NTNU
Norwegian University of
Science and Technology

Extracting Named Entities and Synonyms from Wikipedia for use in News Search

Christian Bøhn

Master of Science in Computer Science

Submission date:	June 2008
Supervisor:	Kjetil Nørvåg, IDI
Co-supervisor:	Jon Atle Gulla, IDI
	Stein L. Tomassen, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Problem Description

In news search it is desirable to recognize entities in both documents and queries and link synonymous entities together. This may to aid in retrieval of relevant documents when the users are searching after one variant of the entity, for instance "United Nations" instead of "UN".

In this project we will explore using Wikipedia as the mining source for automatically building a dictionary of synonyms referring to the same named entity. Next, we will use this dictionary in a search application, where query expansion is used in an attempt at normalize the named entities. In the evaluation we will evaluate the quality of the extracted entities and their synonyms using precision/recall, then we will evaluate the modified search system against the original version to see if it is actually improving the results.

Assignment given: 15. January 2008
Supervisor: Kjetil Nørvåg, IDI

Abstract

News articles are often closely tied to named entities such as a person, a company or similar. One challenge from an information retrieval point of view is that a single entity can have more than one way of referring to it. This means that when users look for articles about a specific person that is appearing in the news, unless they use the same name for the entity they may not find the articles they are looking for. For example, some articles will refer to the *United Nations* by the full name, others will use the abbreviation, *UN*, and the rest may even use both.

In this thesis we explore the idea of using Wikipedia as a subject to data mining with the goal of building a large dictionary of named entities and their synonyms. The motivation for this is that we can then use it to recognize and link different synonyms together and thereby be able to include documents where the entity being sought is included, but where the naming is different.

As part of this thesis we implement a mining component which is capable of extracting named entities from Wikipedia using two different strategies. Secondly, we implement a modified search system using query expansion to reformulate queries to include synonyms when an entity is detected. At the end we evaluate and discuss the results.

The evaluation shows that Wikipedia is well suited as a source of named entities and synonyms as the semi-structure aids in recognizing entities and related synonyms. The use of the dictionary in a modified search solution is giving mixed results on the other hand. One challenge with evaluating the modified search solution is that when only a single entity is used as a query, then the relevant documents will easily exceed the top 10 results. In the end we consider Wikipedia a good source of entities, but the usage of such an entity dictionary can be improved.

Preface

This report presents the master thesis of my 5th year in the Computer Science Master Program at NTNU, Trondheim. The work was carried out at the Department of Computer and Information Science, Faculty of Information Technology, Mathematics and Electrical Engineering at NTNU, first under the supervision of Professor Jon Atle Gulla, and then under Professor Kjetil Nørvåg.

I would like to thank Professor Jon Atle Gulla for his help and very helpful feedback in the initial phase of the thesis and Professor Kjetil Nørvåg for taking over the supervision and providing suggestions and valuable feedback throughout the semester.

Trondheim, 10th of June, 2008

Christian Bøhn

Contents

List of Tables

List of Figures

Part I

Introduction

Chapter 1

Introduction

With the growth of the internet, the web search engines have to deal with more and more documents. And the content they have to deal with spans all kinds of domains, yet the average users will usually specify a very short query using a very limited syntax. This makes it difficult to determine the domain of the users' interest. Another approach which is becoming more popular is to dedicate a search engine to a specific domain, known as vertical search. The advantage of this is that the search engine can make use of domain specific knowledge in order to improve the search quality.

News search is one kinda of specialization and is the domain that will be the focus of thesis. One property of the news is that it is always changing very rapidly. Regular web search engines may only visit a page every few days as they have to cover a verge large amount of sites. A news search engine on the other hand will in comparison only have to index and keep up to date content from a very small selection of sources. Secondly, a news search engine can easily be adapted to only index the actual content of an article and not all the surrounding text that is included on every page as part of the layout. A common property of news sites is the inclusion of a list of links to other news stories on every page, and a visiting web search engine which knows nothing about the structure of the site may mistake the text used in the links as content belonging to the page.

A second property of news search is that the news stories are often related to named entities, for example *United Nations*, *President Bush*, etc. The result of this is that the keywords that the users are looking for when searching are often named entities too, but a closer look at various news stories show that well known entities are often referred to with different names.

For example instead of *United Nations*, some news articles may write *UN*. The result of this can be seen in figure 1.1 and 1.2 where the first query was for *United Nations* and the second was for *UN*. What we see here is that the result set is different despite that both queries are referring to the same entity. We believe that, in a news search context when the users search for a named entity, that they are actually referring to the entity by any name. Instead Yahoo! News search, which was used in both the figures mentioned earlier, treated the queries as if the user was interested in a particular spelling. When looking closer at the two result sets we see that some of the topics referred to in the news articles are actually different.

What we are going to explore in this thesis is how we can automatically generate a dictionary of named entities and synonyms that are all referring to the same entity. With such a dictionary in hand we can then look at how we can make use of it to handle entities in a way so that the spelling of the entity becomes less important, making it so that the search engine can return potentially interesting news articles mentioning the entity, but with a different synonym.

1.1 Approach

As mentioned earlier, the motivation for this thesis is to explore ways of finding and making use of named entities in news search. In an earlier work[1] we explored the possibility of using query logs to generate a set of related entities, but the results were mixed. We were able to find some related entities but often they were only related, and not proper synonyms. Another issues was that unless we sacrificed the precision, we would only find very few entities.

In this thesis we take a different approach in finding entities. Instead of using query logs we use an unrelated data source, namely the well known and freely available encyclopedia, Wikipedia. What we find attractive about Wikipedia is that it is made up of a large amounts of semi-structured data and we think that it would therefore be a good candidate for data mining.

Our main contribution is based on using Wikipedia to automatically construct a large dictionary of named entities and various synonyms. The named entities are recognized and used as a basis for the synonyms extraction.

Using the named entity dictionary we then implement a modified search system that is based around query expansion. The idea is to include different synonyms if a named entity is used by the user, or if the entity is ambiguous, let the user select among a list of possible queries.

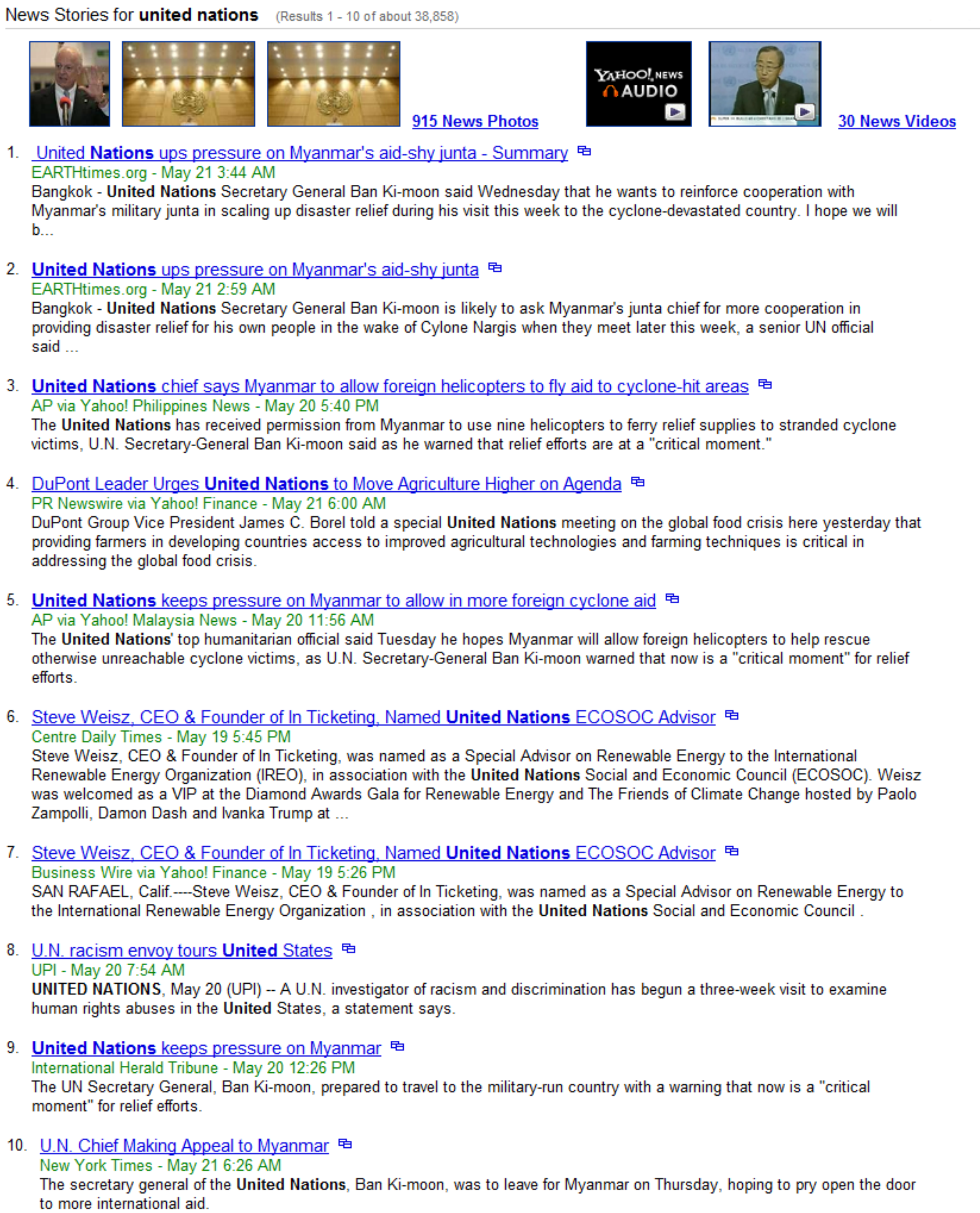
1.2 Results

The results from the named entity recognition shows that Wikipedia can be a very good source of named entities. In fact, it shows that with a very simple approach, we can find news related entities with a very high precision. The method yielding the highest precision would also classify the named entities as *people*, *organizations*, and *companies*, which we consider a bonus.

The results from the improved search system shows mixed results. While the precision is the same as the original system, the improved version seems to favor articles that were ranked considerably lower before due to the mix of synonyms.

1.3 Outline

In chapter 2 we give an overview of the technical concepts that are relevant to information retrieval, followed by a summary of various related work that is relevant to us in chapter 3. In



News Stories for **united nations** (Results 1 - 10 of about 38,858)

915 News Photos

YAHOO! NEWS AUDIO

30 News Videos

1. United **Nations** ups pressure on Myanmar's aid-shy junta - Summary
EARTHtimes.org - May 21 3:44 AM
Bangkok - **United Nations** Secretary General Ban Ki-moon said Wednesday that he wants to reinforce cooperation with Myanmar's military junta in scaling up disaster relief during his visit this week to the cyclone-devastated country. I hope we will b...

2. **United Nations** ups pressure on Myanmar's aid-shy junta
EARTHtimes.org - May 21 2:59 AM
Bangkok - **United Nations** Secretary General Ban Ki-moon is likely to ask Myanmar's junta chief for more cooperation in providing disaster relief for his own people in the wake of Cylone Nargis when they meet later this week, a senior UN official said ...

3. **United Nations** chief says Myanmar to allow foreign helicopters to fly aid to cyclone-hit areas
AP via Yahoo! Philippines News - May 20 5:40 PM
The **United Nations** has received permission from Myanmar to use nine helicopters to ferry relief supplies to stranded cyclone victims, U.N. Secretary-General Ban Ki-moon said as he warned that relief efforts are at a "critical moment."

4. DuPont Leader Urges **United Nations** to Move Agriculture Higher on Agenda
PR Newswire via Yahoo! Finance - May 21 6:00 AM
DuPont Group Vice President James C. Borel told a special **United Nations** meeting on the global food crisis here yesterday that providing farmers in developing countries access to improved agricultural technologies and farming techniques is critical in addressing the global food crisis.

5. **United Nations** keeps pressure on Myanmar to allow in more foreign cyclone aid
AP via Yahoo! Malaysia News - May 20 11:56 AM
The **United Nations**' top humanitarian official said Tuesday he hopes Myanmar will allow foreign helicopters to help rescue otherwise unreachable cyclone victims, as U.N. Secretary-General Ban Ki-moon warned that now is a "critical moment" for relief efforts.

6. Steve Weisz, CEO & Founder of In Ticketing, Named **United Nations** ECOSOC Advisor
Centre Daily Times - May 19 5:45 PM
Steve Weisz, CEO & Founder of In Ticketing, was named as a Special Advisor on Renewable Energy to the International Renewable Energy Organization (IREO), in association with the **United Nations** Social and Economic Council (ECOSOC). Weisz was welcomed as a VIP at the Diamond Awards Gala for Renewable Energy and The Friends of Climate Change hosted by Paolo Zampolli, Damon Dash and Ivanka Trump at ...

7. Steve Weisz, CEO & Founder of In Ticketing, Named **United Nations** ECOSOC Advisor
Business Wire via Yahoo! Finance - May 19 5:26 PM
SAN RAFAEL, Calif.----Steve Weisz, CEO & Founder of In Ticketing, was named as a Special Advisor on Renewable Energy to the International Renewable Energy Organization , in association with the **United Nations** Social and Economic Council .

8. U.N. racism envoy tours **United** States
UPI - May 20 7:54 AM
UNITED NATIONS, May 20 (UPI) -- A U.N. investigator of racism and discrimination has begun a three-week visit to examine human rights abuses in the **United** States, a statement says.

9. **United Nations** keeps pressure on Myanmar
International Herald Tribune - May 20 12:26 PM
The UN Secretary General, Ban Ki-moon, prepared to travel to the military-run country with a warning that now is a "critical moment" for relief efforts.

10. U.N. Chief Making Appeal to Myanmar
New York Times - May 21 6:26 AM
The secretary general of the **United Nations**, Ban Ki-moon, was to leave for Myanmar on Thursday, hoping to pry open the door to more international aid.

Figure 1.1: A query for *United Nations*

News Stories for **un** (Results 1 - 10 of about 31,541)

1,901 News Photos

62 News Videos

1. UN's Ban says will meet Myanmar junta leader
AFP via Yahoo! News - 30 minutes ago
UN Secretary General Ban Ki-moon said Wednesday he would meet Myanmar's reclusive junta leader during a high-profile trip to convince the country to accept a full-scale cyclone relief operation.

