any of the special search functionalities, such as the ones used in the original PMI-IR metrics.

$$SPMI = \frac{hits(C + f_i)}{hits(C)} \tag{8.5}$$

While this approach has its benefits (such as simplicity), it has some drawbacks. First, the nature of Web is focused on what pieces of information are currently trending. This trend, at times, does not follow logical steps and can increase or decrease the number of hits for different pieces of information unrelated to the overtime interest of users on topics. The second issue is the general trustworthiness of search engines hit numbers. Multiple researches have shown that the number can be different on different days, pages or even for different users Kilgarriff [2007]. The third issue arises from the type of negative relations (e.g. isNotA). While we do not include the relation into the submitted query, having a negative relation changes the fact and exponentially affects the number of hits for it. In these cases, a possible improvement is to modify the query to consider the opposite of the current fact. This method in itself is prone to various issues, e.g. considering the wrong sense of the word in question can return the wrong antonym. In regard to the two first issues, a possible solution is to run periodic updates on the facts. To remove some of the load of these extra computation, we can compare the results of the new SPMI with the previous ones and if the degree of change is less than a threshold we remove the fact from the future updates. The issue which arises from this addition is that each local instance of the wisdom hierarchy would differ from the others, but largely consistent if the update procedure is mostly unified.

Finally, another point to note regarding the results is the overlap of the search results for different queries. In these cases, if the original query is replaced by explicitly excluding the overlapping part of the previous query, the number of hits can change exponentially (e.g. "apple cooking" vs "apple -fruit cooking" which resulted from the facts "apple isA fruit" and "apple usedIn cooking"). This result shows that many of the facts are related to each other and having knowledge of the previous fact is required to understand the other related facts, e.g. unless the reader understands that apple is a fruit, comprehending what kind of use (for example as utensil, spice, or ingredient) apples have in cooking would be hard.

8.2.2 Online games

The use of games to retrieve information, which would be hard to extract otherwise, has been considered and implemented in many different computer science fields over the past decade. Earliest examples of such systems go back to 2004 in Luis von Ahn and Laura Dabbish paper Von Ahn and Dabbish [2004]. In this work the authors introduce a game in which users choose labels for different images and based on the similarity of labels of different users decide on the final labels for each image. *Fun* is one of the keys in defense of the approach: "a game that is fun and can be used to create valuable output". Over the years, there have been many attempts both from researchers and industry to use this new medium to gather information with more ease and accuracy compare to other approaches for information retrieval.

An important work in academia to formalize the use of games for extracting otherwise hard to access information is "Games with a Purpose" Von Ahn [2006] which points out the benefit of using the knowledge and time of users which are already spending time playing games to benefit information extraction. To name some of the game-related works in academia we can point to Von Ahn et al. [2006a], to improve image processing via asking users to identify objects in pictures, Von Ahn

et al. [2006b] and Lieberman et al. [2007], for extracting commons-sense knowledge, and Law et al. [2007] to annotate music and sound. Furthermore, the need of more structured knowledge, and the advent of the Semantic Web has encouraged different research groups to use this new venue for ontology and knowledge base extractions. For example, in "Games with a Purpose for the Semantic Web" Siorpaes and Hepp [2008] the authors discuss three games for creating contents on the Semantic Web. These games include *OntoPronto*, used to extract different domain ontologies from *Wikipedia*; *SpotTheLink* to map *eCl@ss* and the *Unspsc*; *OntoTube* for annotating *YouTube*; and *OntoBay* for annotating *eBay* offerings.

We propose two simple games for the purpose of creating a knowledge hierarchy. In the first game, **Babies' Wisdom**, we present a scenario like the Tamagotchi in which the users are presented with a new born baby which needs to acquire knowledge to be able to enter the world. The user selects the facts the infant needs to acquire to have enough information to function in the real world. To make it more interesting to play, for each concept we use BabelNet Navigli and Ponzetto [2010] to extract an image, if available, of the concept and present it to the user. We also introduce different stages of growth of the infant and let the user advance in the game by assigning a specific number of facts for each stage. To win the game, all the stages of development should pass.

A normal scenario in the game begins with selecting a random concept, presenting an image and all the facts related to it. The user selects one of the facts that she sees fit for this stage (e.g. for the first level "apple is a fruit"), then for the same stage another image, concept name and its related facts are shown and the user chooses the most suitable fact. This step repeats till the threshold number of facts for each stage are gathered and the infant gets older. This process continues for a number of stages of growth and at each stage a number of facts for different concepts

is selected. Another threshold on each fact selection is used to give a score to the user. Currently, in our experiment as the number of facts and concepts is limited, we present all the facts for each concept for the first level and remove the facts accordingly when the user has chosen them in previous levels. We consider the game for kids from age 1 to 10 considering that we have 10 facts for each concept. For the concepts, where we do not have 10 facts available, after the user selects all the facts we redact the concept from the data-set. On the other hand, the threshold for moving to the next level in the game, while selecting 10 facts at the beginning, reduces based on the number of the remaining facts and concepts in the data-set which have not been selected in the previous stages. This number can change over time based-on the number of the facts in the data-set. Later on, with expansion of the data-set the number can be assigned based on user interests or system requirement.

As the model in the game Sims has shown Herz [2005], users can be a powerful force for creating content when they are entertained. This logic can be used to extend the approach to let users insert other facts into the knowledge base and let the average repetition of a fact assign a confidence value to it. This idea is similar to some other works have been done in information retrieval such as OMCS Singh et al. [2002] but with the benefit of being more entertaining than plain fact extraction.

The second game we propose, **Share-a-Fact**, is a multi-player game between two or more players. The game between each pair of people is based-on ordering facts related to one concept. The player is presented with a list of facts and a concept. Each user puts their fact priority for that concept, and after submitting the result the system provides a score based on the result of one other player and a score based on answers of everyone else for that concept. Both scores are based on the similarity of the ordering of the facts. To add more interest for players to play this game we can present it as a match making game based on the similarity of opinions (or namely

interests) on different topics between users. First the user visits the Web page of the game, requests to "host a game" or "join a game". If she selects to host a game a unique key will be generated and presented to the user. The key can be given to as many people as someone likes to attend the game and be compared with their friends. Then the facts related to a random common sense subject is shown to the user and the user can continue onto as many concept as they like. Later on, using the same key which the user was provided first, they can see the result of everyone who has participated in the test with the same key. In the case that a user gets a key from a friend the process is similar except on the first screen she selects "join a game". In the next step the game asks for the key, user inputs the key and is redirected to play the game. In this game the goal is for the user to consider, based on common sense, how would I order the facts. In other words, we do not want personal preferences change the way order or rank the facts.

For each fact we calculate the average order for all the user choices as its score. The orders are gathered from both games. The final result of the approach is the facts ordered by their score. For example if a fact has been selected as the first order three times and as third order five times for a concept C, the score of the fact will be $\frac{(1 \times 3 + 3 \times 5)}{8}$ which is 2.25. If this score is the lowest out of all the facts in the data-set, then this fact orders first out of all the facts related to concept C.

8.2.3 Order of appearance

In information retrieval, most approaches consider natural text as a bag of words (hence the bag of words approach) which are connected in the form of a sentence, the order of appearance of words in sentences, and sentences in paragraphs usually do not show the importance of information. A dominant approach in analyzing texts in information retrieval is Latent Semantic Analysis (LSA) Landauer et al. [1998]. LSA converts paragraphs of text to a matrix which considers the frequency of

words and concepts and finds the similarity of text to different concepts. Obviously converting the textual information to frequency numbers removes the meaning that is provided by the order of words. Other works that consider the order of words use it to comprehend a better semantic meaning of sentences. The importance of order of words has been discussed in detail in Landauer et al. [1997], where it is shown that the performance of LSA for understanding and analyzing sentences is mostly comparable to human readers.

Note that while the order of words in sentences has gathered some attention, we have not found any work considering the order of appearance of sentences in the text. The main reason for this lack of attention is that this order does not effect the truth values of sentences or the facts extracted from them. While the information regarding the order of sentences is mostly useless for extracting facts, we believe it can show the authors' interest in a concept, and what they seem important for placement of a fact in the wisdom hierarchy.

As mentioned previously, knowledge bases do not keep any order in the knowledge they are storing in different formats. This lack of order is one of the main reasons why we need a new approach. While this point is true for knowledge bases, in many resources on the Web, specifically in places where the information is presented in natural language, the order of appearance of information results from the understanding of the author about the interestingness/importance of the knowledge s/he is writing about. A good example, and the resource which we focus on, is Wikipedia articles. Wikipedia articles cover a wide variety of subjects from more generic, common knowledge, to very specific knowledge suitable for experts in each field.

