Content Filtering and Enrichment Using Triplets for Text Generation

Teresa Rodríguez Ferreira

MÁSTER EN INGENIERÍA INFORMÁTICA. FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

Trabajo Fin Máster en Ingeniería Informática

20 Junio 2016

Directores:

Gonzalo Méndez Pozo
Raquel Hervás Ballesteros

Autorización de difusión

Teresa Rodríguez Ferreira

20 Junio 2016

La abajo firmante, matriculada en el Máster en Ingeniería en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: "Content Filtering and Enrichment Using Triplets for Text Generation", realizado durante el curso académico 2015-2016 bajo la dirección de Gonzalo Méndez Pozo y Raquel Hervás Ballesteros en el Departamento de Ingeniería del Software e Inteligencia Artificial, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Agradecimientos

Quiero dar las gracias a Gonzalo y a Raquel por dirigir este trabajo entre las otras cien tareas que han tenido estos meses, y por ayudarme y guiarme como siempre han hecho.

Gracias también a todo el equipo de NIL y CONCRETE, por todo su apoyo y por ofrecerme la oportunidad de trabajar en este tema con ellos.

Este trabajo está subvencionado por ConCreTe. El proyecto ConCreTe agradece el soporte financero del programa Future and Emerging Technologies (FET) que se engloba en el Seventh Framework Programme for Research of the European Commission, con la subvención FET número 611733.

Resumen en castellano

Existe una cantidad enorme de información en Internet acerca de incontables temas, y cada día esta información se expande más y más. En teoría, los programas informáticos podrían beneficiarse de esta gran cantidad de información disponible para establecer nuevas conexiones entre conceptos, pero esta información a menudo aparece en formatos no estructurados como texto en lenguaje natural. Por esta razón, es muy importante conseguir obtener automáticamente información de fuentes de diferentes tipos, procesarla, filtrarla y enriquecerla, para lograr maximizar el conocimiento que podemos obtener de Internet.

Este proyecto consta de dos partes diferentes. En la primera se explora el filtrado de información. La entrada del sistema consiste en una serie de tripletas proporcionadas por la Universidad de Coimbra (ellos obtuvieron las tripletas mediante un proceso de extracción de información a partir de texto en lenguaje natural). Sin embargo, debido a la complejidad de la tarea de extracción, algunas de las tripletas son de dudosa calidad y necesitan pasar por un proceso de filtrado. Dadas estas tripletas acerca de un tema concreto, la entrada será estudiada para averiguar qué información es relevante al tema y qué información debe ser descartada. Para ello, la entrada será comparada con una fuente de conocimiento online.

En la segunda parte de este proyecto, se explora el enriquecimiento de información. Se emplean diferentes fuentes de texto online escritas en lenguaje natural (en inglés) y se extrae información de ellas que pueda ser relevante al tema especificado. Algunas de estas fuentes de conocimiento están escritas en inglés común, y otrás están escritas en inglés simple, un subconjunto controloado del lenguaje que consta de vocabulario reducido y estructuras sintácticas más simples. Se estudia cómo esto afecta a la calidad de las tripletas extraidas, y si la información obtenida de fuentes escritas en inglés simple es de una calidad superior a aquella extraida de fuentes en inglés común.

Palabras clave

- Extracción de Información
- Inglés Simple
- Tripletas
- Filtrado de Información
- Enriquecimiento de Información

Abstract

There is an extremely large amount of information on the Internet about almost every topic, and every day this information is constantly expanding. Theoretically, computer programs could benefit from this huge source of information in order to establish new connections between concepts, but this information often appears in unstructured formats such as plain text. For this reason it is very important to be able to automatically obtain this information, process it, filter it and enrich it with data from different sources, in order to maximise the knowledge that we can obtain from the Internet.

This project presents two different parts. In the first one information filtering is explored. The system's input consists in a series of triplets provided by the University of Coimbra (they in turn obtained them through a process of information extraction from natural language text). However, due to the complexity of this extraction task, some of the triplets are of questionable quality, and they must undergo a filtering process. Given this set of triplets about a specific topic, the input will be studied to find out which information is relevant to the subject and which information should be discarded. In order to do this, the input provided will be compared to an online knowledge base.