2. UN eyes talks after 22 killed in Sudan oil flashpoint
AFP via Yahoo! News - May 21 5:34 AM
The **UN** was optimistic about chairing talks on Wednesday to shore up calm in Sudan's flashpoint oil area of Abyei, one day after 22 government troops died in fighting that threatened the peace process.

3. UN chief to visit Myanmar to press leader for more access
International Herald Tribune - May 21 5:48 AM
UN Secretary General Ban Ki Moon was en route to Myanmar to seek a visit with the leader of the ruling military junta and to help coordinate delivery of increased **UN** aid to cyclone survivors.

4. UN's Ban to try to convince Myanmar on aid
AFP via Yahoo! News - May 21 12:07 AM
UN Secretary General Ban Ki-moon was headed Wednesday for Myanmar, looking to convince the generals who have snubbed his phone calls to accept a full-scale relief operation for Cyclone Nargis.

5. Downer wanted for UN mission
News Interactive - 2 hours, 55 minutes ago
FORMER foreign minister Alexander Downer has been approached for a senior job with the **UN** to revive the Cyprus peace process.

6. UN chief says he will meet Myanmar's military leader
Channel NewsAsia - 17 minutes ago
YANGON: **UN** Secretary General Ban Ki-moon said on Wednesday he would meet Myanmar's military leader during a high-profile trip to convince the country to accept a full-scale cyclone relief operation.

7. UN urges Myanmar junta focus on saving lives; plans to visit Thursday
The Canadian Press via Yahoo! Canada News - May 21 9:04 AM
BANGKOK, Thailand - **UN** Secretary General Ban Ki-moon urged everyone involved in Myanmar relief efforts to focus on saving lives not on politics Wednesday a day before the start of his planned visit to the cyclone-ravaged country.

8. Fighting rages in Sudan as UN urges ceasefire, US seeks solutions
AFP via Yahoo! News - May 20 1:00 PM
Deadly fighting raged Tuesday in Sudan's flashpoint oil area of Abyei between government forces and southern troops, threatening a fragile peace process and sparking a truce call from **UN** chief Ban Ki-moon.

9. Burma junta seizes UN aid shipments which would have fed 95,000
Daily Telegraph - 1 hour, 24 minutes ago
Burma's ruling military junta has seized the only two **UN** aid shipments to make it into the devastated country, which would have fed 95,000 people.

10. UN's Ban says will meet Myanmar junta leader
TODAYonline - 1 minute ago
Myanmar's junta leader Senior General Than Shwe (C, uniformed), along with top military brass, inpects relief goods provided to cyclone-affected families on the outskirts of Yangon on May 18. **UN** Secretary General Ban Ki-moon has said he will meet Myanmar's reclusive junta leader during a high-profile trip to convince the country to accept a full-scale cyclone relief operation.

Figure 1.2: A query for *UN*

chapter 4 we give a more detailed description of the approach we took in this thesis, followed by a description of our implementation of an automated named entity extractor and a simple search system making use of these entities in chapter 5. Chapter 6 gives an overview of how we evaluated the results followed by the results themselves, followed by a discussion of the results and different improvements we believe would improve the results in chapter 7. The last chapter is the conclusion, which is found in chapter 8.

Part II

Theoretical Background

Chapter 2

Technological Overview

In this chapter we give an overview of some of the technological background which is relevant to this thesis. We start out with an overview of information retrieval and then take a closer look at a common ranking model. Next we look at text preprocessing techniques commonly used in information retrieval, before we look at some data mining techniques.

2.1 Information Retrieval

Information retrieval is the field of retrieving relevant information about a subject and then present the information nicely ordered by relevance. A retrieval process starts with the user who must translate their information request into a query which can be sent to the retrieval system. A query is made up of one or more terms and usually expert users are better at specifying more terms. Few terms make it difficult for the system to determine which documents are the best answer to the user query. Next, the information retrieval system must match the query terms against all the information in its index. This is where information retrieval is differentiated from data retrieval. An information retrieval system focuses on finding relevant information and presenting the most relevant information first, a data retrieval system on the other hand aims at finding all objects that satisfy clearly defined conditions[2]. The main problem of information retrieval is therefore to determine what constitutes the most relevant documents based on a very limited user query, and especially in the case of web search the search system has to deal with what contexts the user is interested in.

Information retrieval systems are usually built around indexes where all the terms that occur in the indexed documents are stored together with occurrences of each term to speed up the process of finding documents that contain the query terms. The system does not have to read all the documents, but can instead get lists of documents matching the different query terms. Unfortunately from a user's point of view, this result set can be quite huge, and the user is rarely interested in manually going through hundreds or thousands of possibly relevant documents to find the ones they are most interested in. To help with this, different retrieval models are used in order to try to determine which documents will be the most interesting ones. The simplest retrieval model is the boolean model and does not actually provide any ranking of the result set. Instead it will only include the documents that include all the required terms, and exclude any documents that have any unwanted terms. Users rarely browse more than the first few pages of the result set. To overcome this, two different retrieval models are usually used instead; The *vector space model* and *probabilistic models*. In this section we will have a closer look at the *vector space model* works.

The vector space model aims to give a more meaningful ranking of the documents by measuring the similarity of the query terms versus the documents found in the document collection. This is done by assigning weights to both the index and the query terms. The weights are a product of the term frequency and the inverse document frequency. The term frequency is calculated based on term occurrences in the document only, and is given by this formula:

$$tf = \frac{term\_freq}{max\_freq}$$

where $term\_freq$ is the frequency of a given term in the document, and max_freq is the frequency of the most frequent term in the document. This means that the more frequent a term is within a document, the more important it is considered to be. A problem with only weighing a document based on the frequency of a term within the document is very visible in the cases where multi-terms are issued. If one of the terms is very popular throughout the document collection it can dominate the lesser popular one, despite how the lesser popular one can be a better discriminator between relevant documents and non-relevant ones. To counter this, the inverse document frequency is applied and is used to decrease the weight of terms that are globally popular. The formula for this is given by:

$$idf = log\frac{n}{n_t}$$

where n is the total number of documents in the document collection, and n_t is the number of documents that have the term in them. The weight of a term is then given by

$$w = tf * idf$$

and is this formula can be used to weight both the index terms and the query terms.

At last the similarity between the query terms and a document must be found, which is done by combining the *tf-idf* scores of both query terms and index terms and given by the formula:

$$sim(d,q) = \frac{\sum w_{i,j} * w_{i,q}}{\sqrt{\sum w_{i,j}^2} * \sqrt{\sum w_{i,q}^2}}$$

2.1.1 Vertical Search

Vertical search is one approach in information retrieval where the information retrieval system is geared toward a specific domain. Web search on the other hand is a horizontal approach to search, where the goal is to index as much as possible, with as wide a selection of topics as possible. This makes the systems different in the approaches that can be taken to improve the relevance of the results. In vertical search engines domain knowledge can be applied to improve the search results, and they will try to limit their index to only domain relevant documents. For instance, a vertical search engine have the advantage of being able to return only relevant results where query terms have a very different, but popular meaning onside the domain targeted by the vertical search engine. The vertical search system will simply not include all the documents that fall outside the domain.

2.2 Text Preprocessing

In this section we discuss some text preprocessing techniques that are commonly used in information retrieval systems.

2.2.1 Tokenization

Tokenization is the process of converting a character stream into tokens by detecting word boundaries. In the most basic form this would be to split the characters in an input character stream whenever a space is found. In more advanced forms it has to take into account hyphens, punctuations, etc. This is something that can be easily done by humans, but is more challenging when it has to be done by a computer[3]. For instance, a punctuation can be part of an abbreviation and not a sentence ender. In that case it should not split the characters. Another complication is that some language do not even use spaces to separate words.

2.2.2 Normalization

Normalization is a technique which is applied to terms to minimize the number of variations of a word. In natural languages words often appear in multiple forms, plurals, gerund, past tense, etc. These syntactical variations may result in poor recall if an exact match against the query keywords is required. Stemming and lemmatization are two commonly used methods used to normalize words which we will look at a bit closer.

Stemming

These forms usually share a common root, or stem [2]. The stem is what is left after any prefixes or suffixes have been removed, and if the stem is used instead of the whole word, the number of variants are often reduced. One example given in [2] is the word *connect* which can be the stem for *connected*, *connecting*, *connection*, and *connections*. In [4], four stemming algorithms are described; affix removal, table lookup, successor variety and n-grams. According to [2] the most popular stemming algorithm implementation is the Porter stemmer which is a suffix removal algorithm. The Porter stemmer uses suffix lists, and applies a series of rules in a specific order. One disadvantage of stemming is that the stem may not be a real word.

Light Stemming

Light stemming is a simpler variant of stemming where only plural forms are stemmed. For example *papers* becomes *paper*. The advantage of light stemming is that it is very easy to perform precisely and it results in some reduction of variants. This gives us a smaller reduction in index size, but at the same time it improves the recall slightly and it reduces the risk of having unrelated words being reduced to the same stem.

2.2.3 Lemmatization

Lemmatization is another way of translating words into their normalized forms. It is usually based around dictionary lookups, which is different from stemming algorithms that are usually rule based. They may also take in account the context of the word when determining the correct normalized form. An advantage of lemmatization is that it will give valid words, but that comes at the cost of having to know all the words in advance. If the word is not in the dictionary, lemmatization can not be performed. A possible solution to this is to use a hybrid solution where lemmatization is first applied, and if it fails, the normalizer will fall back to using a stemmer.

2.2.4 Stop Word Removal

Stop words are frequently used words that by themselves hold very little meaning and because of their frequency are not very good at discriminating between concepts. Removing them can reduce the index size dramatically, by as much as 40%[2]. But the removal of stop words can also result in weird effects from the reduced recall. For example *The Who* is the name of a group, but both of the words used are usually considered to be stop words. This can make it impossible to find documents about the group, which was very visible with Google in its earlier days when such a query would return an empty result with the message that both query terms had been removed due to being too common. Another problem is that common stop words like *may*, *can*, and *will* are homonyms for rarer nouns[5]. For example *may* is a stop words, but it is also the name of a month, *May*.

2.3 Ontologies

The term *ontology* has multiple meanings, depending on the context. Within computer and information science it is considered to be an agreement on a domain specification[6]. The purpose of an ontology is to provide a shared understanding of a domain, and how the encapsulated concepts are related to each other[7].

2.4 Data Mining

Data mining can in simple terms be described as extracting or "‘mining"’ knowledge from large amounts of data [8]. Another used term for data mining is *Knowledge Discovery from Data*, or *KDD*, which may be less misleading as to the real goal of data mining. [8] describes the process as a 7 step process, where the first step is to remove any unwanted data, often referred to as data cleaning. Next, if the data is coming from multiple sources, it has to be combined before the data relevant to the mining goals is selected. Before the actual mining, the data has to be transformed into a suitable form to ease mining, and after the mining is done, pattern evaluation is done to extract the really interesting bits. Finally, the data has to be presented in a meaningful way.

A data mining algorithm is usually trying to classify data into sets of predefined groups, cluster data into groups based on logical relations, identify associations between data, or identifying sequential patterns. One classification method is decision trees where a decision tree is constructed using a set of training tuples. The construction algorithm uses the training data to select good attributes to partition the data as good as possible.

Another method for classifications are *Markov chains*[8] which generates a sequence where the probability of the next state only depends on the previous state. This makes it memory less, as it only has to keep a one step history. The sum of all possible transitions from one step to the next must be equal to 1. A *Hidden Markov model* is a variant where the parameters are unknown, and the goal is to determine these parameters.

Chapter 3

Related Work

In this chapter we take a closer look at projects we think are somewhat related to our work in the terms of using Wikipedia related to information retrieval, recognizing entities, and using dictionaries in search engines.

3.1 Accessing Wikipedia

In [9] the authors evaluate the usefulness of Wikipedia as a lexical semantic resource, and compares it to more traditional resources, such as dictionaries, thesauri, semantic wordnets, etc. Their motivation is that Wikipedia has most likely become the largest freely available collection of knowledge. As of March 2008, there are more than 2.2 million articles included in the English version [10]. Next they find that Wikipedia is one valuable property that is not found in regular encyclopedias. Heavy linking is used to connect different entry together and make it easy to find a more descriptive page of entries mentioned in the text. Secondly it is not restricted in size.

They point out four possible sources of lexical semantic information found in Wikipedia:

- Articles
- Article links
- Categories
- Disambiguation pages

In each articles, the first paragraph can be considered a definition of the subject, and the full text as a more thorough description of the meaning. Some articles do not hold any content themselves, except for a redirect to another article. These redirects are useful for finding synonyms, spelling variations, common misspellings and abbreviations of article topics. Last, they found the titles to often be made up of nouns, which are useful for named entities or domain specific terms.

The second useful source is the article links. Each link consists of a target article, as well as an option label which defaults to the title of the target article. This can be used for finding context of related terms, and the link labels are another source of synonyms and spelling variations, as well as related terms. Third, the targets can be used to construct a link graph, useful for finding related terms.

3.2 Using Wikipedia for Named Entity Disambiguation

In [11] the authors look at using Wikipedia for detecting and disambiguating named entities in open domain text. Their motivation is to improve search quality by being able to recognize entities in the indexed text, and disambiguate between multiple entities that share the same proper name by making use of the context given by the text. Then during searches they want to group results according to sense rather than as a flat, sense-mixed list. That would give the users access to a wider range of results as today's search engines may easily favor the most common sense of an entity, making it difficult to get a good overview of the available information for a lesser known entity.

The first step they took was to build a dictionary of entities where they use the titles Wikipedia entries as the primary names for the entities. This process is somewhat complicated unfortunately, as every article title in Wikipedia is required to begin with a capital letter. Therefore it is not possible to use the first letter in the title to differentiate between proper nouns or not. Instead they present a simple, three-steps heuristics:

- If multi word title and every word is capitalized, except prepositions, determiners, conjunctions, relative pronouns or negations, consider it an entity.
- If the title is a single word, with multiple capital letters, consider it an entity.
- If at least 75% of the occurrences of the title in the article text itself are capitalized, consider it an entity.

Next, they use the redirect pages to find alternative names for the entities, and disambiguation pages are used to identify different entities that all share the same proper name.