To provide a more in depth example, following is the beginning of Wikipedia entry for **Piano** [18]:

[18] http://en.wikipedia.org/wiki/Piano

> The piano (an abbreviation of pianoforte) is a musical instrument**(1)** played using a keyboard**(2)**. It is widely employed in classical**(3)** and jazz**(4)** music for solo and ensemble performances, accompaniment, and for composing and rehearsal. Although the piano is not portable and often expensive, its versatility and ubiquity have made it one of the world's most familiar musical instruments.
>
> An acoustic piano usually has a protective wooden**(5)** case surrounding the soundboard and metal strings, and a row of 88 black and white keys**(6)** (52 white, 36 black). The strings are sounded when the keys are pressed, and silenced when the keys are released. The note can be sustained, even when the keys are released, by the use of pedals.

As this example shows, the text starts with very general information of the piano by mentioning what it stands for and that it is a musical instrument. While the next few paragraphs also discuss more general concepts of piano, the article mostly goes into details of how a piano works and the history of its invention, that in many cases a user would not be as interested as the more generic information. There are a number of facts in this page but a normal reader possibly will not be interested in all or most of them.

An important point to consider here is that we are not trying to extract the facts from natural text (as done in existing works such as ConceptNet using Reverb Banko et al. [2007]). Our goal is to find the order of appearance (OOA in short from here on) of facts which are already in the data-set in Wikipedia articles of the same concept. To find this order we introduce an approach (similar to other discussed approaches) on the introduction of each concept page of Wikipedia. By introduction we mean the first section of the article before the sub-sections (and not the section named "Introduction" in some of the Wikipedia articles). Our analysis shows that the most clear and useful pieces of facts are normally presented at the beginning of

an article. In many cases the same key facts are repeated at the end of the article, which in most cases is hard to analyze and separate from the

To measure the similarity between each sentence and the fact, normally in triple form, there are different approaches we can take. An older and simpler approach to measure the similarity focuses on similarity of words between the two sentences. This approach creates a vector of words from each sentence (namely w_1 to w_n), then computes the word-wise similarity of the sentences and uses the results for comparing the similarities of different sentences or compares it to a threshold to pass or fail a test Banerjee and Pedersen [2003]; Metzler et al. [2005]. Newer set of approaches point out that a pure vector of words does not present the semantic meaning of a sentence correctly, and suggests other measures to formulate sentence similarities Malik et al. [2007].

For our purpose, the decision of what metric to use, there are two important factors. First, we are not comparing two complete sentences. On one end we have a factual statement normally presented in a triple or other normalized forms, and on the other end we have a sentence presented in natural language. To solve this structural difference, a logical step is to similarize the structure, by normalizing the natural text to make it more structurally similar to our factual statement. This change of structure infers that we are not focused on semantic similarity, but a more generic similarity as introduced in older works. Second, other than being more viable, the simpler approaches are less computationally expensive and easier to implement and use. For each part of the text we first split the text to sentences, run tokenizer, and create word lemmas on the sentences. Using the words lemmas, we measure their similarity to every fact related to the concept in discussion. Considering that the predicate in the fact is normally modified and normalized, and there are many equal and similar terms which can be used in place of the predicates. Thus, we remove the

predicate and focus on similarity of subjects and objects of triples. For each concept we create a matrix of facts and sentences. The score for each fact is the order of the sentence with the highest similarity (Equation 8.7). We use a metric similar to overlap similarity Metzler et al. [2005] as presented in Equation 8.6. The bold numbers in the piano example show the actual order of facts which this approach extracts and stores in the data-set.

$$sim(f_i, s_j) = \frac{overlap(f_i, S_j)}{length(f_i)} \tag{8.6}$$

$$score(f_i) = order(max(sim(f_i, s)) \tag{8.7}$$

The overlap similarity is basically the number of overlapping words divided by the length of the two sentences. In our case we change the second part of the equation to just the length of the fact. The reasoning behind this change is mainly the longer length of the sentences in Wikipedia compared to the length of the facts in our data-set (as discussed in Section 8.5) . In the original equation, in many cases, the second part of the formula would be just the length of the sentence, considering the length of the fact is trivial compared to the total length of the sentence and the fact together.

8.3 Use-Cases and Further Motivation

In the following, we present some more in depth use-cases which our approach can help solve or improve upon. One of the first uses which can potentially benefit from our work is searching, e.g. for apple the fruit. A normal Google search of this concept returns some basic information regarding this fruit which Google deems important for general users. This information includes the first few lines of a Wikipedia article (which we also use for one of our approaches) and a table of nutritional facts regarding the fruit. While we think that the Wikipedia article is a good source of information, we believe that the nutritional facts of a fruit is not one of the first pieces

of information which a general user would be interested in. One main reason for providing this information from Google is its accessibility compared to other facts. Our approach, on the other hand, creates an ordered list of information which on a basic level, by searching for a concept alone, can present the general knowledge related to a concept. These facts can be related to the origin of a concept, how and where it is useful, or any other fact which has been selected via our work as interesting and useful for the users.

Over the years the concept of search engines has expanded and changed gradually. While a few years ago the only way to get related results would be to omit the extra words and search for main keywords in the query, recently most search engines allow users to input complete queries, and in many cases return direct results to users. We state that the result of our proposed approach in this work can be an effective addition to search engines. To explain this use case further, consider a search query on a scientific term. If the user is not an expert or have a generic inquiry about the fact, the search engine would still return the most popular Web page, which in this case or many similar cases is not what the general user is looking for. We believe this result can be improved by using the users' previous search history and their social media profiles (when available). This information can help to predict user expertise in the subject and provide information which is more useful to that specific user. While this expansion is useful in many cases, many users have concerns regarding their Web surfing history and what companies do with their data which needs special attention before going further with this work.

As discussed previously, the Turing test Turing [1950] can be directly affected by the methodology and approaches introduced in this paper. Currently, there are many chatbots approaching the general Turing test from different perspectives. Most common practice for creating a chatbot is based on pattern matching. Over the years,

the bots have advanced further by making use of many different technologies. One of the recent extensions is using knowledge bases for information extraction. For example Tarau et al. Tarau and Figa [2004] uses FrameNet Baker et al. [1998] lexical knowledge base along with Open Mind common sense knowledge collection Singh et al. [2002] and WordNet Miller [1995] to create a conversational agent which simulates a story telling process. While the chatbots can consider the general flow of conversation between two humans, the issue arises when the discussion flow moves to acquiring or asking for knowledge. A machine, without any order on its knowledge would return one of many possible answers to an inquiry, while our approach can present the user with answers which are influenced and ordered based on normal human interest to the facts related to the question and the conversation. In many cases the creators of the chatbots consider specific personas for them, where the personas can be of specific age, gender or level of knowledge. We believe this is not only compatible with our work but our approach can be beneficial in creating an ever evolving persona which can show different level of knowledge based on the context and the partner of the discussion.

Recently, there has been some discussion on use of the mentioned chatbots for e-learning Kerly et al. [2007]. This expanded focus on E-learning shows its importance both now and in the future. One of the biggest challenges in E-learning is creating content which satisfies the user expectations and preferences Turker et al. [2006]. We believe simulating the natural flow of knowledge in our work can minimize the content creators' efforts to organize the knowledge in a suitable format for different audiences. This, and the possibility of creating different levels for different people with various degree of education and age are useful additions to current efforts to expand E-learning.

8.4 Discussion

We posit that the diversity of opinion between different users may pose a concern. The difference in opinions usually results from age, expertise, culture and interests. For instance the concept of apple for a five years old is mostly limited to the fruit to eat as snack. However, for a middle aged person, although the normal use of apple is obvious, there may be more interest in the fact that one can use apple for cooking. Similarly, a nutritional expert compared to a normal person would be interested in some other facts in regards to any edible fruit.

The second issue is how to encode the information with other pieces of knowledge and converge them to create an easy extraction method. In the context of Semantic Web, if each fact is presented as a triple, relating a newly created triple with the score of the fact to the original fact in the flat structure of most triple based languages is problematic. "Reification" is the method used in RDF for addressing this issue, but even though it makes the implementation of this process a possibility, extracting the results still can cause other issues.