In the second part of this project, information enrichment is explored. Several online text sources written in natural language are use and information is extracted from them that could be relevant to the desired topic. Some of these text sources are written in common English and some in Basic English, a controlled subset of the language which has a reduced vocabulary and simpler sentence structures. The way in which this affects the quality of the triplets extracted is studied, and whether information retrieved from sources written in Basic English has a higher quality than that extracted from texts in common English.

Keywords

- Information Extraction
- Basic English
- Triplets
- Information Filtering
- Information Enrichment

Table of Contents

Chapter 1

Introduction

The amount of information available on every topic nowadays on the internet is overwhelming. From online newspapers, to magazine articles, to independent blogs, to online courses and tutorials, to collaborative encyclopedias... This richness of information is at the same time a blessing and a curse. On one hand we are very fortunate to have access to all this content nowadays, and also to be able to contribute to it. But on the other hand, such a large amount of information is hard to manage.

The Internet is an ever growing endless source of knowledge. So endless in fact, that it can be very difficult to obtain clear and structured information on a certain topic. Articles are usually written in natural language, which is ideal for a human to read, but unstructured and hard to process by a computer. A human, on the other hand, would have a very hard time going through all the existing articles on a specific subject they wanted to learn about.

What if this unstructured information could somehow be processed and combined with data from different sources to obtain a large collection of structured knowledge on a certain topic? We would have a very large amount of information on any topic, all grouped in the same place, and it could be easily processed by a computer. The field of Information Extraction focuses on addressing this task.

1.1 Motivation

This project is part of a larger scale European project named ConCreTe (Concept Creation Technology). The goal of ConCreTe is to study conceptual creativity in humans and machines. In the long term it aspires to achieve behaviour in computers that is comparable to human creativity, autonomously and interactively. ConCreTe uses a web service called ConCreTe Flows in which different modules can be activated and combined to produce the desired result, thus allowing the user to create different pipelines depending on the task that they face. The project includes tasks such as poetry generation, song lyrics generation, semantic aware methods for evolutionary art, generation of comparisons and riddles, summarisation of documents or semantic information extraction, among others. The work described in this document focuses on information filtering and information enrichment, producing results that can be used as aid for text summarisation, riddle generation and a number of other tasks. It is an intermediate step that allows for other parts of the project to function. ConCreTe is a collaboration between universities in London, Coimbra, Helsinki, Ljubljana, Twente and Madrid.

Obtaining structured data from unstructured information such as plain text is an important task. When we have structured information that a computer can process, it can be used to establish connections between topics or concepts and to obtain a richer and deeper knowledge on the subject.

Unfortunately, the internet, although an infinitely rich source, is not always the most reliable source of knowledge. The information extracted is not always correct or precise, and it should be filtered to discard as much false or irrelevant data as possible.

1.2 Objectives

The purpose of this project is to have as much correct information as possible on any given topic, discarding as much irrelevant detail as possible. The aim is to be able to use

the Internet to its full potential by obtaining information from different texts written in natural language, keeping only the information related to a specified subject. This can have applications such as automatically generating an abstract on the topic, poetry generation, or any kind of task which requires having structured information on a subject.

As mentioned above, since the information obtained from the internet is not always correct, it must undergo a filtering process. This is one of the tasks that this project will face. A system will be created that is able to obtain information from different sources, and the data will be filtered in order to keep only the relevant details.

If knowledge is obtained from one source only, it will be limited and we will not be taking full advantage of the potential of the internet. For this reason, the second task that will be faced in this project will be enriching the information available with data from different sources. The system will be able to extract knowledge on a subject from any plain text in natural language.

One of the particular goals of the European project consisted in concept blending. Given two different concepts, their properties can be combined and blended in order to create a new concept which is a mixture of the two. In this scenario it is useful to extract as much information as possible from these two concepts from any possible resource. The information on these two concepts received as input is in the form of triplets that represent a semantic graph. However, the tool must be flexible to allow its use by people external to ConCrete. For this reason the project should be built using modules which can be activated or ignored, as the user chooses. For instance a user who is not part of the project may not have any input to introduce into the system, so the modules in charge of parsing and filtering the input can be deactivated.