To evaluate their disambiguation engine, they make use of the hyperlinks embedded in all the articles. Each hyperlink has a label which defaults to the article title of the target article, unless an optional alternative is specified. For the evaluation they use the alternative label strings as queries and the target article as the answer.

3.3 A Semantically Annotated Wikipedia XML Corups

In [12] the authors present their system, YAWN, for converting Wikipedia into semantically annotated articles. Their motivation is to open up for a more advanced query syntax, making it possible to use semantically rich structural queries that are very precise in what they are looking for like //person[about(//work,physics(and about(//born,Germany)] to query Wikipedia.

While the entire collection of Wikipedia articles are available in XML, the articles themselves do not use XML to structure the content. Instead, XML is only used to bind the article text together with the title and various meta data. The structure of the articles is specified using a Wikipedia specific markup language.

The first step taken by YAWN is to convert the Wikipedia markup into a more formally specified XML format, with the goal of capturing some of the semantics inherited in the articles. In doing so they run into problems of having to deal with HTML markup being mixed in with the Wikipedia markup, as well as the flexibility of the markup language which allows different components to be combined arbitrarily. The Wikipedia HTML generator, which converts Wikipedia markup into HTML is fault-tolerant, allowing the users to specify incorrect or inconsistent markup. This causes problems when attempting to convert the articles into well-formed XML where the rules are very strict.

3.4 Different Approaches to Named Entity Recognition

Named entity recognition is nothing new, but traditional the focus has been on recognizing named-entities embedded in text. Different approaches have been taken to do this, both rule based and statistical variants have been made. In this section we will look closer at some of the used approaches.

3.4.1 Rule Based

Rule based systems rely on predefined rules in order to recognize proper names and classify them based on entity categories such as *people*, *organization*, *location*, etc. One such system is the Proteus system [13][14] which make use of local contextual cues such as capitalization, prefixes and suffixes. For example *Mr.* or *Ms.* followed by a capitalized word indicates a person, or if a capitalized word is followed by *Inc.* or *Co.*, it follows the pattern of a company name. This gives a system that is heavily reliant on its authors ability to discover the various patterns during its design [15]. One problem with this approach is when entities make use of some of the cues as part of its name, for example *Mrs. Field's Cookies* is a corporation, not a person.

3.4.2 Decision Trees

A decision tree can be considered to be composed of three elements[16]; futures, history, and questions. The futures are the possible outcomes, and in the case of named entity recognition, it is the classification of each entity. For instance this can be different categories. The history is the data available to the model[15] and can be the current word, or any of the words before and after. The questions are the split points used in growing the tree. When growing the tree, the most determining ones should be at the top.

3.4.3 Hidden Markov Models

Hidden Markov Models were used in [17][18]. Conceptually they build a Hidden Markov Model with 8 states two special states. The 8 states represents the different categories, including a *not-a-name* category used for the non-entities. For each entity-class state they use a statistical bigram language model. On mixed case input they are able to achieve an F-measure of 90%, but if the text is converted to upper case this drops by 3%.

3.4.4 Maximum Entropy

Maximum entropy is another statistical method original described in [19][20] and within information retrieval it has been used in different settings. [15] suggests using it for named entity recognition and presents a system that was well above the median of other systems in the MUC-7 named entity evaluation, but is beaten by the best handcoded systems. The advantage of their system is shown in its portability. Without any knowledge of Japanese, the authors were able to port their system and archive results that could compete with the systems created by native speakers.

Similar to decision trees, the maximum entropy approach makes use of *futures*, *histories*, and *features*. The *features* are used to predict the *futures*, which are the possible outcomes or categorizations. The *history* is used to assign probabilities to the possible *futures*.

3.4.5 Combining WordNet and Wikipedia

WordNet is a large lexcial database of English where nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms, or *synsets*, expressing distinct concepts[21]. Synsets are linked through a hierarchal structure, based on conceptual-semantic and lexical relations. In [22], Magnini et al. describes a method where WordNet is used to create a collection of named-entities, grouped by categories such as *person*, *location*, *organization*, etc. Their method is based around capturing external and internal evidence, where internal evidence are the words in the text that are considered to be an entity and the external evidence are the surrounding sentence. First an part of speech tagger is applied to find the nouns, together with multiword recognition. Basic rules are then applied to tag potential named-entities before high level rules are used to clear up any ambiguity.

Toral et al. used this as the basis for their named-entity extraction in [23]. They use the first sentence of Wikipedia articles as the source for their named-entity recognition and apply it to a randomly selected subset of 3517 articles. The resulting named-entities they find are split among three categories: location, organization and people. Location makes up 58% of the resulting 695 entities, with 34% being people and only 8% are organizations.

3.5 Query Expansion

Query expansion is a technique where the users' queries are expanded with additional keywords with the goal of improving recall or improving precision. Some query expansion systems are automatic while others rely on user feedback, such as [24] where the users are provided with upto 100 refinement suggestions. Their algorithm is based around precomputed suggestions for single term queries, and yield similar results to the much slower DM algorithm.

In [25] the authors use WordNet for query expansion of geographical terms. They start off by identify proper nouns in the query string and when a proper noun is found, WordNet is used to expand the query with any *country*, *state*, or *land* related synset. This turns the users' queries into new queries where the original query strings have been ORed together with synonyms. As a result they found that while the recall was slightly the improved, the precision suffered.

In [26] they take a different approach where instead of expanding the query, they expand the index. They still make use of WordNet, this time to find holonyms of geographical locations. A holonyms of A is a concept which encapsulates A among other concepts. In a geographical context *Trondheim* is part of *Norway* and therefore documents mentioning *Trondheim* are related to *Norway* even if *Norway* is not explicitly mentioned in the text. After the geographical locations are expanded to include holonyms, they are separated from the rest of the text resulting in a geographical index and a text index. This is done to prevent other entities sharing the same name of a location to match queries for the graphical location. For example *John Houston* should not match a query searching for the city in Texas, *Houston*. The results are similar to [25], where the recall was slightly improved at the cost of the precision.

Another approach is taken in [27] where WordNet is use for sense disambiguation where their goal is to improve precision. WordNet is used to list all senses of the user query, and also to find conceptual relations which are used in the reformulated query. For example, if the users are searching for *Java*, they give them the option between returning documents of all related senses, or refined to only the island, coffee, or the programming language. They also implement a version where the results are classified based on the different senses, but find that the query rewriting approach is yielding better results. Another problem with the classification of the result set is that it can become rather large.

3.6 Entity Recognition and Normalization

Instead of using query expansion to include synonyms, entity normalization can be used to change the ambiguous entities into non-ambiguous ones. [28] uses an automatically constructed a named entity dictionary of genes and proteins. The entity dictionary is built from five existing databases where all aliases for unique identifiers are added together. From their earlier work[29] they had noticed that most synonyms were simply orthographic variations. Using the synonyms extracted from existing database they create missing variants of hyphens replaced by space, space replaced by hyphens, etc. After the entity dictionary was complete, they used it to normalize the documents before indexing. Their idea was that ambiguous terms would also have a non-ambiguous reference earlier in the document. If one such unique reference was found, the ambiguous ones would be replaced by the unambiguous one.

Part III

Realization

Chapter 4

Approach

In this chapter we describe the approach taken to build a prototype of extracting named-entities from Wikipedia for us in a news context.

4.1 General Idea

The origin of this thesis was to improve the handling of entities when used in news search. The idea was that when users are querying for named entities that appear in the news, they are not really only interested in the hits that use exact same name of the entity as they did.

Instead, we believe they are interested in finding articles about the entity itself independent of which synonym the article is using to refer to the entity. The problem is made up of two steps, a collection of entities and their synonyms must be found before they can be used. In [1] we tried finding these synonyms by looking at query logs. We used query log aggregates with a granularity of 1 week, which seemed to be insufficient. In the end we were only able to produce a very small set of entities without sacrificing precision.

For this thesis we decided to try to build a collection of named entities from a different source. One option we considered was to use dmoz[30], which is an open directory project. While we believe it could have been a useful source of entities, it did not seem to provide very much in the way of synonyms. Instead we took an interest in Wikipedia, which is another project that freely provides a full dump of all its content for download.

4.2 Wikipedia

In this section we will look at some the Wikipedia features that make it attractive as a mining source when building a large collection of named entities. There are four features that are in particular interest to us: internal links, redirects, disambiguations, and categories. In the following sections we will shortly discuss how me intend to make use of them.

The Wikipedia dumps themselves are freely available for download at http://download.wikipedia.com. The dumps are available in different versions, where the difference is in how complete they are. The most complete dumps contain the entire editing history of every article, as well as user pages and talk pages. Because of this, the complete dump file is very huge at about 150GB compressed and almost 3TB uncompressed. Since we are not interested in the editing history, nor the user and talk pages we can use the lighter versions, which contains only the latest version of each article. Due to this is is only 3.5GB compressed and around 12GB uncompressed. The dumps are also available in multiple languages, but we are only interested in the English version.

Internal Links

Internal links are used to link words in one article with another article, thereby making it very easy for the users to find more information about a specific keyword mentioned in the article text. Listing 4.1 shows an excerpt of the article text, including wiki markup, from the article about the *Shortest path problem*. The internal links are enclosed in Wiki-markup, *[[* and *]]*. Inside the tag an label of the link can be set to another value than the default label which is the name of the target article by separating the target name from the link label using a |.

We intend to collect all links pointing to the same article and then aggregate them based on the label to find synonyms as well as the popularity of that synonym. But not all tags that appears to be links behave by inserting a linked piece of text in the rendered page. The target article can be prefixed with a namespace, and the namespace can be a language code. In this case, the tag is used to indicate that the article is available in a different language, and the language represented by the language code is automatically added to the list of versions of the page available in different languages. Secondly, if the namespace is *Category*, it means the article is a member of the category rather than it trying to link to a category page. Actually, to make it a link from an entry to a category page, the category namespace has to be prefixed with a colon.

Listing 4.1: Wiki markup for the *Shortest Path Problem* article

```
In [[graph theory]], the '''shortest path problem''' is the
   problem of finding a [[path (graph theory)|path]] between
   two [[vertex (graph theory)|vertices]] such that the sum of
    the [[Glossary of graph theory\#Weighted graphs and
   networks|weights]] of its constituent edges is minimized.
```

Redirects

Redirects are almost similar to links, except that they can not include an alternative text. We intend to use them as another source of synonyms or alternative spellings of entities, as was done in [9]. A difference between redirects and links are that the links pointing to different articles can share the same display text, but a redirect can only redirect to a specific article.

This makes the redirects less ambiguous. An example of a redirect is shown in listing 4.2. It is used to redirect *Shortest path* to *Shortest path problem*.

Listing 4.2: Wiki markup for the *Shortest Path*

```
#REDIRECT [[Shortest path problem]]
```

Disambiguations

Disambiguation pages are used by Wikipedia to resolve conflicts between terms having multiple senses [31] by either listing all the senses for which articles exist, or treat the dominant sense as the primary article, and then presenting a small link to less popular senses. An example of an ambiguous term is *Mercury* which can refer to both the element and the planet as all Wikipedia article titles start with a capital letter. An excerpt of the disambiguation page for *Mercury* is shown in listing 4.3. Sense unique titles are made by attaching the domain as a suffix.

Disambiguation pages are easy to detect. They have the suffix *(disambiguation)* attached to the title in cases where one sense is considered the primary meaning. Another identifier is the *{{disambig}}* template which is usually referred to from disambiguation pages to automatically add some standard text.

Listing 4.3: Excerpt of the disambiguation page for *Mercury* including wiki markup

```
'''Mercury''' commonly refers to:
* [[Mercury (planet)]], the nearest planet to the Sun in the
  solar system
* [[Mercury (element)]], the chemical element (also called ''
  quicksilver'')
* [[Mercury (mythology)]], a Roman god

____
'''Mercury''' may also refer to:

== Architecture ==
* [[Mercury City Tower]], a Moscow, Russian skyscraper
* [[Mercury Court]], a large office building in Liverpool,
  England

{{disambig}}
```

Categories

Categorization is used to group one or more article together, and every article should be a member of at least one category[32], but this is only encouraged, not required. The categories that a page is a member of are always shown at the bottom of the article, and can help the users in finding other articles related to the domain. Listing 4.4 shows the markup used to include the article *Shortest path problem* in the categories *Graph theory* and *Polynominal-time problems*. The categorization system is flexible as it is not limited to a tree structure, instead it is a direct cyclic graph. While avoiding cycles is encourage, it is not enforced by the software and therefore some cycles exist. This may make it difficult to determine which category is the parent category and which one is a sub-category.

Listing 4.4: Categorization of the *Shortest path problem* article

```
[[Category:Graph theory]]
[[Category:Polynomial−time problems]]
```

4.3 Prototype Overview

As part of this thesis we are making a simple prototype system made up of two components. A named entity mining component that is used to extract entities from Wikipedia in order to build a dictionary of named entities, with a list of common synonyms for each entity. It will also be used to build a list of ambiguous entities, that is, entities with multiple senses. For example, *Bush* may refer to *George W. Bush* as well as *George H. W. Bush*. To experiment with the use of the automatically constructed named entity dictionary we are making a search system which is making use of the dictionary to do query expansion.

4.4 Named-Entity Recognition

Originally we considered using capitalization of words to find entities. The idea was that if all the words in an article title were capitalized we were most likely dealing with a proper noun, or a named-entity. The problem was that all article titles have their first letter capitalized even if they are nouns rather than proper nouns. This means we would be unable to use this to find single-word entities, as well as entities containing a non-capitalized words like *Prince of Wales*. To work around this we considered making use of the categories and try to find categories that seemed like they were mainly made up of entities and then mark the remaining entries in the category as entities, as an alternative to the capitalization requirement.

Instead we decided to use an algorithm similar to [11] which also builds upon the idea of looking for capitalized words. But to handle single-word titles or small uncapitalized words, they look at the occurrences of the title in the text itself. In other words, if the number of instances of the title that are exactly matching the title is above a certain threshold then the title is considered to represent a named entity.

Since we are especially interested in finding entities related to news, we thought of a much simpler algorithm as well, aimed at just finding people, companies, and organizations entities. Instead of trying to look at the categories as some kind of tree and try to find a node that represents all entities of a kind, we looked at how the category names often follow certain patterns when multiple categories are related. For instance, there are multiple category groups that follow a *Companies based in xxx* pattern where *xxx* is a geographical location. We believe that this can possibly be very useful for gathering a large collection of entities related to a few groups. Also, this collection of entities will be useful in evaluating the recall of the more generic entity recognition algorithm.