Furthermore, even after adding the value we assign to facts, the flat structure of information, as used in Semantic Web or similar contexts, does not provide the tools to comprehend human behaviors such as sarcasm, etc. For example, "I work 40 hours a week to be this poor" which cause a machine to establish the fact that working 40 hours a week results in being poor. Similarly, it is not possible for a machine to understand jokes or idioms (a good example of a combination of both an idiom and a joke would be "Time flies like an arrow, fruit flies like banana" which has 2 facts in it, first "time files" and second "fruit flies", where the first one is from an idiom and the second one is to expand it to a joke). Moreover, the common practice in Semantic Web, using a semantic reasoner (such as Pellet Sirin et al. [2007] and Hermit Shearer

et al. [2008]) on top of a knowledge base considers these pieces of information as knowledge discrepancies (e.g. in the phrase "bite the bullet" considering the facts (1) human bites edible item, (2) bullet is not edible, this piece of information is discrepancy) and removes them, which in turn reduces the value of the knowledge.

Another problem arises from the diversity of opinions as some personal opinions show themselves as facts in our data-set. Considering that the main source of knowledge in our approach is common sense knowledge bases, some of the facts are more opinionated than factual. For example the triple "Microsoft is evil", which is an opinion, effects the score, specially SPMI, when we consider that the number of hits for this query is not that different from a factual statement like "Microsoft locatedIn Redmond". An appropriate solution for this issue can be the construction of better confidence metrics to separate facts and opinions, or facts and wrong information which is out of the scope of this work.

Finally, there is the issue of scalability. For an approach like OOA, which stores the numerical value of which fact appears where in a body of text, addition of new facts can change the order completely which requires reassigning all the other scores of the other facts. The same issue can be extended to online games. SPMI on the other hand does not have this issue and as many facts as required can be added to the data-set without effecting the other scores. Also we have to note that while higher score of SPMI shows higher wisdom order of the fact it is the opposite for the other two approaches (lower score shows highest wisdom order).

8.5 Experiments

In the following, we review the results of our three approaches in creating the wisdom hierarchy. In each section we discuss and review the important notes in regards to each special and general case(s) in that approach. Notes, sample results, and

comparison pertaining to all the three approaches are discussed in the "comparison" subsection.

Setup: For the experiments we have selected 10 concepts and 10 facts for each concept. These facts and concepts are handpicked from three main knowledge bases: ConceptNet Liu and Singh [2004]; Speer and Havasi [2013], FreeBase Bollacker et al. [2008] and DBPedia Auer et al. [2007]. The reason behind manually selecting both the concepts and the facts is to diversify the concepts and select facts which are on different degree of interest to users. Table 8.24 shows some details of the data-set, where the concepts are listed in the first column, the second column shows the number of hits for the concept and the last column shows number of facts for each concept in the data-set. Each fact is in the form of a triple consisting of "subject predicate object". For example three of the facts for the concept apple are "Apple isA fruit", "Apple hasProperty Sweet" and "Apple hasProperty green".

Table 8.24: Data-set details

Concepts	**# of facts**	**# of hits**
Apple	10	1.51B
Piano	10	635M
Cow	10	203M
Hatred	10	63.8M
Math	7	397M
Ipad	9	917M
Armadillo	10	15.1M
Microsoft	10	1.08B
Brownie	7	43.9M
Barack Obama	10	196M

8.5.1 Approach 1: SPMI

We ran our SPMI metric on a number of concepts in different fields. The results show that SPMI is effective for gathering and ordering facts, i.e. in most cases the results are what is expected for users' interests in different fields. However, the

variance of the SPMI scores for facts in each concept shows a difference in SPMI values. This variation is mainly due to the number of hits (changing from billions to thousands) and second, from the difference in referencing facts ("hatred isRelatedTo cyberhate", vs "hatred isRelatedTo cyberspace", which considering that both facts are in regard to hatred, "cyberhate" would become synonym of "cyberspace", and the facts entail the same meaning).

Table 8.25: Number of hits and SPMI for the concept "cow" in our data-set

Related fact	**# of hits**	**SPMI**
isA animal	71.1M	0.35
produces milk	10.8M	0.053
livesIn barn	13.1M	0.064
eats grass	28.5M	0.14
relatedTo hamburger	800K	0.003
relatedTo religious	17M	0.083
has mapped genome	463K	0.002
produces leather	16.9M	0.083
isAbleTo pull carts	516K	0.002

Second, the variance of all the SPMI values in our experiment is 0.02804034 or a little less than 3%. The highest variance we have is for the concept **brownie** with 0.060856215 or 6%. On the other hand the minimum variance is for **armadillo** with 0.000155401. An interesting point regarding these results is that the maximum and minimum value of variance in all the concepts belongs to two of the concepts with near minimum or minimum number of hits. Considering that the variance of SPMI presents the diversity of opinions of users, this shows that the interest of users to facts is more normalized for more popular concepts. Figure 8.3 shows the correlation of number of hits and variances of concepts for our data-set. There are two extreme situations we need to consider. First when we have a very high number of hits for a concept, while the interest to the facts related to it is not as much. In this case the

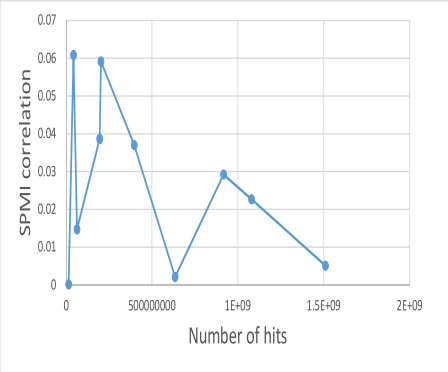

Figure 8.3: SPMI variance and hit correlation

SPMI for most of the facts is very low which in turn effects the variance and other metrics related to the concept. The second extreme case is when the number of hits for a concept is considerably low and the facts have relatively similar number of hits. We can trace this situation to when the hits for different facts of a concept are mainly similar in meaning.

On the other hand, the number of hits (while as mentioned previously are not precise) show the popularity or interest in each concept. In all the concepts we have selected we have at a minimum 15.1 millions hits for **Armadillo** which in accordance has reduced the number of hits for the facts related to it. On the other hand, for some other concepts such as **Microsoft** we have exponentially higher hits (1.08 billions).

In some cases, the number of hits does not exhibit the real importance of the fact. This can result from the difference in terms where users describe the facts. For example, "piano isA musical instrument" returns 11.6 million results compared to **piano** 636 million results or "piano jazz music" 84.1 million results. While for most people the most basic fact regarding "piano" is that it is a music instrument, the terms people use for describing it would be different. Another effective variable on the number of hits is the terms which are related to each concept, in the context of Internet, e.g. "virtual" and "piano". This Internet specific term is effective enough

to change the first result to a Web site which presents virtual piano. Finally, in some cases, specially in case of isA relation, when the fact shows the main category of a concept, mentioning the fact seems too obvious for many users on the Web. In turn, this greatly effects the number of hits for those specific facts.

8.5.2 Approach 2: Online games

We can consider the result of the games as a gold standard for comparison of our results. In some cases, the results are the same as when a user, aware of the purpose of the system, selects and orders the results. The results show that in most cases the players' opinions are similar to each other. However, there are two main exceptions to this. First, in cases where the concept is not very commonly used. For example regarding the concept **Armadillo** (from Table 8.24), after the first three facts (Armadillo is an animal, a mammal and is native to south America), the others are not very well-known to public (such as Armadillo is nocturnal, and one of Xenarthra animals). Moreover, when the fact is less known, the choice is usually made on instant interest. The second exception in user opinions arises from the facts which are related to moral ideas and opinions of the player. An instance of this issue is "Microsoft hasProperty monopolist" which not only is an old discussion in regard to Microsoft, but the players can put it ahead or after many other facts based on their personal opinion.

For some facts and concepts, the similarity of interest in different concepts has its biggest effect in the game scores. These facts can be the most popular or the least popular facts specifically for the concepts which had the least SPMIs from the previous subsection. For the facts which gather the minimum interest, many players choose to select the answers randomly or based on their first impression. In a more general perspective, the lack of interest for any concept affects the order selected by

the users. A possible solution to this issue is to give users the choice of concepts they want to play on.

Another group of concepts which have special circumstances are emotions. Emotions are deeply connected to experiences of the users playing the game. This situation can also be expanded to the concepts which are connected to special experience for different people or cultures. For example the concept of **war** for someone who has experienced it first-hand or has grown up with it is very different from someone who has a remote connection (e.g. from movies or news). In general, the psychological state of mind of the users is an important factor in the final order of facts when the sample size of the users does not contain many different cultural and social backgrounds. A limited group of players, as in our experiment, can include very specific personalities and personal histories which can effect the final result of the system. This is why it is an important requirement of the online games to increase the sample size of the participants.