The following are the specific objectives that this project aims to achieve:

- Given a set of triplets that represents information regarding a certain topic, filter this information in order to discard as much incorrect or irrelevant information as possible, keeping only triplets related to the topic.

- Explore different online resources from which additional information could be extracted to enrich the existing triplets.
- Build a tool that is able to extract triplets representing definitions or properties of a topic from any text written in Natural Language. These new triplets will be used to enrich the available information.

1.3 Document Structure

Chapter 2 describes the State of the Art regarding Information Extraction, hyponymy, graph representations of text, information filtering and Content Determination.

Chapter 3 contains a short analysis of the possible tools that could be used for text analysis and why Freeling was chosen, and a description of the online knowledge bases considered. The programming language used and the internal structure of the project are also established at this point.

Chapter 4 describes the input provided for the system and explains the methods used and process followed to filter the data contained in the input.

Chapter 5 fully explains the behaviour of the language analysis tool suite used to analyse the text. It also contains an explanation about how this tool is used to aid in Information Extraction from Natural Language text sources. Different information sources that can be used for this task are examined, and the results obtained with each one are compared. Finally there is an explanation on how another tool, ConceptNet 5, can be used to further enrich the information available on a certain subject.

Lastly, the final results obtained are discussed in Chapter 6, along with a summary of possible future work.

Two annexes have also been included. The first one contains a paper on Improving Information Extraction from Wikipedia Texts using Basic English, that was published in LREC 2016 Language Resources and Evaluation Conference. The second annex details how the project should be configured before use.

Chapter 2

State of the Art

2.1 Information Extraction

Information Extraction (IE), the process of automatically extracting structured information from unstructured texts, has progressed greatly over the last few decades [Etzioni et al., 2008]. Although the ambiguous nature of plain text makes the task a difficult one, it is possible to find many systems that have obtained quite good results. For instance TextRunner [Yates et al., 2007], one of the pioneers in Open Information Extraction (OIE), is able to obtain high-quality information from text in a scalable and general manner.

OIE [Yates et al., 2007], is a domain independent extraction paradigm that can extract relations from corpora without any user interaction. Text Runner is an example of a highly scalable OIE system.

Rusu et al. [Rusu et al., 2007] present an approach to extracting triplets from sentences by relying on well known syntactic parsers for English.

Shinyama and Sekine [Shinyama and Sekine, 2006] propose in their work the idea of "unrestricted relation discovery", which attempts to extract information without the need to specify the desired relations, discovering and saving any valuable relations it finds in the text.

Some systems, such as Rapier [Mooney, 1999], focus on extracting information by using pattern matching. This tool uses sample documents and filled templates to obtain pattern

match-rules, which can later be used to obtain relations from texts.

WebSets [Dalvi et al., 2013] is an unsupervised information extraction technique which obtains concept-instance pairs from HTML tables in a given corpus.

Wikipedia is considered an excellent source of texts for IE systems due to its broad variety of topics and advantageous characteristics such as the quality of the texts and their internal structure. Therefore there are some IE systems that work using Wikipedia texts and/or their structured metadata, like Wanderlust [Akbik and Bross, 2009] or WOE (Wikipedia-based Open Extractor) [Wu and Weld, 2010].

Weldet al. [Weld et al., 2009] restrict their process to infoboxes, tabular summaries of an article's salient details which are included in a number of Wikipedia pages. Wanderlust [Akbik and Bross, 2009] is an algorithm that automatically extracts semantic relations from natural language text. The procedure uses deep linguistic patterns that are defined over the dependency grammar of sentences. Due to its linguistic nature, the method performs in an unsupervised fashion and is not restricted to any specific type of semantic relation. The applicability of the algorithm is tested using the English Wikipedia corpus.

WOE (Wikipedia-based Open Extractor) [Wu and Weld, 2010] is a system capable of using knowledge extracted from a heuristic match between Wikipedia infoboxes and the corresponding text. In particular, Krawczyk et al. [Krawczyk et al., 2015] present a method of acquiring new ConceptNet triplets automatically extracted from Japanese Wikipedia XML dump files. In order to check the validity of their method, they used human annotators to evaluate the quality of the obtained triplets.