4.5 Synonym Extraction

After a set of named entities have been identified, we want to find their synonyms. As described earlier we intend to use the internal links, redirects and disambiguation pages for this, and we can easily extract all of these after we have the named entities. This will give us a list of captions, all used on links to a particular entity, which is likely to generate a various amount of junk synonyms. That is, synonyms that are not really synonyms, but instead the result of people vandalizing articles. The second type of noise are link captions where a noun has been appended to the proper noun which it is linking to. For example *Bush administration's* is linking to the article about *President Bush*, yet it is not a good synonym.

To filter out the noise we have considered two options, one is weighting each link caption based on the number of links using the same caption. Then we can filter out the less popular ones, which are less likely to be good synonyms since they are used infrequently. Another option we considered was to apply the same algorithm used to the classification of link captions, where we use versions of the link captions that are capitalized in different ways as an alternative since we have no article text.

4.6 Query Expansion

The motivation for wanting to automatically building a dictionary of named entities was to put it to use in a search context. What we wanted to do was to use it to perform named entity disambiguation and improve robustness. By disambiguation we simply want to be able to recognize entities in order to detect when a user specifies a query that is ambiguous, i.e. it can be a reference to multiple entities. For example *Jennifer* can refer to both *Jennifer Aniston* and *Jennifer Lopez*. The second usage we had in mind was robustness, by which we mean being able to improve the recall when entities are referred to using different names in the query and the documents. For example *United Nations* and *UN* are two names that are both referring to the same entity, yet some news articles will use the full version while others use the abbreviated one. In the case of news search, we believe the user is more interested in stories about the entities referred to than stories where the entity is using the exact same spelling as in the query.

One approach to making use of the entity dictionary would be to perform entity normalization before indexing. That is, we would translate all occurrences of an entity into its main entity reference if we can determine which entity the document is about, or a list of the unique names of multiple entities if there are no unambiguous references in the text. The problem is that to do this we need to have the original news articles and the ability to normalize them before indexing.

Instead we tried to use query expansion by expanding the query to include multiple synonyms. By taking the users' original query we can determine if it is referring to one or multiple entities in our dictionary. If we have multiple matches, we will present the user with a list of possible queries that would be less ambiguous, while at the same time present the user with the original result set. In the other case, where the query is uniquely referring to an entity and we can expand the query to include synonyms.

4.7 Web Application

A web application was made to provide a simple user interface to the search system. It uses the dictionary for query expansion if the query given is a known entity. If the system is given an ambiguous entity, the system will list possible unambiguous entities with the most popular entities first. The most popular entities will be determined based on how each of the synonyms are when appearing in the link captions. The modified queries are then run against the original search engine, and the results are extracted an presented to the user.

Chapter 5

Implementation

In this chapter we describe the system we implemented for extracting named entities and synonyms from Wikipedia and how we used the resulting dictionary in a modified search system.

5.1 Overview

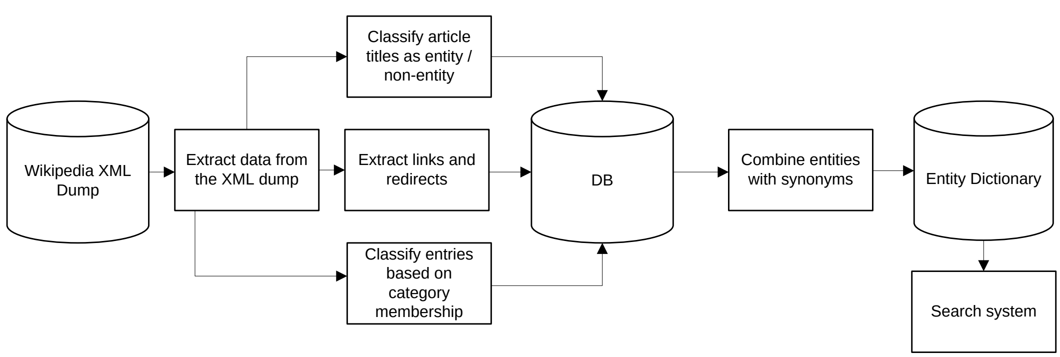


Figure 5.1: System overview

An overview of the system design is shown in figure 5.1. The entries are extracted from the Wikipedia XML dump and are then use for the following processes:

- Classify the entry titles as entity/non-entity
- Extract all the links and redirects

- Extract the category memberships for each page

All these processes are independent and the output of each of them are therefore stored on disk for later when we want to use the links to find synonyms for the recognized entities.

The named entity recognition and synonym extracting was implemented in Ruby, with small amounts of c++ used in performance critical spots to speed up the processing, making it easier to do multiple test runs with different parameters. Various libraries were used, sqlite for storage along a simpler flat-file format used for reading and storing data sequentially, libxml was used to parse the Wikipedia XML dump, Hpricot was used to extract search results from the HTML returned by the original search engine, ferret was used to index and search the named entity dictionary. The search application was implemented as a web interface using Ruby on Rails.

5.2 Constraints

The scope of this thesis is limited to news search, which limits our interest in entity categories to a few we consider very relevant to news. It also means that we have made assumptions that we do not would hold true in a general sense.

Secondly, we did not have direct access to a news index except through the public front end. This limited us in how we could use the entity dictionary to modify the search results as we could only issue queries like everyone else, with a very limited query syntax. We could not use normalization on the news articles before they were indexed. One reason why we would have wanted to try this is because of how the ranking algorithms work where word frequency within the document and the document collection affect the results. The idea was that if the articles were normalized then synonyms would not be considered more important than others simply before their use was less frequent.

5.3 Preprocessing

The Wikipedia XML dumps are available in multiple versions, with the largest version containing the entire editing history for every article, as well as user and talk pages. As we are only interested in the title and the article text, we used smaller dump. The one we used was the January dump of 2008 that was available at http://download.wikimedia.org/enwiki/20080103/enwiki-20080103-pages-articles.xml.bz2 at the point of writing.

In listing 5.1 an excerpt from the dump file used is shown. Of all the information, we are only interested in the <title> and <text> tags of the latest revision of the entry, and we stored the information of interest in a binary file to save us from having to parse the XML file every time we wanted to rerun any of the processes that took the Wikipedia entries as input. The entry shown in the excerpt is a redirect that is use to point an entry using an old naming convention to the new entry title with the new naming convention.

Listing 5.1: Wikipedia XML dump extract

```
<mediawiki xmlns="http://www.mediawiki.org/xml/export-0.3/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.mediawiki.org/xml/export-0.3/␣http://www.mediawi
version="0.3" xml:lang="en">
  <page>
    <title>AlbaniaGovernment</title>
    <id>35</id>
    <revision>
      <id>74467128</id>
      <timestamp>2006-09-08T04:19:45Z</timestamp>
      <contributor>
        <username>Rory096</username>
        <id>750223</id>
      </contributor>
      <comment>cat rd</comment>
      <text xml:space="preserve">#REDIRECT [[Politics of Albania]]
      {{R from CamelCase}}</text>
    </revision>
  </page>
  <page>...</page>
</mediawiki>
```

5.4 Generic Named Entity Recognition

The generic named entity recognition is only classifying a Wikipedia entry as an entity or not. It starts out by looking at the title of the entry, since as mentioned earlier, most of the article titles are nouns, and the only nouns we are interested in are the proper nouns. To classify the entries we implemented an algorithm using the following steps when given a title, T, and the text of an entry:

1. Remove any domain suffix from T
2. Tokenize T into n units, $w_1, w_2, ..., w_n$
3. Remove any w_i from W where w_i is included in S
4. Classify as an entity if any of these conditions holds true:
    - $\sum C(w_i) = n$ and $n >= 2$
    - $\sum D(w_i) >= 2$
    - $\frac{\sum E(T)}{\sum N(T)} >= \alpha$

A domain suffix is the text enclosed in parentheses that follows the title of entries with multiple senses. They are used to disambiguate between the senses, but since they are not part of the

Article Name	Is Named-Entity?
Esoteric knowledge	false
Princess of Wales	true
Doñana National Park	true
English literature	false
Single occupancy vehicle	false
High occupant vehicles	false
High occupancy vehicles	false
High-occupancy vehicle lane	false
Clinton County	true
DeWitt Clinton	true

Table 5.1: Excerpt of the named-entity recognition output

entity name, we must first strip them from the title. Next we strip all w_i which are found in S, which is a list of stop words. See appendix B for the list of stop words used[33]. The classification makes use of multiple functions:

- C=1 if any $l_i \in [A..Z]$, 0 otherwise
- D=1 if $|Q| >= 2$ where $Q = \sum C(l_i)$, 0 otherwise
- D returns 1 if the parameter has multiple capital letters, 0 otherwise C is a function that returns 1 if the parameter is capitalized, and 0 otherwise, while D is a function that that returns 1 if the parameter has multiple capital letters, and 0 otherwise. α is a variable used as a threshold for the third condition.

An excerpt of the output generated is shown in table 5.1. It shows the titles of a random selection of a few Wikipedia entries and whether or not they were classified as entities using the algorithm described above.

5.5 Category Based Named-Entity Recognition

The category system can also be used to to perform entity extraction. As mentioned earlier, the Wikipedia categories form a directed cyclic graph, which makes it more difficult to find nodes in the category graph that designates that all sub-categories are people, organizations, or companies. Since it does not follow a tree structure, we risk running into cycles, which could turn the remaining of a graph into a sub-category of a choose parent node. Since we intended to use the extracted entity dictionary in a news context, we picked three categories of entities we thought were highly relevant:

- People
- Organizations

- Companies

The first entity category is easy to find entities for, as there is a category named "‘Living people."’ This category exists mainly because living people may suffer harm if wrongful information is attributed to them, and therefore these pages must be watched more carefully[34] than other pages. This makes it a very useful category to us as it should cover most people who are news relevant.

The second and third entity categories are more difficult to extract as there are no superior category for either of them which are used to indicate that all children are either organizations or companies. Instead we ended up using pattern matching to identify categories holding entries that would fit under the respective named entity categories.

Using simple wildcards we found category patterns that matched categories that are made up of entities, as seen in table 5.2 where the patterns we used are listed.

Entity Category	Pattern
Companies	"‘Companies headquartered in *"’
Companies	"‘Companies established in *"’
Companies	"‘Companies based in *"’
Companies	"‘Companies listed on *"’
Companies	"‘* companies of *"’
Companies	"‘* companies"’
Organizations	"‘* organizations"’
Organizations	"‘Organizations based in *"’
Organizations	"‘Organizations established in *"’
People	"‘Living people"’

Table 5.2: Patterns used for category matching

5.6 Synonym Extraction

In the synonym extraction step we wanted to extract all the possible synonyms for all the named entities we had identified earlier. We collected all the links and redirects with destination and caption. Since we are not interested in the source article, we accumulated all links pointing to the same title, using the same caption. The synonyms listed in table 5.3 are an example of what we found through the synonym extraction. The synonyms listed here and their frequencies are real, but the selection of synonyms was done manually in this case.

Unfortunately the links do not provide us with a perfect set of synonyms as the link captions are very contextual dependent in some cases. What this means is that we found link captions pointing to named entities were the link was made up of a pronoun or other terms than proper nouns. In some cases the entity name used in the link caption is not even the same entity that

Main Name	Synonym	Frequency
George W. Bush	George W. Bush	7166
	Bush	453
	President Bush	392
	George Bush	129
	President George W. Bush	65
	G.W. Bush	62
	George W. Bush	32
United Nations	United Nations	9943
	UN	816
	U.N.	88

Table 5.3: Example of a synonym set

the link is pointing to, instead they are only related in some way. To deal with some of the noise we apply some simple filtering:

- Given the set S of potential synonyms for an entity, for each s_i
  - Remove any suffix enclosed in parentheses and apply a light stemming which strips it of any possessive form
  - Classify the synonym as good or bad synonym, remove s_i from S if it turns out to be bad
  - Given w_i as the frequency of s_i, remove s_i if $w_i < \sum w_i * \beta$

When trying to classify the synonym as a good or bad synonym we use a similar algorithm as the one described in 5.4, except we do not have an article text with occurrences we can use, therefore we ignore that rule. Since we then lose the rule which was used to handle single word names, we lower the limit of the minimum capitalized words required to one. We also use the frequency of a potential synonym to weight its importance and remove the ones that fall below a given threshold.

5.7 Search Application

We implemented a simple search application to make it easy to experiment with usage of the named entity dictionary. The search application accepts user queries which are then matched against the named entity dictionary. If only a single entity is found, we use this one to perform query expansion. On the other hand, if multiple entities are found to match the user queries, we present the different entities to the users and let them select the one they are actually looking for. We use the number of internal links to each entity to determine popularity so that we can display the most ones we deem more popular first.

When an unambiguous entity has been found it is time to do query expansion. Often the named entities had a very large selection of synonyms and we had to limit this or the expanded query would be too big. We select the five most popular synonyms based on the internal links pointing to the entity before we combined the synonyms using an OR clause to get the expanded query. Next the query is sent to the search engine to obtain the results that we want to show to the users. Figure 5.2 shows an example of a user query and the results obtained using the expanded query and the different synonyms.