In out experiments, for Share-a-Fact game, we had 65 participants which were divided to two groups. One group were randomly matched with other participants. These players were not aware of the identity of the other participants. On the other hand, another group of participants participated in the game with their friends. In both cases we created groups of 2, 3 and 4 and let the players know of the size of their team. Our experiment shows that in case of random teams, the result of the experiment is more rationalized, meanins it is more similar to what one normally would choose the orders. For groups of friends, in some cases the result is not following the logical order which can be translated to the players knowledge of each other. Table 8.26 is a summary of the groups and players of this game.

We limited number of rounds in Babies' wisdom to 5 for the purpose of our experiments, meaning each of the players could play one to five rounds of the game.

Table 8.26: Share-a-fact participants statistics

Team size	Random match	Pre-made match
2	6	4
3	4	3
4	3	3

Each round consists of 5 questions and each of the questions is related to one concept. Table 8.27 provides an overview of the players in this game.

Table 8.27: Babies' wisdom participants statistics

# of rounds	# of participants	Percentage
1	10	17
2	16	28
3	10	17
4	8	14
5	13	23

8.5.3 Approach 3: Order of Appearance

This approach is comparably different from the other two mentioned previously. The main point of difference can be found in the nature of documents we use as reference for finding the order of facts. While the previous approaches directly tapped into the wisdom of crowds, in this case the documents are prepared by a limited number of users. Considering that Wikipedia entries can be modified and changed by any user, the users can be experts or complete novices in regards to the subject of the documents. The other problem is when a user has knowledge about one aspect of a concept and not the others. For example, a user writing the document for artificial intelligence can be an expert in neural networks, thereby focusing mostly on concepts related to neural networks and not paying attention to many other topics of interest related to artificial intelligence. On the other hand, facts which are not of interest to an expert may be the facts which are interesting for general users, while

the general user writing a document may not have the depth of knowledge to write more detailed facts for a concept.

A benefit of the current state of this approach compared to the other ones is its execution time complexity. The majority of the process for this approach is to analyze the sentences and find the related facts to a given fact. As discussed in Section 8.2 we have chosen a more simplistic approach as necessary for our work which simplifies this process exponentially. Out of the three approaches, SPMI is the other automated one which while not computationally very expensive, is still more expensive than this approach. To compare, the process for this approach consists of extracting the text of the Wikipedia article, extract the keywords and compare the facts to each other. Considering that the normal list of facts for each concept in our work is 10 on average, the resulting matrix of facts and sentences will not be hard to process. On the other hand for SPMI, the process consists of running N+1 queries on Google (where N is number of facts) which because of the response and loading time of the pages is considerably slower than the order of facts (between 45% and 65% for different concepts).

The main issue with this approach are the facts which are not presented in the introduction of Wikipedia articles. We found two extreme case of this issue in our data-set, one for **Armadillo** and second for **hate**. We can trace this back to two completely different root. First in case of hate, as hate is an emotion, there are many different opinions which are considered facts for an emotion. These cases do not generally appear in Wikipedia articles as the articles focus on more scientific aspects of emotions. On the other hand, in case of Armadillo, the mentioned facts are mostly facts which appear in the information box in the article. In many cases, as in Armadillo, these information are not repeated in the introduction of the article.

A possible solution for cases similar to **Armadillo** concept above is to consider the information boxes on Wikipedia pages. This addition raises two new questions for our approach. First, where in the order of facts the information box will be placed, and second, how do you consider the precedence of information considering that there is a standard format for information boxes on Wikipedia. Considering these issues, and the acceptable results of our approach in its current state, we decide against processing the information boxes in our experiments and approach.

8.5.4 Approach Comparison and Results

To give a better feel of the output of the system, we start with presenting the results for all three approaches for the concept apple. Table 8.28 shows this result. It is noteworthy that for each approach while we get a score, the scores are not comparable on their own, and the order of facts has more information than the score itself. In case of apple, the OOA approach does not provide the score for the first two facts, as they are not presented in the introduction of the Wikipedia entry. The most difference between facts in this example are, first, OOA result for Ganus Malus compared to the other two approaches (because of the more scientific nature of Wikipedia entry) and second, for the fact "apple is eatable" specially using SPMI (normally users do not use the term eatable for apple even when they mention eating it. In comparison if we replace **eatable** with **eat**, the order would change to 3).

The main metric we consider for comparison of approaches is correlation of orders. The correlation shows how similar the results of ordering using the three approaches are to each other. For this purpose we use "Pearson Product-Moment Correlation Coefficient" as presented in Equation 8.8. The reduction of the average order in the equation helps by removing some of the noise resulting from the lack of order in some of the facts in OOA approach. An example of this situation shows itself in the highest correlation in our data-set between games and OOA for the concept

Table 8.28: Order of facts for the concept apple.

Fact	**SPMI**	**OOA**	**Games**
hasProperty Green	1		3
hasProperty Sweet	2		4
isA Fruit	3	2	1
usedFor Cooking	4	6	6
grownOn Apple tree	5	1	5
growsFrom Seed	6	5	7
usedFor Cider	7	8	8
hasProperty Eatable	8	7	2
GrownIn Central Asia	9	4	9
belongsTo Ganus Malus	10	3	10

"hate", when we note that using OOA we just have the orders of 4 out of 9 total facts for this concept.

$$r = \frac{\Sigma(x - \bar{x})(y - \bar{y})}{\sqrt{\Sigma(x - \bar{x})^2 \Sigma(y - \bar{y})^2}} \tag{8.8}$$

Figure 8.4 and Table 8.29 show the relation of number of hits and correlation of different approaches. The data shows that the number of results is related to higher correlation for concepts with high or low hits. We have ordered the number of hits and removed the numbers for the purpose of creating this figure mainly because the difference between the hit numbers is large which after scaling makes the numbers meaningless. On the other hand, the correlations show similar trend towards different concepts in general. With all the similarity in trend, the correlation between games and OOA has the largest difference which is mainly because these two approaches are not related to hit numbers in any way. As mentioned previously the highest correlation we have is for concept **hate** in games and OOA, on the other hand, the lowest correlation is for OOA and SPMI for the concept **Armadillo** with -0.97. Both the high number and low number of correlation for these two concepts is because

OOA does not cover all the facts in the data-set (four out of nine facts were found in Armadillo Wikipedia article).

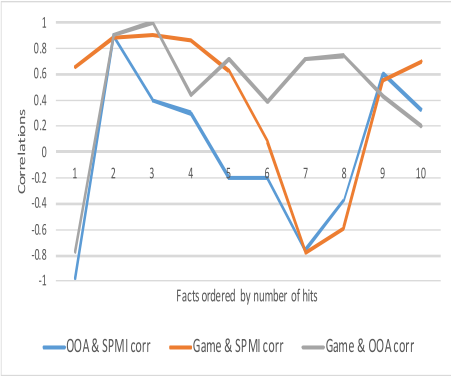

Figure 8.4: Number of hits and correlation relation

Table 8.29: Concepts and correlation of approaches

Concept	**Game & OOA Corr**	**Game & SPMI Corr**	**OOA & SPMI Corr**
Apple	.19	.7	.32
Piano	.71	-.78	-.76
Cow	.71	.63	-.2
Hatred	1	.9	.4
Math	.38	.08	.2
IPad	.73	.59	.38
Armadillo	-.77	.6	-.97
Brownie	.9	.88	.9
Barack Obama	.44	.85	.29

Next, we discuss the run time of OOA and SPMI. Considering that games do not have a specific time window to run and the time for each user to play a round of game is up to the user, for this part of the work we do not consider the games' run times. In the following, first we discuss the time required to retrieve the data for one fact, and later expand it to the run time of full measurement for each concept. For each fact, SPMI requires to perform two search queries and one simple

calculation. On the other hand, for each OOA calculation we have to retrieve one Web page and process on average 8 lines of text in our approach. For each sentence we have to remove the stop words, run a tokenizer and normalize the sentence and then calculate the similarity of the fact to the sentence in question. Considering all these steps, the time complexity of this approach is higher than SPMI for one fact. We also ran a separate set of experiments by removing the intermediary steps of the process (normalizing and tokenizing). This change simplifies and shortens the process considerably (to retrieve the page, and search for the word in the fact in the introduction which only depends on the length of it and has linear time complexity).