In this project Information Extraction is used in the second phase of the work. In order to enrich the information available on a certain subject, data is retrieved from text sources written in natural language (in English only) and combined with the available triplets in order to maximise the knowledge on the topic.

2.2 Hyponymy

A hyponym is a word or phrase whose semantic field is more specific than another term, its hypernym. The hyponym is a specific instance of the hypernym. These are often called IsA relations in computer science, in which concept A, the hyponym, is an instance of concept B, the hypernym. It is important to highlight that hyponymy is a transitive relation. If concept A is an instance of concept B, and concept B is an instance of concept C, then concept C is also concept A's hypernym. In this project hyponymy is used as one of the fundamental relations that are extracted from texts and represented in the triplets, although it is not the only relation used. IsA relations can be used to define concepts and a concept's definition is usually its most important piece of information. This relation can also be used to find links between concepts that seem unrelated, but appear in the same document, such as concepts A and C in the previous example of transitivity. If the concept "animal" appears in a text concerning hamsters, and we know that a hamster is a mammal, by extracting information that indicates that a mammal is an animal we can also learn that a hamster is an animal. The transitive relation of hyponymy also allows a distinction to be made between concepts which are connected to each other indirectly, such as concept A and concept C in the previous example, and concepts which are not connected.

There is a lot of research on the automatic and semi-automatic extraction of hyponymy or IsA relations from natural language text. Hearst [Hearst, 1992] presented in 1992 six different patterns, now known as Hearst patterns, for automatically retrieving hyponymy relations from natural language text. Mititelu [Mititelu, 2008] presented an experiment in which she identified hyponymy patterns in corpora in English and Romanian. Shinzato and Torisawa [Shinzato and Torisawa, 2004] also describe an automatic method for obtaining hyponymy relations from HTML documents on the Internet. Also, Sumida and Torisawa [Sumida and Torisawa, 2008] present a method for extracting hyponymy relations from the Japanese Wikipedia by using a machine learning technique and pattern matching.

2.3 Information filtering

Information filtering is the act of discarding irrelevant, unwanted or redundant information from a certain source automatically or semi-automatically. This is required especially in a system such as this one, which retrieves information from any text source on the Internet, since some kind of control must be established when dealing with this amount of information. In this project, the information filtering is carried out in the first phase of the work, in order to remove unwanted or incorrect triplets from the input received. Information filtering is widely used in, for instance, spam filtering.

Hanani et al. [Hanani et al., 2001] describe the underlying concepts of information filtering systems and the techniques used to implement them. Sheth [Sheth and Maes, 1993] explains how different techniques can be combined to develop a semi-automated information filtering system which dynamically adapts to the changing interests of the user.

Delgado et al. [Delgado et al., 1998] describe a multi-agent system, RAAP (Research Assistant Agent Project) that combines content-based information filtering with collaborative information filtering within complex and open environments, such as the Internet.

Yu et al. [Yu et al., 2002] propose a probabilistic framework that also unifies content-based information filtering with collaborative information filtering, named collaborative ensemble learning. Their work combines a society of users' preferences to predict an active user's preferences.

There has been a lot of research in this field, and in this project a simple approach to information filtering by using data from a collaborative online knowledge base is presented.

2.4 Content Determination

Content Determination is one of the several subtasks of Natural Language Generation (NLG) [Reiter et al., 2000], and is in charge of deciding which information should be included in a text. In this system this task is present in a simple manner, and the user can decide how selective to be with the information included. In the first phase of the work, information filtering, this task is present when deciding which information is not relevant to the subject and should not be included in the final output. In the second phase of the work content determination is present when deciding which relations should be included in the information enrichment phase. The most simple approach is for the user to include all the relations that the system can find in the text, but the user can also specify which relations they are interested in including, and the system will ignore the rest. This is useful since depending on the purpose that the output is intended to have, different relations may be more or less useful. For instance, if the end goal is to generate a brief summary of a topic, relations such as "IsA" (which define the concept) and "HasProperty" (which give additional information about the concept's properties) might be enough to give a good overview of the topic. If, however, the task is a more creative one, other types of relations may be more convenient. Say for instance that a short text for children is being generated where each concept is a different animal and the text describes where each of them would live. In this situation it would be useful to extract "AtLocation" relations only.