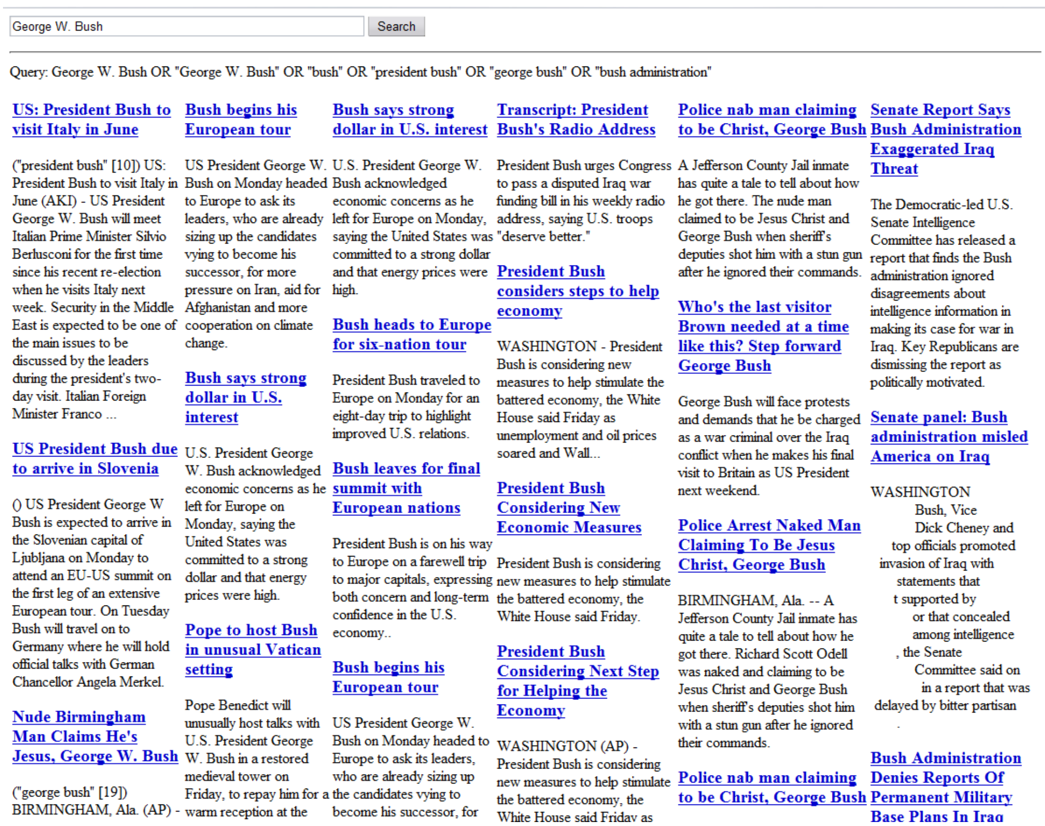


Figure 5.2: Search interface with results

5.7.1 Evaluating the Results

To aid in evaluating the results we built the search interface so that it would display the results of the expanded query alongside the results of queries made from each synonym. This means that we would have to issue up to six different queries to Yahoo News[35] which was the news search engine that we used. Therefore, to minimize load and improve the responsiveness of the user interface we cached the responses for each query string.

After all of the queries had been answered, either by cache or from the server we need to extract the result set. This was done using Hpricot[36] which makes it a very simple to do using XPath queries. After the results are parsed, they were displayed alongside each other, and for each hit in the expanded query result set, we list the individual synonym queries that included the same entry in the top 10.

Part IV

Evaluation and Results

Chapter 6

Evaluation

In this chapter we describe how we intend to evaluate the result and then we present them.

6.1 Evaluation Strategy

In this section we describe how we evaluated the results. We start by giving a description of precision and recall which we used extensively in the evaluation, followed by a description of the focus of our evaluation.

6.1.1 Precision and Recall

Precision and recall are common performance measures used in information retrieval [37][2][5]. They are used to measure the signal versus noise ratio of the result set according to a reference set. The reference set is the set of items that would be generated from the input set if the operation performed on the input set was perfect. Precision is therefore the fraction of relevant items in the result set, while recall is the fraction of relevant items that were included in the result set. If N denotes the size of the reference set, M the size of the generated result set and C the number of correct items in the result set, precision and recall are defined as:

$$P = \frac{C}{M}$$

$$R = \frac{C}{N}$$

In the case of information retrieval these measures are used to represent the percentage of the documents in the result set that are relevant to the user query, which is the precision, and the percentage of the relevant documents that were included, which is the recall. A good result set is a balance of precision and recall. To achieve high precision, recall tends to suffer, and precision tends to suffer if high recall is desired.

Query
Hillary Clinton
Barack Obama
John McCain
George W. Bush
United Nations

Table 6.1: Queries used in the evaluation

6.1.2 F-measure

An alternative to having two measures is the F-measure which combines precision and recall into a single performance measure [15][37]. The formula has the option of weighting precision and recall differently, but given identical weighting of precision and recall, the F-measure is defined as:

$$F = \frac{2PR}{P+R}$$

but with a parameterized weighting, α, it is defined as:

$$F = \frac{PR}{(1-\alpha)P + \alpha R}, 0 <= \alpha <= 1$$

6.1.3 Test Setup and Focus

The thesis has a divided focus. The first part is automatic generation of a named entity dictionary, and the second is using the dictionary to better handle the occurrences of different synonyms in news articles. In the first part of the evaluation, where the named entities extracted from Wikipedia are to be evaluated, we extracted smaller subset for in precision/recall calculations. These subsets were randomly chosen and then manually classified. See appendix A for the evaluation data.

To evaluate the query expansion we manually selected a set of sample queries, listed inin table 6.1. The reason for the selection was that they are all referring to entities with multiple synonyms that are appearing frequently in the news. A second reason is that the news search engine we will be using has the US as its target market. What we wanted to look at was how the result set of the expanded query was compared to the results sets from the individual queries.

6.2 Named Entity Recognition Results

In this section we will present the results of the evaluation of the named entities. First we present the results from the global named entity recognition, followed by the results from the three categories we extracted entities from, and last we use the category based entities to evaluate the algorithm used in the global extraction.

6.2.1 Global Recognition

In figure 6.1 the precision/recall for different values of α are shown. Here recall is the percentage of the entries that were recognized as entities, while the precision is the percentage of the entries correctly classified as named entities. The test data we used for this was a random subset of the Wikipedia entries which was manually classified as entity/non-entity and it can be found in appendix A.1.

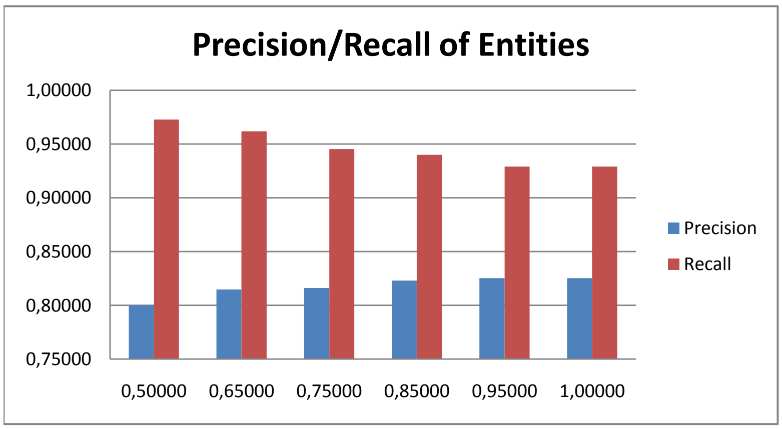


Figure 6.1: Precision and recall of the recognized entities

As the recall drops fairly evenly while the precision improves similarly for different values of α, it is difficult to see what the optimal value of α is. Figure 6.2 shows the F-measure for the different thresholds, with an equal weight given to precision and recall, and shows that in this case, $\alpha = 0.65$ is the one giving the best results.

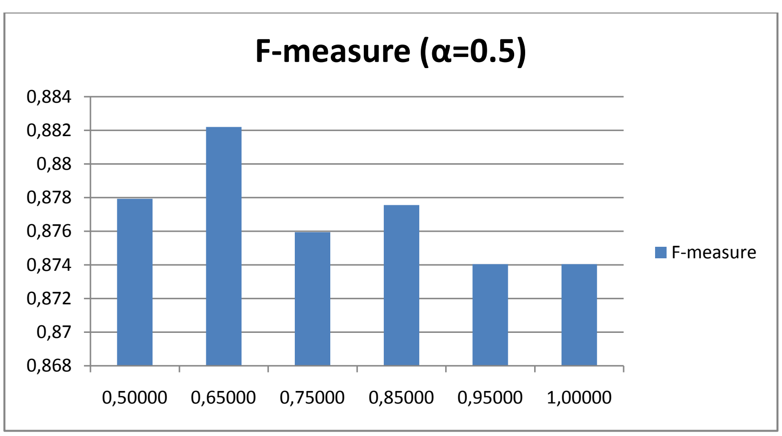


Figure 6.2: F-measure of the recognized entities

6.2.2 Named Entities from Categories

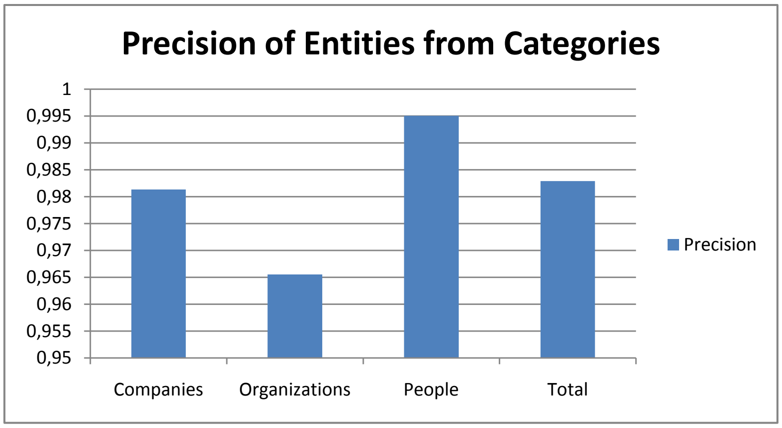


Figure 6.3: Precision of the categorized entities

Category	Pattern	Entities
Companies	"‘Companies headquartered in *"’	204
	"‘Companies established in *"’	7518
	"‘Companies based in *"’	8555
	"‘Companies listed on *"’	1365
	"‘* companies of *"’	15728
	"‘* companies"’	10955
Organizations	"‘* organizations"’	12661
	"‘Organizations based in *"’	1640
	"‘Organizations established in *"’	1

Table 6.2: Number of entities matching each of the patterns

The second approach we used to generate lists of entities was based around the use of string patterns to recognize the categories used for different kinds of entities. Table 6.2 shows a breakdown of how entries matching the different patterns were divided. As one entry can be a member of multiple category, the total number of entities per category is less than the sum of the entries matched by each pattern, and the number of unique entities per category can be seen in table 6.3.

In appendix A.2 we have listed a random subset of 585 entities that match any of the patterns. We calculated the precision by manually classifying this subset. From this we found a very small list of entries that were not named entities. These are shown in table 6.4. As can be deduced from the names, most of these are in reality entries that list multiple entities or general terms, except for *Albert and David Maysles* which we consider a misclassification still since it is an entry about two different entities that are related, but not a single entity. In figure 6.3 the precision of the different categories is shown.

Category	Unique Entities
Companies	27188
Organizations	11988
People	228071

Table 6.3: Number of unique entities per category

Category	Non-Entity
Companies	China-based financial stocks in Hong Kong
	Dynamic packaging
	List of assets owned by Time Warner
	List of national and international moving associations
	Norwegian types of company
Organizations	Charity badge
	Death squad
	List of Aikido organizations
	List of fictional companies
People	Albert and David Maysles

Table 6.4: Non-entities tagged with entity categories

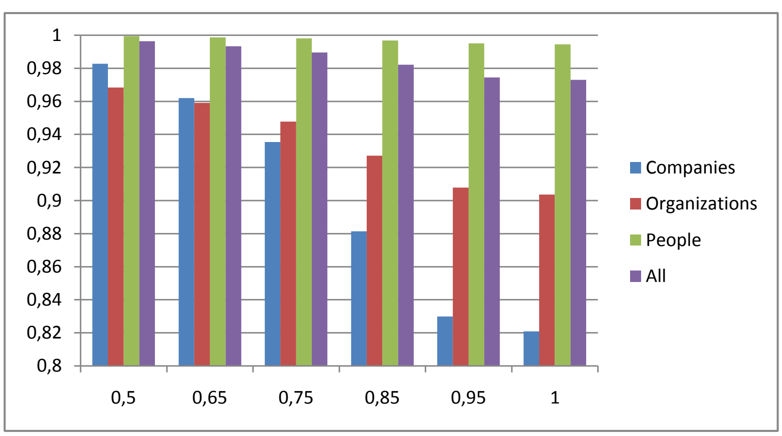


Figure 6.4: Precision of the named entity classification algorithm when used on the categories

6.2.3 Classification of Category Based Entities

In figure 6.4 the recall of the general entity classification algorithm is evaluated using the three categories of entities extracted using the category patterns as test data. We see that the recall of *companies* and *organizations* vary depending on the α threshold. This is because small uncapitalized words are more common in these entities. Overall the average recall is high since the *people* category is considerably larger than the other two.

6.3 Synonyms

Category	Number of Entities	Average number of synonyms	Max number of synonyms
Companies	25284	3.2	103
Organizations	11122	2.7	69
People	221207	1.9	153
All	257613	2.1	153

Table 6.5: Statistics from the synonym extraction

The synonym extraction was based around the categorized entities and the average number of synonyms found per category is shown in table 6.5. As we can see the number of synonyms found was in average lower among people than the other categories. We believe this is because of the large amount of people entries in Wikipedia that are very short on content as they are less popular entries, and are therefore having very few links pointing to them. Also, for companies and organizations, the use of abbreviations are more common, resulting in more synonyms on average.

We classified a random subset, listed in appendix A.4, of the potential synonyms and used this to calculate precision/recall of the link labels and redirects classified as synonyms. As shown in figure 6.5, the precision/recall of people was considerably higher than for companies and organizations. Especially for organizations, the subset used for the evaluations contained very few organizations, which may have affected the precision/recall calculation of this category.

6.4 Query Expansion

In table 6.6 we see the queries used together with their expanded versions when limiting the expanded query to the top 5 synonyms. The synonyms selected were the ones with the most inbound links using the synonyms as link captions. Figure 6.6 shows the number of original top 10 hits that made it into the result set of the rewritten query.