Table 8.30 shows the results of these experiments. As shown in this table, for one fact the modified OOA is the fastest approach, and SPMI follows it with a small margin. The original OOA is slower than the other two and based on the ordering results has around 30% increased performance (meaning it found 3 more facts than the modified OOA out of 10 facts). When considering the scores for all the facts in one concept, SPMI run time greatly exceeds the other approaches considering that for all the approaches the most time consuming part is retrieving pages. In both OOA and modified OOA we just retrieve the page once and the other parts of the process runs locally while for SPMI it is required to do one search query for each fact, which increases the run time exponentially. Considering that SPMI always returns results and in most cases these results are satisfactory, there is a trade-off between better speed (OOA and modified OOA) and better performance (SPMI).

Table 8.30: Time consumption comparison of approaches (in seconds)

Approach	**Single fact**	**One concept**
SPMI	2.84	17.2
OOA	4.21	4.9
Modified OOA	1.67	2.8

First, an important factor in all the results is the importance of IsA relation in our data-set. Recognition of each concept in more than 95% of concepts achieved the highest rate. This result is even consistent in some of the cases when there are multiple instances of isA relation for one concept. Still, there is an exception to this case. For concepts in which its category or named entity is well known that it is considered common sense for everyone, and to discuss the concept you do not need to explicitly mention what kind of object it is, we can see that the rank dramatically decreases (specially in SPMI or OOA approach).

CHAPTER 8: CONCLUSIONS

Providing users with the information they require is the task I have focused on during the years. To try and ease the retrieval and improving the accuracy of this retrieval we start with implementing an approach for question answering using Semantic Web. In this work we first translate the question to a triple form, and then use SPARQL to search for the answer in an RDF knowledge base.

The mentioned approach depends heavily on a Semantic knowledge base. To expand the possibility of use of this approach we suggest expanding it by converting other knowledge bases to Semantic Web format. To do so, we focus on ConceptNet knowledge base and first introduce an upper ontology with capability to be used for the conversion process. We then propose a process for the conversion process.

The next step on our work is in regard to ranking products based on user reviews. I believe this work is to look at the question answering process from another perspective. This work includes analyzing customer reviews, extract sentiment values based on different aspects of the products and assign a value to a product based on its brand, which we address respectively and propose solutions for them.

Other than customer reviews over products in e-commerce Web sites, people's opinion in microblogging Web sites such as Twitter is also affects the product sale and popularity. We believe that the opinion of people in regard to a product depends highly on related concepts and products. In this regards we propose an approach to extract these terms and concepts and create a weighted taxonomy which shows how effective each of these extracted terms are to a product.

My research in these works has shown a different issue with the current state of information retrieval. I believe that the information is available on the Web in different formats, but we need to create a system which build a hierarchy on top of the knowledge available and practically convert this knowledge to wisdom. So the final

challenge which I like to address is to create an order for the knowledge presented in different content and knowledge bases. Considering that none of the presented facts have any preference on which piece of information is more interesting or closer to the core of the knowledge compared to other related facts. In my opinion, this is an interesting topic which requires through analysis and experimentation which can result in many interesting uses.

Finally, I believe all the works I have done till now have created a backbone on both the knowledge and functionalities which can positively influence the future works and create new methodologies and approaches for creating an easier and more accurate information gathering process for users.

REFERENCES

"Mindpixel project," 2005, http://www.mindpixel.org.

"Upper mapping and binding exchange layer," 2012, http://umbel.org/.

D. A. Aaker and K. L. Keller, "Consumer evaluations of brand extensions," *The Journal of Marketing*, pp. 27–41, 1990.

R. L. Ackoff, "From data to wisdom: Presidential address to isgsr, june 1988," *Journal of applied systems analysis*, vol. 16, no. 1, pp. 3–9, 1989.

B. Adida and M. Birbeck, "Rdfa primer: Bridging the human and data webs," *Retrieved June*, vol. 20, p. 2008, 2008.

D. Allemang and J. Hendler, *Semantic web for the working ontologist: effective modeling in RDFS and OWL*. Access Online via Elsevier, 2011.

R. Arp and B. Smith, "Function, role, and disposition in basic formal ontology," *Nature Precedings*, vol. 1941, no. 1, pp. 1–4, 2008.

S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, *Dbpedia: A nucleus for a web of open data*. Springer, 2007.

C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The berkeley framenet project," in *Proceedings of the 17th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 1998, pp. 86–90.

S. Banerjee and T. Pedersen, "Extended gloss overlaps as a measure of semantic relatedness," in *IJCAI*, vol. 3, 2003, pp. 805–810.

M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open information extraction for the web," in *IJCAI*, vol. 7, 2007, pp. 2670–2676.

K. Barker and N. Cornacchia, "Using noun phrase heads to extract document keyphrases," in *Advances in Artificial Intelligence*. Springer, 2000, pp. 40–52.

D. W. Barnes and J. M. Mack, "Zermelo-fraenkel set theory," in *An Algebraic Introduction to Mathematical Logic*. Springer, 1975, pp. 52–61.

S. Bechhofer, F. Van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, L. A. Stein *et al.*, "Owl web ontology language reference," *W3C recommendation*, vol. 10, pp. 2006–01, 2004.

T. Berners-Lee, J. Hendler, O. Lassila *et al.*, "The semantic web," *Scientific american*, vol. 284, no. 5, pp. 28–37, 2001.

A. Bernstein, E. Kauffmann, C. Kaiser, and C. Kiefer, "Ginseng: A guided input natural language search engine," *Proc. of the 15th Workshop on Information Technologies and Systems*, 2005.

K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. ACM, 2008, pp. 1247–1250.

G. Bordea, P. Buitelaar, S. Faralli, and R. Navigli, "Semeval-2015 task 17: Taxonomy extraction evaluation (texeval)," *SemEval-2015*, vol. 452, no. 465, p. 902, 2015.

D. B. Bracewell, F. Ren, and S. Kuriowa, "Multilingual single document keyword extraction for information retrieval," in *Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE'05. Proceedings of 2005 IEEE International Conference on*. IEEE, 2005, pp. 517–522.

D. Brickley and R. V. Guha, "{RDF vocabulary description language 1.0: RDF schema}," 2004.

D. Brickley and L. Miller, "Foaf vocabulary specification 0.98," *Namespace Document*, vol. 9, 2012.

S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer networks and ISDN systems*, vol. 30, no. 1, pp. 107–117, 1998.

R. R. Brinkman, M. Courtot, D. Derom, J. Fostel, Y. He, P. W. Lord, J. Malone, H. E. Parkinson, B. Peters, P. Rocca-Serra *et al.*, "Modeling biomedical experimental processes with obi." *J. Biomedical Semantics*, vol. 1, no. S-1, p. S7, 2010.

E. Cambria, R. Speer, C. Havasi, and A. Hussain, "Senticnet: A publicly available semantic resource for opinion mining." in *AAAI Fall Symposium: Commonsense Knowledge*, vol. 10, 2010, p. 02.

A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, and T. M. Mitchell, "Toward an architecture for never-ending language learning." in *AAAI*, vol. 5, 2010, p. 3.

P. Carpena, P. Bernaola-Galván, M. Hackenberg, A. Coronado, and J. Oliver, "Level statistics of words: Finding keywords in literary texts and symbolic sequences," *Physical Review E*, vol. 79, no. 3, p. 035102, 2009.

K. W. Church and P. Hanks, "Word association norms, mutual information, and lexicography," *Computational linguistics*, vol. 16, no. 1, pp. 22–29, 1990.

P. Cimiano, P. Haase, and J. Heizmann, "Porting natural language interfaces between domains – an experimental user study with the orakel system," *5th International Semantic Web Conference*, 2006.

D. Crockford, "The application/json media type for javascript object notation (json)," 2006.

W. Daelemans, J. Zavrel, K. Van der Sloot, and A. Van den Bosch, "Timbl: Tilburg memory-based learner," *version*, vol. 4, pp. 02–01, 2003.

D. Damljanovic, M. Agatonovic, and H. Cunningham, "Natural language interface to ontologies: Combining syntactic analysis and ontology-based lookup through the user interaction," *Proc. of the European Semantic Web Conference, Heraklion*, 2010.

L. Data, "Linked data - connect distributed data across the web," 2012, http://linkeddata.org/.

T. . Dataset, "Trec," 2004, http://trec.nist.gov/data/qa/t2004_qadata.html.

N. Dawar and P. Parker, "Marketing universals: consumers' use of brand name, price, physical appearance, and retailer reputation as signals of product quality," *The Journal of Marketing*, pp. 81–95, 1994.

M. Dean, G. Schreiber, S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein, "Owl web ontology language reference," *W3C Recommendation February*, vol. 10, 2004.

X. Ding, B. Liu, and P. S. Yu, "A holistic lexicon-based approach to opinion mining," in *Proceedings of the international conference on Web search and web data mining*. ACM, 2008, pp. 231–240.