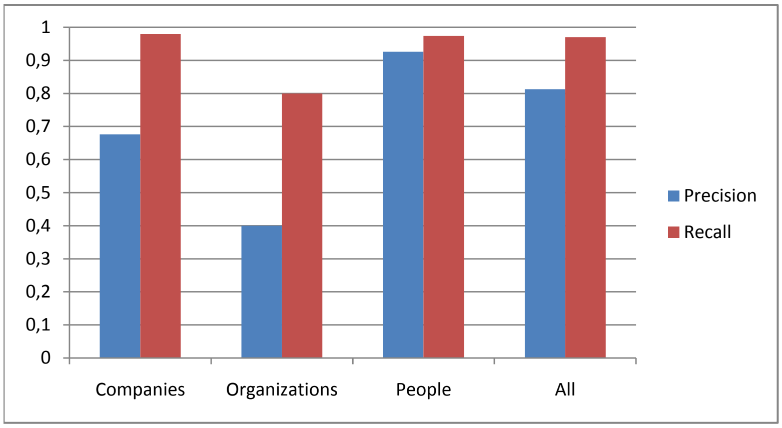


Figure 6.5: Precision and recall for the synonyms

Query	Expanded Query
Hillary Clinton	Hillary Clinton OR "Hillary Rodham Clinton" OR "hillary clinton" OR "hillary" OR "hillary rodham" OR "clinton"
Barack Obama	barack obama OR "Barack Obama" OR "obama" OR "barack obama" OR "senator barack obama" OR "barak obama"
John McCain	John McCain OR "John McCain" OR "mccain" OR "john s. mccain iii" OR "senator john mccain" OR "john mccain"
George W. Bush	George W. Bush OR "George W. Bush" OR "bush" OR "president bush" OR "george bush" OR "bush administration"
United Nations	United Nations OR "United Nations" OR "un" OR "uno" OR "u.n." OR "the un"

Table 6.6: Expanded queries

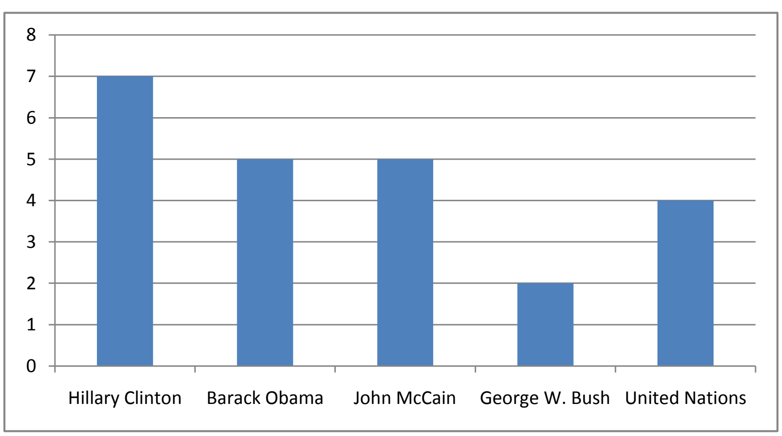


Figure 6.6: Number of hits in the expanded query's result set that appeared in the top 10 of any of the synonyms

Chapter 7

Discussion

In this chapter we discuss the design choices, the results and possible changes that could be done to improve the results.

7.1 Named Entity Recognition

We considered two different approaches for extracting named entities from Wikipedia entries. Both methods have advantages and disadvantages. The first one is a generic method in the sense that it is able to recognize entities from all of Wikipedia. It is based around the fact that proper nouns are capitalized, and named entities are proper nouns. There is one problem, and that is that all Wikipedia entries have the first character in their title capitalized by convention, which means it is not useful to look at the first character to recognize proper nouns. If it was not for that, it would have been considerably easier to recognize named entities with a high precision. Instead we had to rely on a set of heuristics. As seen in figure 6.1 and 6.2 we are able to obtain a precision of 80% and higher with a recall around 95% using these heuristics.

As the goal of this thesis was to find entities that were relevant in a news context, we tried a second approach which yielded a considerably improved precision over the first method, in addition to giving us the entities grouped by categories. The categories selected were categories that are highly related to news, and the smaller list of entities generated through this method may actually be an advantage. A problem with generating too many entities is that only a fraction of them are actually news relevant and the irrelevant ones may become noise as they match the wrong person. That is why we selected only a few news related categories.

From what we have seen, it would be fairly easy to use this method to generate a collection of geographical entities, including which entities that are part of another entity simply by looking at the entry's categories and title. In the case of geographical entities, the entry titles often follow a pattern where the things like county, state, or country follow the entity name separated by comma.

There is a third approach that we thought would have been interesting to explore, but we did not have time to pursue this one further. Wikipedia has a template system where articles can include a template while passing along a set of variables that are used by the template. This is commonly used to provide info boxes that display various bits of information in a structured way. The template usage could therefore be useful both in named entity recognition and categorization since for example info box templates are used for specific categories. Another advantage in our opinion is that info boxes are very common for well known entities, which would be of great help if we were to limit the extracted entities to popular ones.

7.2 Synonyms

Finding synonyms was an important part in the creation of the entity dictionary, and using the entities found earlier we considered all links and redirects to any of them as potential synonyms. What was saw was that the popular entities usually had a very large list of potential entities, often made up of various spelling variations and different uses of abbreviations or titles. One reason for the very large amount of synonyms with very tiny differences is that while the pages for popular entities are of high quality, the same may not hold true for the entries linking to them which results in a lower quality of the link captions coming from these entries. A possible approach to this would be to try to determine the quality of the entries the links are coming from and use that to weight the synonyms.

The results in figure 6.5 indicate a very high precision for the synonyms found for people, but for companies and organizations this is considerably lower. One reason is that companies in some cases have subsidiaries which did not have separate pages, but instead they were only given a short description on the parent company's page. We did not consider this to be the same entity, and therefore filtering the of company synonyms is more difficult than people synonyms. Another possible explanation is that the people category was very large compared to the other categories, including many short stub articles, and because of this they had fewer average synonyms.

The average number of synonyms listed in table 6.5 would have been considerably higher if we had only looked at popular entities. This is to be expected as Wikipedia has more than 200000 people entities, where the majority are not commonly known. These lesser known entities are likely to have very few synonyms.

7.3 Query Expansion

The results of the query expansion were more difficult to evaluate than the entities themselves. What we observed was that the precision was very high with queries made up of news relevant entities, ie. the top 10 results would all be relevant. Our basic idea for how to interpret the results were then that a good result set would be made up of a combination of the top hits for each of the individual synonyms of the entity used as a query, but the results of the expanded queries were very far from that. Instead they were made up of a few of the top 10 matches appearing among the synonyms and the rest would be articles that did not make it to the top 10 for any of the synonyms.

After having a closer look at some of these articles we think it is very obvious why this happened when having in mind how common retrieval models like the vector space model work. The articles that would be ranked in the top 10 only when the expanded queries were used were articles were multiple synonyms would appear in the same text. The articles that did well when each of the synonyms were used individually were articles which were more consistent in their use of synonyms. Another thing that became visible was that some articles would include both the full entity name and an abbreviation next to it like "United Nations (UN)".

Another thing that became very visible when we tried to compare the results of the queries using different synonyms was the number of duplicate articles. With duplicate articles we mean articles that were not only reporting on the same topic, but they seemed to be the same article but the URL was different and at times they had some very minor differences in editing. It seemed like the news search engine we used would arbitrarily remove all but one of the duplicates from the result set during query time. What this meant was that the removed duplicates would change depending on the query used, making it more difficult to do automatic comparison of result sets to see if the same article appeared in both.

A problem with the query expansion was that the popular entities had a very large amount of synonyms with very small variations. The entity with the most synonyms had as many as 153 different synonyms as shown in table 6.5. If we were to expand the queries with all the synonyms of the entities specified, we would get queries so large they would most likely result in a serious performance hit. This would be unacceptable in a real world usage, and instead of trying to select the synonyms to include, normalizing the indexed documents would avoid the problem of having to select only a subset of the synonyms. Entity normalization would only have to be done once per document, and since the documents are limited to news articles the total number of documents is very small compared to the document collected indexed by web search engines. In other words, matching against the entire synonym list during indexing would be considerably cheaper. The second benefit is that during searching, the query would not have to be expanded to multiple terms meaning that the queries would not be more expensive than before performance wise.

Originally we wanted to normalize the entities before indexing, but in the end we did not have access to the raw news documents. Another outcome of normalization would be that rarer synonyms would not be ranked higher to to their infrequent occurrences.

Another possible solution to focus the synonyms would be to combine it with the use of query logs. The idea is to use the query logs to identify the synonyms that are actually in use by the users from the large set of possible synonyms. This could result in a much smaller, but highly relevant collection of named entities and synonyms

Chapter 8

Conclusion

The market of specialized search engines, or verticals is growing. Today we have search engines that target a specific domain like news, blogs, videos, images, academic research, etc. These specialized search engines are in contrast to regular web search engines limiting their scope of what they are trying to index which opens up for the use of domain specific knowledge in order to improve the search quality.

In this thesis we have been focusing on using Wikipedia to automatically build a dictionary of named entities and their synonyms. The intend usage of this dictionary is named entity recognition and disambiguation of news search queries with the purpose of helping the users find articles about the entity independent of which entity name is used in the article.

The evaluation shows that Wikipedia is well suited as a data source for named entity mining. We were able to extract a large amount of entities with a high precision, and the synonyms found were mostly relevant, but in some cases, the number of synonyms were very high. This resulted in lots of synonyms that were correct, but would rarely be used in a search query as they were very context specific.

The usage of the entity dictionary did not yield the results we were hoping for. The expansion of the search queries made hits that would normally rank outside the top 10 appear on the top ranks, while the usual top ranked hits would disappear from the ranking. Part of this, we believe, is due to having very little control over the ranking in the news search engine we used as the back-end of our modified system. Despite this, we believe the entity dictionary could prove valuable together with normalization.

Bibliography

[1] y. E. Christian Bøhn, "Entity extraction from structured yahoo news query logs," 2006.

[2] B. R.-N. Ricardo Baeza-Yates, *Modern Information Retrieval*. 1999.

[3] G. Grefenstette and P. Tapanainen, "What is a word, what is a sentence? problems of tokenization," 1994.

[4] W. Frakes and R. Baeza-Yates, *Information Retrieval: Data Structures and Algorithms*. 1992.

[5] T. C. B. Ian H. Witten, Alistair Moffat, *Manging Gigabytes - Compressing and Indexing Documents and Images*. 1999.

[6] X. Su and J. A. Gulla, "Semantic enrichment for ontology mapping," in *Proceedings of the 9th International Conference on Applications of Natural Language to Information Systems*, (Manchester, UK), 2003.

[7] G. Solskinnsbakk, "Ontology-driven query reformulation in semantic search," 2007.

[8] J. Han and M. Kamber, *Data Mining - Concepts and Techniques*. 2006.

[9] M. M. Torsten Zesch, Iryna Gurevych, "Analyzing and accessing wikipedia as a lexical semantic resource," 2007.

[10] "Wikipedia english main page." http://en.wikipedia.org/wiki/Main_Page.

[11] M. P. Razvan Bunescu, "Using encyclopedic knowledge for named entity disambiguation,"

[12] G. K. Ralf Schenkel, Fabian Suchanek, "Yawn: A semantically annotated wikipedia xml corpus,"

[13] R. Grishman, "The nyu system for muc-6 or where's the syntax," 1995.

[14] R. Yangarber and R. Grishman, "Nyu: Description of the proteus/pet system as used for muc-7 st," 1998.

[15] A. Borthwick, "A maximum entropy approach to named entity recognition," 1999.

[16] D. M. Magerman, *Natural language parsing as statistical pattern recognition*. PhD thesis, Stanford, CA, USA, 1994.

[17] D. Bikel, S. Miller, R. Schwartz, and R. Weischedel, “Nymble: a high-performance learning name-finder,” 1997.

[18] S. Miller, M. Crystal, H. Fox, L. Ramshaw, R. Schwartz, R. Stone, R. Weischedel, and the Annotation Group, “Algorithms that learn to extract information–BBN: Description of the SIFT system as used for MUC,” in *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, 1998.

[19] E. Jaynes, “Information theory and statistical mechanics,” *Physics Reviews*, vol. 106, pp. 620–630, 1957.

[20] E. T. Jaynes, *Probability Theory : The Logic of Science*. Cambridge University Press, April 2003.

[21] “Wordnet - a lexical database for the english language.” http://wordnet.princeton.edu/.

[22] “A wordnet-based approach to named entities recognition,” 2002.

[23] R. M. n. Antonio Toral, “A proposal to automatically build and maintain gazetteers for named entity recognition by using wikipedia,” 2006.

[24] B. Velez, R. Weiss, M. Sheldon, and D. Gifford, “Fast and effective query refinement,” in *Proceedings of the 20th annual international ACM SIGIR conference on research and development in information retrieval*, pp. 6–15, ACM Press New York, NY, USA, 1997.

[25] D. Buscaldi, P. Rosso, and E. S. Arnal, “Using the wordnet ontology in the geoclef geographical information retrieval task.,” in *CLEF* (C. Peters, F. C. Gey, J. Gonzalo, H. Müller, G. J. F. Jones, M. Kluck, B. Magnini, and M. de Rijke, eds.), vol. 4022 of *Lecture Notes in Computer Science*, pp. 939–946, Springer, 2005.

[26] D. Buscaldi, P. Rosso, and E. Sanchis, “A wordnet-based indexing technique for geographical information retrieval.,” in *CLEF* (C. Peters, P. Clough, F. C. Gey, J. Karlgren, B. Magnini, D. W. Oard, M. de Rijke, and M. Stempfhuber, eds.), vol. 4730 of *Lecture Notes in Computer Science*, pp. 954–957, Springer, 2006.

[27] P. M. Kruse, A. Naujoks, D. Rösner, and M. Kunze, “Clever search: A wordnet based wrapper for internet search engines,” *CoRR*, vol. abs/cs/0501086, 2005. informal publication.

[28] A. M. Cohen, “Unsupervised gene/protein named entity normalization using automatically extracted dictionaries,” 2005.

[29] A. M. C. W. H. C. D. K. Spackman, “Using co-occurrence network structure to extract synonymous gene and protein names from medline abstracts.,” 2005.

[30] “Open directory project.”

[31] “Wikipedia description of the use of diambiguation pages.” http://en.wikipedia.org/wiki/Wikipedia:Disambiguation.

[32] “Description of the categorization system used in wikipedia.” http://en.wikipedia.org/wiki/Wikipedia:Categorization.

[33] "List of english stop words." http://www.idi.ntnu.no/emner/tdt4215/resources/englishST.txt.

[34] "Wikipedia category used for living people." http://en.wikipedia.org/wiki/Category:Living_people.

[35] "Yahoo news." http://news.yahoo.com/.

[36] "Hpricot - a fast, enjoyable html parser for ruby." http://code.whytheluckystiff.net/hpricot/.

[37] J. Makhoul, F. Kubala, R. Schwartz, and R. Weischedel, "Performance measures for information extraction," 1999.

Part V

Appendixes

Appendix A

Evaluation Data

A.1 Classification of Entity/Non-Entity Subset

A.1.1 List of Entities

Electoral district of South-West Coast
Milo Keynes
Ralph Taeger
International Research on Working Children
Luca Cigarini
"-And He Built a Crooked House-"
LÉ Róisín (P51)
Britton Johnsen
The Lost Battalion
Royal College Port-Louis (Mauritius)
Rhys Evans
Robert Latta (White House intruder)
Juan Downey
First Restoration
Argentina national rugby union team
Lindsey Wallace
Meezen
Stored Waste Examination Pilot Plant
William Byrd
Attila József
Comte Desbassayns de Richemont
Niemand hört dich
Tess Bateman
Earshot (Buffy episode)
Simon de Vlieger
Failsworth West
Harry van der Meer
Duane Bobick
Eddy Ko
Hunter Johnson (disambiguation)
Hartvig Svendsen
Emma Roberts
Victoria (New Brunswick electoral district)
Brighton Robins
Dance Got Sick!
Bhawal
Night Skies (film)
Amatole District Municipality
Cadishead
The Killers (short story)
Manuel Gutiérrez Nájera
Revés/Yo Soy
Prince Umberto of Bulgaria
Ray Cunningham
Mlynica
Lee Seung-Ho
Elbit Systems
Makyla Smith
War of Genesis
Drumoak

Rupert Holmes
Via dei Fori Imperiali
Arcadia Publishing
Rubens Farias Jr.
Staten Island University Hospital North Campus
Gladius DB
Joseph Alfred Lamy
Zazu
Jacopo Bertucci
West Seneca East Senior High School
Junius Hillyer
Ray Lawson
Pueblo del Arroyo
Philip A. Kent
Carlisle Upperby TMD
Venus (Frankie Avalon song)
Angela Summers
Edward II
Jackie Wright
Castle Ashby
Three Towns
Di Air
Murder on the Nile/Hidden Horizon
My Antonia (film)
Fire in the Abyss
Florida Atlantic Owls baseball
Pierre Ducasse (footballer)
Maria Luisa of Orléans
Cleethorpes Pier
Audubon Avenue (Manhattan)
Grand Pass (Washington)
Isabelle Breitman
Buckman Tavern
Playwutchyalike: The Best of Digital Underground
Ken Caillat
Samir El Moussaoui
Chobe National Park
Yusuf Hamied
Timothy Chambers
Onyx 2 On The Bay
James White (General)
Umanità Nova
Fish Leong
W. Allen Wallis
Stewart Reburn
John Wilton
Bernard Herrmann
Barbro Martinsson
Esko Rekomaa
From This Moment On (Cole Porter song)
Trouble at the Henhouse
NetJets
VFinity
Florence Marina State Park
Sümeyra Kaya
JMax
Sheriff (band)
Frank Mossfield
Hyderabad District (Pakistan)
Pure Frosting
Imperial College Boat Club
Albedo (Xenosaga)
United States Secretary of Transportation
Andrew Horning
Leszek Dunecki
Lao Bao town
Andrew McGarry
Ballymeanoch
Angkor International Airport
County Route 506 (New Jersey)
Ben Moon
Charles Fickert
A. H. J. Prins
Celestial Season
Martin Evans
Corippo
Alvega
Jacqui Abbott
Emil Molt
Konstantin Mirchev
Josée Chouinard
Buckeye Municipal Airport
Invisible Ones
Joey Eischen
Unity (Georgia)
2003 U.S. Open - Men's Singles
Clarence Hammar
Erythnul

Grouse Mountain
You Must Believe in Spring
Tube Mice
Designline
Gabaldón
Stanley K Hornbeck
Miguel Maria N'Zau Puna
Vauxhall and I
The Young Master
Broadholme
Terragnolo
John Newman (Australian politician)
Johnny Douglas (conductor)
Berry Oakley
Francis Pemberton
3rd Shanghai International Film Festival
Harold Acton
Voices from the Sky
Pictures of Home
Dirk van Hogendorp (1761-1822)
Gus and Jaq
Christopher John Farley
Dave Hudson
2008 NCAA Men's Division I Basketball Tournament
Carol Giambalvo
Lodi High School (California)
National Technical University of Athens
Randy Bowen
Gamera 2: Attack of Legion
Ecuador
Michael Jeffery (manager)
Workers Party of the Netherlands (build-up organisation)
Michael Gallagher (translator)
UEFA Champions League 2005-06
Colleen Farrington
Robert Neal Adams
François Jacques Boeri
Colonel By Secondary School
Marmaris
Trianon (Frankfurt am Main)
Dale Atkeson
Festive Overture (Shostakovich)
Milicent Shinn
Indira Jaising

A.1.2 List of Non-Entities

Acuticostites
Mitochondrial trifunctional protein
Commemorative coins of Denmark
NH RSA Title LXIII
Chamanto
OX postcode area
Security Force Auxiliaries
Formula Renault
Parliamentary representation from Buckinghamshire
Streptococcus mitis
Afflicted (band)
Armenians in Kuwait
Mepolizumab
Sports Illustrated Cover Jinx
Earl of Moray
Alanine
European Ratsnake
1006 in poetry
List of Canadian airports by location indicator: CT
Moonlander
Proper name
Neohouzeaua
Schimmel
Embryonic disk
List of host cities of the Eurovision Song Contest
Anthropoides
Nesquik
Sword-leaved Helleborine
Conformal field theory
Anta
Cocek

Administrative divisions of Chukotka Autonomous Okrug
Hedeoma pulegioides
W Ursae Majoris variable
Undulator
Cat thyme
Jenmi
Panicfire
Railroad nicknames
In My Own Time
Sensu
Fire control
Chindro
Parting tradition
Artistic License
Shadow knitting
Dirichlet algebra
Snap (dance move)
Shoshannim
County cricket
NASA Exceptional Service Medal
St. Johnstone F.C. seasons
System image
Hummock
Niederwil
List of high schools in Massachusetts
Botaniska Notiser
Reading copy
Sarcosinemia
Tramontana (sports car)
Editing Agency of Korean History
Musa (name)
Independence class aircraft carrier
Biological membrane
California Manroot
Olive (color)
300 m Standard Rifle
Restricted product

A.2 Entities Recognized based on Categories

Following is a sample of the named entities found, grouped in their categories.

A.2.1 Companies

84 Lumber
AC Moore
ARAMARK
Aberdeen and Asheboro Railroad
Acme Whistles
Advanced Cell Technology
After Dark Films
Aji Ichiban
Alcon
Allens Boots
Altera
American Christian Press
American Zoetrope
Anderson Valley Brewing Company
ApS
ArcheDream
Arno Political Consultants
Ashanti Goldfields Corporation
AstraZeneca
Au
Auto AG Rothenberg
Axcom Trading Advisors
BEAM.TV
Bandwidth.com
Barclays Global Investors
Bay Networks
Belcan
Berkeley Systems
Bif Bang Pow!
Bird & Bird
Blausen Medical Communications
Bluescope Lysaght

BookFinder.com
Bowater Forest Products
BridgePort Brewing Company
British Touring Shakespeare Company
Brush Turbogenerators
Bunnpris
C venues
CJM Racing
Cabot Corporation
Calyon
Canadian Pacific hotels
Cardkey
Cary Safe Company
Celestial Digital Entertainment
Century 21
Charles Schwab Corp.
Cheshire Bus and Coach
China-based financial stocks in Hong Kong
Cincinnati Opera
Civic Hall Performing Arts Center
Cluj-Napoca Companies
Cole Haan
Commercial Aircraft Sales and Leasing
Computas AS
Container Corporation of America
Crain Communications Inc.
Cromwell Radio Group
Curves International
DC10
Dai Pai Dong
Dari Mart
De Brauw Blackstone Westbroek N.V.
Delta Faucet Company
Deutsche Bank
Digital Entertainment Network
Divine Chocolate
Dorado Wings
Drum Workshop
Dynamic packaging
EG Wrigley and Company
EarthLink
Eden Studios, Inc.
Eko guitars
Elizabeth Hurley Beach
Encore Computer
Entra Eiendom
Estar
EverBank
EyeCatcher Entertainment
Fairchild Group
Farrel Corporation
Ferrocarril General Roca
Fineos
First Second Books
Florida East Coast Railway
Forex AB
Fram
FremantleMedia
FujiGen
GAINSCO
Galaxy Communications
Gate Gourmet
Genesco Inc.
Ghana Airways
Glenmorangie
Golden Lamb Inn
Graham, Anderson, Probst & White
Great Western and Great Central Joint Railway
Group Sense PDA
Gupta Technologies
HT Motorsports
Hampshire Mall
Harman Kardon
Hay Group
Helsinki City Transport
Hideous
Hits & Favorites
Honda Atlas Cars Pakistan
Hovertravel
Hussain Industries
IBP, Inc.
ITV Digital Channels Ltd
ImageMovers Digital
Indian Railways
Inmarsat
Interceptor Micros
Interval International
Ironclad Games
J-Air

JW Marriott Hotels
Jaycar
Joffrey Ballet
Journal of Irreproducible Results
KLM Telephone
Kansas City, Pittsburgh & Gulf Railroad
Kemira
Kim Son
Klei Entertainment Inc.
Korea General Magnesia Clinker Industry Group
Kuwait Petroleum International
LXD Incorporated
Land Systems OMC
Le Coq Sportif
LendingTree
Life is Good
Lionhead Studios
List of assets owned by Time Warner
List of national and international moving associations
Lledo
Lonely Planet
Lowrance Electronics
MAN Roland
MTVX
Magna International
Manchester, South Junction and Altrincham Railway
Marcus Clark & Co.
Martin Band Instrument Company
Maurice Girodias
McKinsey & Company
Meier & Frank
Merix Corporation
Midnight Insanity
Mingxing Film Company
Mitsuwa Marketplace
Monkeystone Games
Morris & Company
Moxi
Mutual insurance
NI 43-101
Nanosight
National Orchestra Service
Nekeme Prod
Nevada Power Company
Newcastle Publishing Company
Nigerian National Petroleum Corporation
Nokia
North Eastern Railway
Norwegian types of company
Nuyorican Productions
Ocean Software
Old America Stores
Ontario Knife Company
Optus Television
Orpak
Overseas Shipholding Group
PHONE+ magazine
Pacific Publishing Company
Panic
Parry Sound Colonization Railway
Pechiney
Perceptis
Petrol Ofisi
Pic 'N' Save
Pizza Haven
Point of View, Inc.
Ports of Auckland
Presbyterian Publishing Corporation
Pro Arts Inc.
Provincial Airlines
Q-Telecom
Quicksilver Software
RPath
Raisio Group
ReactiveMicro.com
Redmonol Chemical Products Company
Renaissance Books
RheoTec Messtechnik GmbH
Riverside Methodist Hospital
Rogers Telecom
Rover
Ruskin Pottery
SBM Offshore
SNET America
SafeTV
Samsung Techwin
SaskEnergy

Schoolhouse Press
Seagull Camera
SemGroup
Seven Stories Press
Shemaroo Entertainment
Sick Room Records, LTD
Simmons Bedding Company
Skelly Oil
Smith International
Softdisk
Sonokong
Southeastern Power Administration
Spark Unlimited
SportsBooks Limited
Standard Electric Time Company
Statprobe
Stolt-Nielsen
Studio Fantasia
SunTrust Banks
Surrey Iron Railway
Symyx Technologies
TARTA
TUI Travel PLC
Tallinna Autobussikoondis
Taxijet
TeleComputing
Tembec
Texize
The Customart Press
The MathWorks
The Tabletop Group
Thomson Holidays
Time Warner
Tomioka silk mill
Towle Silversmiths
Transnational Corporation of Nigeria
Triple Canopy, Inc.
Tundra Publishing
U.S. Robotics
Ultra Electronics
United Development Company
Unsanity
Vajra Enterprises
Venray sheep companies
Victoria Express
ViroPharma
Volatile Games
WSP Group
Warner Aircraft Corporation
Weather Underground
West Coast Railway
Westnet
Wild Whirled Music
WingTips Airport Services
Woolworths
Worshipful Company of Glovers
XITEX Software
Yardbirds Home Center
Yves Saint-Laurent
Ziv Television Programs

A.2.2 Organizations

ANZUS
Action Palestine
Aid to Artisans
AllBusiness.com
American Association of Orthodontists
American Friends Service Committee
American Social Science Association
Animal Defenders International
Armenian Revolutionary Army
Association of Business Executives
Astrophysical Institute Potsdam
Automobile Journalists Association of Canada
Baptist Student Union
Bhaktivedanta Manor
Blue Cross and Blue Shield Association
British Association for Cemeteries in South Asia
Building society
CSTC Trenton

Canadian Association of Promotional Marketing Agencies
Canine Companions for Independence
Center for Media and Public Affairs
Charity badge
Children's Film Foundation
Churches of God General Conference
Coalition for the Good of All
Committee on Institutional Cooperation
Competitiveness Policy Council
Constantian Society
Council of Major Superiors of Women Religious
DAIA
Death squad
Diamond Sangha
EC-SAR
Education Conservancy
Engineers for a Sustainable World
European Association of Conservatoires
European and Mediterranean Plant Protection Organization
Famine Early Warning Systems Network
Film Unit
Forsaken
French Defence Health service
Gawad Kalinga
Girl Guides Association of Papua New Guinea
Got Questions
Guidelines International Network
Harvard-Radcliffe Science Fiction Association
Hindu Makkal Katchi
Howard Brown Health Center
IPIC
Independent Task Force on North America
Institute in Basic Life Principles
International Accounting Standards Committee
International Colour Authority
International Football Association Board
International Progress Organization
International Yoga Federation
Islamic Mission of Belize
Japan Baptist Association
John Aspinall Foundation
Kashi Mutt
Kobayashi aikido
Lake View Citizens' Council
Legion of Doom
List of Aikido organizations
List of fictional companies
London Club
Magician Alliance of Eastern States
MassEquality
Merit School of Music
Minnesota Zen Center
Muddy York Rugby Football Club
NCPAD
National Association of Military Marching Bands
National Council of Resistance of Iran
National Lesbian and Gay Journalists Association
National Union of South African Students
New England Research Institutes
Nippon Foundation
Norwegian Maritime Directorate
Odinic Rite
OpenTravel Alliance
Orpheum Foundation for the Advancement of Young Soloists
Pakistan Boy Scouts Association
Peace Society
Philalethes Society
Political Research Associates
Program for Appropriate Technology in Health
Quackwatch
Republican Conference Chairman of the United States Senate
Rodobrana
Royal Order of Scotland
SPQ Libre
Self-Realization Fellowship
Sigma Theta Epsilon
Society for Electro-Acoustic Music in the United States
Soroptimist