C. O. C. A. ENGLISH, "Coca," 2012, http://corpus.byu.edu/coca/.

E. Erfan Najmi, K. Hashmi, Z. Malik, A. Rezgui, and H. Khan, "Capra: a comprehensive approach to product ranking using customer reviews," *Computing, DOI*, vol. 10, 2015.

A. Esuli and F. Sebastiani, "Sentiwordnet: A publicly available lexical resource for opinion mining," in *Proceedings of LREC*, vol. 6. Citeseer, 2006, pp. 417–422.

O. Etzioni, M. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates, "Web-scale information extraction in knowitall:(preliminary results)," in *Proceedings of the 13th international conference on World Wide Web*. ACM, 2004, pp. 100–110.

A. Fader, S. Soderland, and O. Etzioni, "Identifying relations for open information extraction," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 1535–1545.

C. Fellbaum, *WordNet*. Wiley Online Library, 1999.

Q. Feng, K. Hwang, and Y. Dai, "Rainbow product ranking for upgrading e-commerce," *Internet Computing, IEEE*, vol. 13, no. 5, pp. 72–80, 2009.

L. Floridi, *The Blackwell guide to the philosophy of computing and information*. John Wiley & Sons, 2008.

E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning, "Domain-specific keyphrase extraction," in *IJCAI*, vol. 99, 1999, pp. 668–673.

L. Gazendam, C. Wartena, and R. Brussee, "Thesaurus based term ranking for keyword extraction," in *Database and Expert Systems Applications (DEXA), 2010 Workshop on*. IEEE, 2010, pp. 49–53.

S.-L. Ge and R. Song, "Automated error detection of vocabulary usage in college english writing," in *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, vol. 3. IEEE, 2010, pp. 178–181.

A. Ghose and P. G. Ipeirotis, "Designing ranking systems for consumer reviews: The impact of review subjectivity on product sales and review quality," in *Proceedings of the 16th Annual Workshop on Information Technology and Systems*. Citeseer, 2006, pp. 303–310.

M. Grassi, "Developing heo human emotions ontology," in *Biometric ID Management and Multimodal Communication*. Springer, 2009, pp. 244–251.

M. Grassi and F. Piazza, "Towards an rdf encoding of conceptnet," in *Advances in Neural Networks–ISNN 2011*. Springer, 2011, pp. 558–565.

B. F. Green, A. K. W. Jr., C. Chomsky, and K. Laughery, "Baseball: An automatic question-answerer," *IRE-AIEE-ACM '61 (Western) Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference*, 1961.

R. J. Hartley, *Organizing knowledge: an introduction to managing access to information*. Ashgate Publishing, Ltd., 2008.

K. S. Hasan and V. Ng, "Conundrums in unsupervised keyphrase extraction: making sense of the state-of-the-art," in *Proceedings of the 23rd International Conference on*

Computational Linguistics: Posters. Association for Computational Linguistics, 2010, pp. 365–373.

M. A. Hearst, "Automatic acquisition of hyponyms from large text corpora," in *Proceedings of the 14th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics, 1992, pp. 539–545.

M. Hepp, "Goodrelations: An ontology for describing products and services offers on the web," in *Knowledge Engineering: Practice and Patterns*. Springer, 2008, pp. 329–346.

H. Herre, B. Heller, P. Burek, R. Hoehndorf, F. Loebe, and H. Michalek, "General formal ontology (gfo)–a foundational ontology integrating objects and processes," *Onto-Med Report*, vol. 8, 2006.

J. P. Herrera and P. A. Pury, "Statistical keyword detection in literary corpora," *The European Physical Journal B*, vol. 63, no. 1, pp. 135–146, 2008.

J. Herz, "Harnessing the hive," *Creative industries*, pp. 327–41, 2005.

D. Hiemstra, "A probabilistic justification for using tf× idf term weighting in information retrieval," *International Journal on Digital Libraries*, vol. 3, no. 2, pp. 131–139, 2000.

M. Horridge, H. Knublauch, A. Rector, R. Stevens, and C. Wroe, "A practical guide to building owl ontologies using the protégé-owl plugin and co-ode tools edition 1.0," *University of Manchester*, 2004.

E. Hovy, L. Gerber, U. Hermjakob, C. Lin, and D. Ravichandran, "Toward semantics-based answer pinpointing," *Proceeding HLT '01 Proceedings of the first international conference on Human language technology research*, 2001.

C.-W. Hsu, C.-C. Chang, C.-J. Lin *et al.*, "A practical guide to support vector classification," 2003.

M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 168–177.

N. Hu, J. Zhang, and P. Pavlou, "Overcoming the j-shaped distribution of product reviews," *Communications of the ACM*, vol. 52, no. 10, pp. 144–147, 2009.

A. Hulth, "Improved automatic keyword extraction given more linguistic knowledge," in *Proceedings of the 2003 conference on Empirical methods in natural language processing*. Association for Computational Linguistics, 2003, pp. 216–223.

A. Ittycheriah, S. Roukos, and S. Roukos, "Ibm's statistical question answering system-trec 11." in *TREC*, 2002.

C. M. Joy and S. Leela, "Review on sentence-level clustering with various fuzzy clustering techniques," *International Journal of Engineering*, vol. 2, no. 12, 2013.

E. Kauffmann, A. Bernstein, and R. Zumstein, "Querix: A natural language interface to query ontologies based on clarification dialogs," *Proc. of the 5th International Semantic Web Conference*, 2006.

A. Kennedy and D. Inkpen, "Sentiment classification of movie reviews using contextual valence shifters," *Computational Intelligence*, vol. 22, no. 2, pp. 110–125, 2006.

A. Kerly, P. Hall, and S. Bull, "Bringing chatbots into education: Towards natural language negotiation of open learner models," *Knowledge-Based Systems*, vol. 20, no. 2, pp. 177–185, 2007.

R. Khare and T. Çelik, "Microformats: a pragmatic path to the semantic web," in *Proceedings of the 15th international conference on World Wide Web*. ACM, 2006, pp. 865–866.

A. Kilgarriff, "Googleology is bad science," *Computational linguistics*, vol. 33, no. 1, pp. 147–151, 2007.

S.-M. Kim, P. Pantel, T. Chklovski, and M. Pennacchiotti, "Automatically assessing review helpfulness," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2006, pp. 423–430.

S. N. Kim, O. Medelyan, M.-Y. Kan, and T. Baldwin, "Automatic keyphrase extraction from scientific articles," *Language resources and evaluation*, vol. 47, no. 3, pp. 723–742, 2013.

G. Klyne and J. J. Carroll, "Resource description framework (rdf): Concepts and abstract syntax," 2006.

J. Ko, L. Si, E. Nyberg, and T. Mitamura, "Probabilistic models for answer-ranking in multilingual question-answering," *ACM Transactions on Information Systems*, 2010.

Z. Kozareva and E. Hovy, "A semi-supervised method to learn and construct taxonomies using the web," in *Proceedings of the 2010 conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2010, pp. 1110–1118.

T. K. Landauer, D. Laham, B. Rehder, and M. E. Schreiner, "How well can passage meaning be derived without using word order? a comparison of latent semantic analysis and humans," in *Proceedings of the 19th annual meeting of the Cognitive Science Society*. Citeseer, 1997, pp. 412–417.

T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse processes*, vol. 25, no. 2-3, pp. 259–284, 1998.

O. Lassila and R. R. Swick, "Resource description framework (rdf) model and syntax specification," 1999.

E. L. Law, L. Von Ahn, R. B. Dannenberg, and M. Crawford, "Tagatune: A game for music and sound annotation." in *ISMIR*, vol. 3, 2007, p. 2.

C.-H. Lee, C.-H. Wu, and T.-F. Chien, "Burst: a dynamic term weighting scheme for mining microblogging messages," in *International Symposium on Neural Networks*. Springer, 2011, pp. 548–557.

S. Lemaignan, R. Ros, L. Mosenlechner, R. Alami, and M. Beetz, "Oro, a knowledge management platform for cognitive architectures in robotics," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 3548–3553.

D. B. Lenat, M. Prakash, and M. Shepherd, "Cyc: Using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks," *AI magazine*, vol. 6, no. 4, p. 65, 1985.

D. Li, S. Li, W. Li, W. Wang, and W. Qu, "A semi-supervised key phrase extraction approach: learning from title phrases through a document semantic network," in *Proceedings of the ACL 2010 conference short papers*. Association for Computational Linguistics, 2010, pp. 296–300.