Sporting Arms and Ammunition Manufacturers' Institute
Student Environmental Action Coalition
Swedish Film Institute
Taxpayer groups
The Banyan
The Girl Guides Association of Antigua and Barbuda
The Order
The Waffle
Transportation Alternatives
UFORM
Union of International Associations
United States National Karate Association
Vaccine and Infectious Disease Organization
Vision America
Wayne RESA
Wireless Toronto
World Buddhist Forum
World Taiwanese Congress
Young Men's Institute

A.2.3 People

"Hungry" Charles Hardy
Abdur Razzak
Ahmet Zappa
Alan Brinkley
Albert and David Maysles
Alex Grammas
Alexi Giannoulias
Aliza Olmert
Amber MacArthur
Andrew Howe
Anton Villatoro
Arild Andersen
Arturo Torres
Avery Cardoza
Barbara Mertz
Becky Morgan
Beverlei Brown
Bill Schwab
Blu Greenberg
Bob Wolff
Brad Childress
Brent Patterson
Brian Price
Bruce Reid
Carl Hewitt
Carmine Boal
Cathy Hughes
Charles E. Barkley
Chase Daniels
Chris Burke
Chris Smith
Christophe Bordeau
Cindy O'Callaghan
Clifford Ray
Conrad Brooks
Craig Sager
D. Ray Perdue Jr.
Dan Gillespie Sells
Daniel Kaluuya
Danny Strong
Dashon Goldson
David Atherton
David Giffin
David Meyer
David Ushery
Deborah Gordon
Dennis K. Villa
Dimitar Stilianov
Don Carter
Donovan Patton
Drew Coleman
Eberhard Weise
Edmund Purdom
Eitan Cabel
Ella Tripp
Emmanuel Lubezki
Eric Rupe
Erwin Schild
Ewan McCray
Felipe Baloy
Floris Jansen
Frank Broome
Freaky Flow
Fuzzy Zoeller
Gary Anderson
Geert Versnick
George Gao
Gerald Sibon
Gil da Cruz Trindade
Glenn Kaiser
Graham Day
Gregory C. Farrington
Guy Whittall
Hank Aaron
Harry Fowler
Heinrich Mussinghoff
Herb Grubel
Holly Davidson
Hugues Claude Pissarro
Ian Sample
Isolde Kostner
J. Stuart Perkins
Jacob Smith
James Blaylock
James O'Connor
Jared Boice
Javid Hussain
Jean-Jacques Burnel
Jeff Sagarin
Jeon Kwang-cheol

Jim Doyle
Jimmy Dixon
Jodi Santamaria
Joel Dreessen
John Branney
John Gardiner
John Sabini
Johnny Kerr
Jonathan Kerrigan
Julianne Baird
Justin Wheatley
Kang Soo Jin
Katalin Szili
Kaylynn
Kelly Overton
Kenneth Schellenberger
Kevin L. Bryant
Kim Jagtiani
Ko Jong-Soo
Kunio Kitamura
Lance Davids
Laura Freixas
Lee Blackburn
Leo Hayden
Lew Krausse Jr.
Lindsay Frost
Logan Vander Velden
Lowitja O'Donoghue
MC Romeo
Malcolm Boyden
Marc Gicquel
Marcus Stephen
Marie Plourde
Mark Blundell
Mark Ormrod
Marshall Faulk
Marty Feldman
Masashi Nakayama
Matt Stewart
Mauricio de Sousa
Mel Machin
Michael Blaudzun
Michael Johnson
Michael Stegmayer
MichÃ¨le Jacot
Mike Deodato
Mike Stahr
Mohammad Reza Mamani
Moshe Ohayon
Nacanieli Seru
Natalio Lorenzo Poquet
Neil Nunes
Nick Johnson
Niilo Halonen
Padraig Parkinson
Pat Sobeski
Pattie Boyd
Paul Kehoe
Paul de Casteljau
Per Wikstrï¿½m
Peter G. Tsouras
Peter Staples
Philip Carlo
Piet Keizer
Prosper Avril
Rafael Palmeiro
Randall Godfrey
Ray Williams
Renaldas Seibutis
Richard A. Pittman
Richard O. Spertzel
Ricky Steamboat
Robert AhMat
Roel Luynenburg
Ron Allen
Rory McCarthy
Ryan Gosling
Sajib Miah
Sammy Lee
Sarah Huck
Scott Maslen
Seiji Osaka
Shahid Israr
Shawn Stasiak
Shona Moller
Simon Mrashani
Sonja Bennett
Stephen Lodge
Steve Kariya
Steven Rathman
Sulley Muntari
Takako Katou
Tatiana Poutchek
Terry Bickers
Thom Fitzgerald
Tiffany Brissette
Timothy R. Ferguson
Tom Dine
Tony Kendall
Travis Diener
Ty Esler
Valentin Simion
Vic Bubas
Vincent Ribeton
Warren Munson
Wilfried Nelissen
William Prochnau
Wu Shih-Hsih
Yoann Lachor
Yuval Yairi
Zintis Ekmanis

A.3 All Link Captions used for Elizabeth II of the United Kingdom

Following are all the different link captions used to link to the article *Elizabeth II of the United Kingdom* and the frequency of each link caption.

Synonym	Frequency
Queen Elizabeth II	1817
Elizabeth II of the United Kingdom	291
Queen	257
The Queen	163
Queen Elizabeth	131
the Queen	113
HM The Queen	107
HM Queen Elizabeth II	43
Her Majesty Queen Elizabeth II	41
Her Majesty The Queen	27
Her Majesty the Queen	26
Queen Elizabeth II of the United Kingdom	21
HM the Queen	21
Elizabeth	16
Her Majesty	11
The Princess Elizabeth	9
Her Britannic Majesty	6
H.M. Queen Elizabeth II	6
Sovereign	5
HM Queen Elizabeth	5
Princess Elizabeth, Duchess of Edinburgh	4
H.M. The Queen	4
Princess Elizabeth Alexandra Mary of York	4
H.M. the Queen	4
Queen of Australia	3
Queen of England	3
HRH The Princess Elizabeth	3
Elizabeth II	3
The Sovereign	3
Elizabeth II, Duke of Normandy	3
HM Queen Elizabeth II of the United Kingdom	2
QEII	2
Queen Elizabeth II, Queen of Canada	2
The Queen of the United Kingdom	2

HRH The Princess Elizabeth, Duchess of Edinburgh	2
Queen Elizabeth II of Australia	2
Her Majesty Queen Elizabeth	2
Elizabeth the Second	2
Elizabeth II of Canada	2
ELIZABETH . II.	2
Elizabeth II, Queen of the United Kingdom	2
Monarch	2
Princess Elizabeth	2
Elizabeth Regina	2
Queen Elizabeth the second	1
Princess Elizabeth Duchess of Edinburgh	1
Queen Elizabeth II, Queen of Canada,	1
Elizabeth II of England	1
The Queens Household	1
The Queen and religion in the UK	1
HRM Queen Elizabeth II	1
Queen Elizabeth IIs Military titles	1
Mother	1
Princess Elizabeth Alexandra Mary, Duchess of Edinburgh	1
the new Queen	1
From address by HM the Queen	1
Queen Elizabeth II, Queen of the United Kingdom	1
The Queen of Great Britain	1
Elizabeth II of Tuvalu	1
Elizabeth II of Belize	1
Queen Elizabeth II of New Zealand	1
the reigning monarch	1
Elizabethan	1
sovereign	1
fierce dissent in Scotland	1
Princess Elizabeth of York	1
Mrs. Queen	1
present queen of the United Kingdom	1
Her Royal Highness The Princess Elizabeth	1

HM The Queen of Papua New Guinea	1
Queen Elisabeth II	1
Monarch Elizabeth II	1
EiiR	1
HM The Queen, Duke of Lancaster	1
Elizabeth II of Saint Kitts and Nevis	1
Queens	1
The Duchess of Edinburgh	1
H.M. Elizabeth II	1
Princess Elizabeth, later Queen Elizabeth II	1
HM Queen	1
Queen Elizabeth IIs military career	1
Elizabeth II of The Bahamas	1
Elizabeth II of Saint Vincent and the Grenadines	1
present monarch	1
The Queens Speech	1
Queen of Ceylon	1
II	1
Queen.	1
Elizabeth II.	1
Royal Crown	1
HRH Queen Elizabeth II	1
the Queen’s	1
Elizabeth II of Papua New Guinea	1
Duchess of Edinburgh	1
EIIR	1
Elizabeth Alexandra Mary	1
Brenda	1
Queen Elizabeth the Second	1
Elizabeth II of Saint Lucia	1
monarch of the	1
Buckingham Palace	1
Elizabeth II, Queen of Malta	1
Elizabeth II	1
Princesses Elizabeth	1

Queen of the United Kingdom	1
HRH Princess Elizabeth	1
Her Majesty Elizabeth II	1
British Queen	1
Royal Majesty Queen Elizabeth II	1
Queen Elizabeth II	1
the Queen of the United Kingdom	1
Her Majesty, Queen Elizabeth	1
HM Queen Eliabeth II	1
Princess Elizabeth of England	1
Elizabeth II of Barbados	1
the Queen of England	1
Queen Elizabeth II of the Commonwealth	1
Princess Elizabeth Alexandra Mary	1
the current British monarch	1
Elizabeth Windsor	1
E II R	1
HM The Queen, The Duke of Lancaster	1
Queen, Elizabeth II	1
Queen Elizabeth II of Great Britain	1
the present Queen	1
Elizabeth II of Grenada	1
Her Majesty the Queen Elizabeth II	1
Queen’s	1
Elizabeth II of Australia	1

A.4 Classified Synonym Subset

Appendix B

Stop Words

Following is the list of stop words we used.

a
a’s
able
about
above
according
accordingly
across
actually
after
afterwards
again
against
ain’t
all
allow
allows
almost
alone
along
already
also
although
always
am
among
amongst
an
and
another
any
anybody
anyhow
anyone
anything
anyway
anyways
anywhere
apart
appear
appreciate
appropriate
are
aren’t
around
as
aside
ask
asking
associated
at
available
away
awfully
b
be
became
because
become
becomes
becoming
been
before
beforehand
behind
being
believe
below
beside
besides
best
better
between
beyond
both
brief
but
by
c
c’mon
c’s
came
can
can’t
cannot
cant
cause
causes
certain
certainly
changes
clearly
co
com
come
comes
concerning
consequently
consider
considering
contain
containing
contains
corresponding
could
couldn’t
course
currently
d
definitely
described
despite
did
didn’t
different
do
does
doesn’t
doing
don’t
done
down
downwards
during
e
each
edu
eg
eight
either
else
elsewhere
enough
entirely
especially
et
etc
even
ever
every

everybody
everyone
everything
everywhere
ex
exactly
example
except
f
far
few
fifth
first
five
followed
following
follows
for
former
formerly
forth
four
from
further
furthermore
g
get
gets
getting
given
gives
go
goes
going
gone
got
gotten
greetings
h
had
hadn't
happens
hardly
has
hasn't
have
haven't
having
he
he's
hello
help
hence
her
here
here's
hereafter
hereby
herein
hereupon
hers
herself
hi
him
himself
his
hither
hopefully
how
howbeit
however
i
i'd
i'll
i'm
i've
ie
if
ignored
immediate
in
inasmuch
inc
indeed
indicate
indicated
indicates
inner
insofar
instead
into
inward
is
isn't
it
it'd
it'll
it's
its
itself
j
just
k
keep
keeps
kept
know
knows
known
l
last
lately
later
latter
latterly
least
less
lest
let
let's
like
liked
likely
little
look
looking
looks
ltd
m
mainly
many
may
maybe
me
mean
meanwhile
merely
might
more
moreover
most
mostly
much
must
my
myself
n
name
namely
nd
near
nearly
necessary
need
needs
neither
never
nevertheless
new
next
nine
no
nobody
non
none
noone
nor
normally
not
nothing
novel
now
nowhere
o
obviously
of
off
often
oh
ok
okay
old
on
once
one
ones
only
onto
or
other
others
otherwise
ought
our
ours
ourselves
out
outside
over
overall
own
p
particular
particularly
per
perhaps
placed
please
plus
possible
presumably
probably
provides
q
que
quite
qv
r
rather
rd
re
really
reasonably
regarding
regardless

regards
relatively
respectively
right
s
said
same
saw
say
saying
says
second
secondly
see
seeing
seem
seemed
seeming
seems
seen
self
selves
sensible
sent
serious
seriously
seven
several
shall
she
should
shouldn’t
since
six
so
some
somebody
somehow
someone
something
sometime
sometimes
somewhat
somewhere
soon
sorry
specified
specify
specifying
still
sub
such
sup
sure
t
t’s
take
taken
tell
tends
th
than
thank
thanks
thanx
that
that’s
thats
the
their
theirs
them
themselves
then
thence
there
there’s
thereafter
thereby
therefore
therein
theres
thereupon
these
they
they’d
they’ll
they’re
they’ve
think
third
this
thorough
thoroughly
those
though
three
through
throughout
thru
thus
to
together
too
took
toward
towards
tried
tries
truly
try
trying
twice
two
u
un
under
unfortunately
unless
unlikely
until
unto
up
upon
us
use
used
useful
uses
using
usually
uucp
v
value
various
very
via
viz
vs
w
want
wants
was
wasn’t
way
we
we’d
we’ll
we’re
we’ve
welcome
well
went
were
weren’t
what
what’s
whatever
when
whence
whenever
where
where’s
whereafter
whereas
whereby
wherein
whereupon
wherever
whether
which
while
whither
who
who’s
whoever
whole
whom
whose
why
will
willing
wish
with
within
without
won’t
wonder
would
would
wouldn’t
x
y
yes
yet
you
you’d
you’ll
you’re
you’ve
your
yours
yourself
yourselves
z
zero