H. Lieberman, D. Smith, and A. Teeters, "Common consensus: a web-based game for collecting commonsense goals," in *ACM Workshop on Common Sense for Intelligent Interfaces*, 2007.

M. Litvak, M. Last, H. Aizenman, I. Gobits, and A. Kandel, "Degexta language-independent graph-based keyphrase extractor," in *Advances in Intelligent Web Mastering–3*. Springer, 2011, pp. 121–130.

H. Liu and P. Singh, "Conceptneta practical commonsense reasoning tool-kit," *BT technology journal*, vol. 22, no. 4, pp. 211–226, 2004.

J. Liu, Y. Cao, C.-Y. Lin, Y. Huang, and M. Zhou, "Low-quality product review detection in opinion summarization," in *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007, pp. 334–342.

X. Liu, Y. Song, S. Liu, and H. Wang, "Automatic taxonomy construction from keywords," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 1433–1441.

V. Lopez, M. Pasin, and E. Motta, "Aqualog: An ontology-portable question answering system for the semantic web," *Lecture Notes in Computer Science*, pp. 135–166, 2005.

V. Lopez, V. Uren, M. R. Sabou, and E. Motta, "Cross ontology query answering on the semantic web: An initial evaluation," *Proceedings of the fifth international conference on Knowledge capture*, pp. 17–24, 2009.

V. Lopez, M. Pasin, and E. Motta, "Aqualog: An ontology-portable question answering system for the semantic web," in *The Semantic Web: Research and Applications*. Springer, 2005, pp. 546–562.

V. Lopez, V. Uren, E. Motta, and M. Pasin, "Aqualog: An ontology-driven question answering system for organizational semantic intranets," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 5, no. 2, pp. 72–105, 2007.

Y. Lu, P. Zhang, J. Liu, J. Li, and S. Deng, "Health-related hot topic detection in online communities using text clustering," *PloS one*, vol. 8, no. 2, p. e56221, 2013.

R. Malik, L. V. Subramaniam, and S. Kaushik, "Automatically selecting answer templates to respond to customer emails." in *IJCAI*, vol. 7, 2007, pp. 1659–1664.

G. S. Mann, "Fine-grained proper noun ontologies for question answering," in *Proceedings of the 2002 workshop on Building and using semantic networks-Volume 11*. Association for Computational Linguistics, 2002, pp. 1–7.

J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers, "Big data: The next frontier for innovation, competition, and productivity," 2011.

Y. Matsuo, J. Mori, M. Hamasaki, T. Nishimura, H. Takeda, K. Hasida, and M. Ishizuka, "Polyphonet: an advanced social network extraction system from

the web," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 5, no. 4, pp. 262–278, 2007.

J. McCarthy, *Programs with common sense*. Defense Technical Information Center, 1963.

D. L. McGuinness, F. Van Harmelen *et al.*, "Owl web ontology language overview," *W3C recommendation*, vol. 10, no. 10, p. 2004, 2004.

P. Mcnamee and J. Mayfield, "Character n-gram tokenization for european language text retrieval," *Information retrieval*, vol. 7, no. 1-2, pp. 73–97, 2004.

A. Mehri and A. H. Darooneh, "Keyword extraction by nonextensivity measure," *Physical Review E*, vol. 83, no. 5, p. 056106, 2011.

D. Metzler, Y. Bernstein, W. B. Croft, A. Moffat, and J. Zobel, "Similarity measures for tracking information flow," in *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM, 2005, pp. 517–524.

A. Miles and S. Bechhofer, "Skos simple knowledge organization system reference," Technical report, W3C, Tech. Rep., 2009.

G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to wordnet: An on-line lexical database*," *International journal of lexicography*, vol. 3, no. 4, pp. 235–244, 1990.

E. T. Mueller, *Natural language processing with ThoughtTreasure*. Citeseer, 1998.

S. Nadali, M. Murad, and R. Kadir, "Sentiment classification of customer reviews based on fuzzy logic," in *Information Technology (ITSim), 2010 International Symposium in*, vol. 2. IEEE, 2010, pp. 1037–1044.

D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.

E. Najmi, K. Hashmi, F. Khazalah, and Z. Malik, "Intelligent semantic question answering system," in *Cybernetics (CYBCONF), 2013 IEEE International Conference on*. IEEE, 2013, pp. 255–260.

E. Najmi, K. Hashmi, Z. Malik, A. Rezgui, and H. U. Khan, "Conceptonto: An upper ontology based on conceptnet," in *Computer Systems and Applications (AICCSA), 2014 IEEE International Conference on*. IEEE, 2014.

P. Nakov, Z. Kozareva, A. Ritter, S. Rosenthal, V. Stoyanov, and T. Wilson, "Semeval-2013 task 2: Sentiment analysis in twitter," 2013.

P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani, and V. Stoyanov, "Semeval-2016 task 4: Sentiment analysis in twitter," in *Proceedings of the 10th international workshop on semantic evaluation (SemEval 2016), San Diego, US (forthcoming)*, 2016.

R. Narayanan, B. Liu, and A. Choudhary, "Sentiment analysis of conditional sentences," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. Association for Computational Linguistics, 2009, pp. 180–189.

R. Navigli and S. P. Ponzetto, "Babelnet: Building a very large multilingual semantic network," in *Proceedings of the 48th annual meeting of the association for computational linguistics*. Association for Computational Linguistics, 2010, pp. 216–225.

V. Nguyen, O. Bodenreider, and A. Sheth, "Don't like rdf reification?: making statements about statements using singleton property," in *Proceedings of the 23rd international conference on World wide web*. International World Wide Web Conferences Steering Committee, 2014, pp. 759–770.

N. F. Noy, M. Sintek, S. Decker, M. Crubézy, R. W. Fergerson, and M. A. Musen, "Creating semantic web contents with protege-2000," *IEEE intelligent systems*, vol. 16, no. 2, pp. 60–71, 2001.

L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." 1999.

B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2004, p. 271.

B. Parsia and E. Sirin, "Pellet: An owl dl reasoner," in *Third International Semantic Web Conference-Poster*, vol. 18, 2004.

E. A. Pessemier, "A new way to determine buying decisions," *The Journal of Marketing*, pp. 41–46, 1959.

L. Polanyi and A. Zaenen, "Contextual valence shifters," in *Computing attitude and affect in text: Theory and applications*. Springer, 2006, pp. 1–10.

A.-M. Popescu and O. Etzioni, "Extracting product features and opinions from reviews," in *Natural language processing and text mining*. Springer, 2007, pp. 9–28.

E. PrudHommeaux, A. Seaborne *et al.*, "Sparql query language for rdf," *W3C recommendation*, vol. 15, 2008.

R. Quirk and D. Crystal, *A comprehensive grammar of the English language*. Cambridge Univ Press, 1985, vol. 6.

F. Radulovic and N. Milikic, "Smiley ontology," in *Proceedings of The 1st International Workshop On Social Networks Interoperability*, 2009.

Y. Raimond, S. A. Abdallah, M. B. Sandler, and F. Giasson, "The music ontology." in *ISMIR*. Citeseer, 2007, pp. 417–422.

D. Ravichandran and E. Hovy, "Learning surface text patterns for a question answering system," *ACL '02 Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pp. 41–47, 2002.

S. K. Ray and B. J. S. Singh, "A semantic approach for question classification using wordnet and wikipedia," *Pattern Recognition Letters*, vol. 31, no. 13, pp. 342–351, 2010.

J. E. Rowley, "The wisdom hierarchy: representations of the dikw hierarchy," *Journal of information science*, 2007.

G. Salton, C.-S. Yang, and C. T. Yu, "A theory of term importance in automatic text analysis," *Journal of the American society for Information Science*, vol. 26, no. 1, pp. 33–44, 1975.

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

C. Sauper, A. Haghighi, and R. Barzilay, "Content models with attitude," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 350–358.

R. E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," *Machine learning*, vol. 39, no. 2-3, pp. 135–168, 2000.

R. Shearer, B. Motik, and I. Horrocks, "Hermit: A highly-efficient owl reasoner." in *OWLED*, vol. 432, 2008, p. 91.

P. Singh, T. Lin, E. T. Mueller, G. Lim, T. Perkins, and W. L. Zhu, "Open mind common sense: Knowledge acquisition from the general public," in *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*. Springer, 2002, pp. 1223–1237.

A. Singhal, S. Abney, M. Bacchiani, M. Collins, D. Hindle, and F. Pereira, "At&t at trec-8," *Proceedings of the 8th Text Retrieval Conference, NIST*, 2000.

K. Siorpaes and M. Hepp, "Games with a purpose for the semantic web," *IEEE Intelligent Systems*, no. 3, pp. 50–60, 2008.

E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical owl-dl reasoner," *Web Semantics: science, services and agents on the World Wide Web*, vol. 5, no. 2, pp. 51–53, 2007.

B. Smith, W. Ceusters, B. Klagges, J. Köhler, A. Kumar, J. Lomax, C. Mungall, F. Neuhaus, A. L. Rector, and C. Rosse, "Relations in biomedical ontologies," *Genome biology*, vol. 6, no. 5, p. R46, 2005.

M. Song, I.-Y. Song, and X. Hu, "Kpspotter: a flexible information gain-based keyphrase extraction system," in *Proceedings of the 5th ACM international workshop on Web information and data management*. ACM, 2003, pp. 50–53.

R. Speer and C. Havasi, "Conceptnet 5: A large semantic network for relational knowledge," in *The Peoples Web Meets NLP*. Springer, 2013, pp. 161–176.

M. Sporny, T. Inkster, H. Story, B. Harbulot, and R. Bachmann-Gmür, "Webid 1.0: Web identification and discovery," *W3C Editors Draft*, 2011.

E. Stamatatos, "Intrinsic plagiarism detection using character n-gram profiles," *threshold*, vol. 2, pp. 1–500, 2009.

N. Stanford, "Group (2005)," *Stanford parser. Retrieved*, vol. 12, no. 1, p. 2005, 2005.

V. Stoyanov and C. Cardie, "Topic identification for fine-grained opinion analysis," in *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2008, pp. 817–824.

F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: A Core of Semantic Knowledge," in *16th international World Wide Web conference (WWW 2007)*. New York, NY, USA: ACM Press, 2007.

J. Surowiecki, *The wisdom of crowds*. Anchor, 2005.

V. Tablan, D. Damljanovic, , and K. Bontcheva, "A natural language query interface to structured information," *Proceeding ESWC'08 Proceedings of the 5th European semantic web conference on The semantic web*, pp. 361–375, 2007.

M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Computational linguistics*, vol. 37, no. 2, pp. 267–307, 2011.

P. Tarau and E. Figa, "Knowledge-based conversational agents and virtual storytelling," in *Proceedings of the 2004 ACM symposium on Applied computing*. ACM, 2004, pp. 39–44.

M. Thelwall, "Quantitative comparisons of search engine results," *Journal of the American Society for Information Science and Technology*, vol. 59, no. 11, pp. 1702–1710, 2008.

P. Tian, Y. Liu, M. Liu, and S. Zhu, "Research of product ranking technology based on opinion mining," in *Intelligent Computation Technology and Automation, 2009. ICICTA'09. Second International Conference on*, vol. 4. IEEE, 2009, pp. 239–243.

M. B. Traylor, "Product involvement and brand commitment." *Journal of Advertising Research*, 1981.

D. Tsarkov and I. Horrocks, "Fact++ description logic reasoner: System description," in *Automated reasoning*. Springer, 2006, pp. 292–297.

A. M. Turing, "Intelligent machinery, a heretical theory," *The Turing Test: Verbal Behavior as the Hallmark of Intelligence*, p. 105, 1948.

——, "Computing machinery and intelligence," *Mind*, pp. 433–460, 1950.

——, "Intelligent machinery. reprinted in cybernetics: Key papers. ed. cr evans and adj robertson," 1968.

A. Turker, İ. Görgün, and O. Conlan, "The challenge of content creation to facilitate personalized e-learning experiences," *International Journal on E-Learning*, vol. 5, no. 1, pp. 11–17, 2006.

P. Turney, "Mining the web for synonyms: Pmi-ir versus lsa on toefl," 2001.

P. D. Turney, "Learning algorithms for keyphrase extraction," *Information Retrieval*, vol. 2, no. 4, pp. 303–336, 2000.

——, "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews," in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 417–424.

P. D. Turney and M. L. Littman, "Measuring praise and criticism: Inference of semantic orientation from association," *ACM Transactions on Information Systems (TOIS)*, vol. 21, no. 4, pp. 315–346, 2003.

C. Unger, L. Bhmann, J. Lehmann, A. N. Ngomo, D. Gerber, and P. Cimiano, "Template-based question answering over rdf data," *Proceedings of the 21st international conference on World Wide Web*, pp. 639–648, 2012.

P. University, "Wordnet," 2012, http://wordnet.princeton.edu/.

L. Von Ahn, "Games with a purpose," *Computer*, vol. 39, no. 6, pp. 92–94, 2006.

L. Von Ahn and L. Dabbish, "Labeling images with a computer game," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2004, pp. 319–326.

L. Von Ahn, M. Kedia, and M. Blum, "Verbosity: a game for collecting common-sense facts," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 2006, pp. 75–78.

L. Von Ahn, R. Liu, and M. Blum, "Peekaboom: a game for locating objects in images," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 2006, pp. 55–64.

P. Vossen, "Introduction to eurowordnet," in *EuroWordNet: A multilingual database with lexical semantic networks*. Springer, 1998, pp. 1–17.

C. Wang, M. Xiong, Q. Zhou, and Y. Yu, "Panto: A portable natural language interface to ontologies," *4th European Semantic Web Conference*, 2007.

X. Wang, A. McCallum, and X. Wei, "Topical n-grams: Phrase and topic discovery, with an application to information retrieval," in *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*. IEEE, 2007, pp. 697–702.

S. Weibel, J. Kunze, C. Lagoze, and M. Wolf, "Dublin core metadata for resource discovery," *Internet Engineering Task Force RFC*, vol. 2413, no. 222, p. 132, 1998.

E. Wiener, J. O. Pedersen, A. S. Weigend *et al.*, "A neural network approach to topic spotting," in *Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval*. Citeseer, 1995, pp. 317–332.

Wikipedia, "The free encyclopedia," 2012, http://www.wikipedia.org/.

T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis," *Computational linguistics*, vol. 35, no. 3, pp. 399–433, 2009.

W. Wu, H. Li, H. Wang, and K. Q. Zhu, "Probase: A probabilistic taxonomy for text understanding," in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. ACM, 2012, pp. 481–492.

R. S. Wurman, *Information anxiety*. Doubleday, 1989.

A. Yates, "Web-scale information extraction in knowitall," in *Proceedings of the 13th International*, 2004.

D. Zhang and W. S. Lee, "Question classification using support vector machines," *SIGIR '03 Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 2003.

K. Zhang, R. Narayanan, and A. Choudhary, "Voice of the customers: mining online customer reviews for product feature-based ranking," in *3rd Workshop on Online Social Networks*, 2010.

K. Zhang, Y. Cheng, W.-k. Liao, and A. Choudhary, "Mining millions of reviews: a technique to rank products based on importance of reviews," in *Proceedings of the 13th International Conference on Electronic Commerce*. ACM, 2011, p. 12.

K. Zhang, Y. Cheng, Y. Xie, D. Honbo, A. Agrawal, D. Palsetia, K. Lee, W.-k. Liao, and A. Choudhary, "Ses: Sentiment elicitation system for social media data," in *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*. IEEE, 2011, pp. 129–136.

ABSTRACT

IMPROVING USER EXPERIENCE IN INFORMATION RETRIEVAL USING SEMANTIC WEB AND OTHER TECHNOLOGIES

by

ERFAN NAJMI

December 2016

Advisor: Dr. Zaki Malik

Major: Computer Science

Degree: Doctor of Philosophy

The need to find, access and extract information has been the motivation for many different fields of research in the past few years. The fields such as Machine Learning, Question Answering Systems, Semantic Web, etc. each tries to cover parts of the mentioned problem. Each of these fields have introduced many different tools and approaches which in many cases are multi-disciplinary, covering more than one of these fields to provide solution for one or more of them. On the other hand, the expansion of the Web with Web 2.0, gave researchers many new tools to extend approaches to help users extract and find information faster and easier. Currently, the size of e-commerce and online shopping, the extended use of search engines for different purposes and the amount of collaboration for creating content on the Web provides us with different possibilities and challenges which we address some of them here.

AUTOBIOGRAPHICAL STATEMENT

Erfan najmi was born in Tehran , IRAN. He received his undergraduate in Information Technology Engineering from Tabriz University of IRAN. He joined the Wayne State University in the Spring Semester of 2010 to pursue graduate studies in Computer Science. After receiving a Master's degree in Computer Science, he later on transferred with his advisor Dr. Zaki Malik to continue his Ph.D. studies. His research interests include Information Retrieval and Machine Learning.