This task has been explored in many different contexts. For instance Dale and Haddock [Dale and Haddock, 1991] talk about content determination in the generation of referring expressions, or what information should be included in computer-generated descriptions of objects and people. Stripada et al. [Sripada et al., 2001] propose a general architecture for content determination in data summarisation systems.

Chapter 3

Preparing the Project

Before going into details about the steps followed to achieve the objectives for this project, I will talk about the tools I considered using and why I decided on FreeLing and ConceptNet. Then I will go on to describe the design and implementation details of the system.

3.1 Text analysis tool

In order to extract information from texts written in natural language, a text analysis tool must be used to aid in the extraction of relevant pieces of information from each sentence. There are many tools available that work with different languages and that offer different functionalities related to text analysis.

The tools considered at the beginning of the project were Maltparser [Nivre et al., 2006], FreeLing [Carreras et al., 2004] and GATE [Cunningham et al., 2011]. All three can be used for text analysis in different languages.

MaltParser is a system for data-driven dependency parsing, that is able to induce a parsing model from treebank data and to parse new data using an induced model. Instead of constructing a parser given a grammar, this tool constructs the parser given a treebank (a parsed text corpus that annotates syntactic or semantic sentence structure). The system has been used with different languages giving a dependency accuracy of 80–90%. This tool can function in two different modes. The first is a learning mode where given specifications of a parsing algorithm, a feature model and a learning algorithm, it receives a dependency

treebank as input and induces a classifier for predicting parser actions. The second is a parsing mode where, using the same parsing algorithm and feature model used during the learning mode, it receives a set of sentences and constructs and produces a projective dependency graph for each sentence, using the classifier induced in the previous mode.

FreeLing is an open source language analysis tool suite which provides language analysis functionalities that can be used in NLG applications. Some of the services that it offers are text tokenization, sentence splitting, morphological analysis, suffix treatment, compound-word recognition, flexible multiword recognition, contraction splitting, probabilistic prediction of unkown word categories, named entity detection, rule-based and statistical dependency parsing, coreference resolution or semantic graph extraction. It also works with a variety of languages (English, Spanish, Portuguese, Italian, French, German, Russian, Catalan, Galician, Croatian, Slovene, among others).

GATE is a free open-source infrastructure for developing and deploying software components that process natural language. The tool includes a desktop client for developers, a workflow-based web application, a Java library, an architecture and a process. It also supports different languages such as English, Chinese, Arabic, Bulgarian, French, German, Hindi, Italian, Cebuano, Romanian, Russian or Danish.

The reason for FreeLing being chosen as the main tool for this project was that there was already an existing project in the research group that the system was created for that used this tool, and since I had access to it this saved time in learning how to correctly install the tool and incorporate it into the project. There was a drawback to this decision, which was that the version of FreeLing used was FreeLing 2.2, when at the time of the start of the project, the current version was FreeLing 3.1 (at the time that this document was written, the latest version of FreeLing is 4.0). This means that if the tool is updated, the results may vary slightly. In this project the tool has been used mainly for morphological analysis and dependency parsing. The system works in English.

3.2 ConceptNet

For information filtering, ConceptNet 5 [1] was chosen as the reference knowledge base. ConceptNet is an open source, free online semantic network that offers a very large amount of information on concepts in natural language. It is also multilingual. It contains labeled *nodes* or *terms* that represent concepts, words or short phrases in natural language and the relations (represented by labeled edges) that exist between them. The knowledge available in ConceptNet comes from a variety of resources, such as crowd-sourced resources (for instance Open Mind Common Sense), games with a purpose (such as Verbosity and nadya.jp), and expert-created resources (such as WordNet).

The edges that connect nodes representing relations are assigned a weight, indicating how important and informative that edge should be. This way some relations have priority over others. Edges contain the following fields of information:

- id: a unique ID for the edge
- uri: the URI (Uniform Resource Identifier) of the assertion being expressed by the relation. The uri is unique among assertions, but not among edges, because the same assertion can be expressed by multiple edges
- rel: the URI of the relation
- start: the URI of the source concept of the relation
- end: the URI of the target concept of the relation
- weight: the strength with which this edge expresses the assertion
- sources: the sources from which the information was extracted

[1] http://conceptnet5.media.mit.edu/

- license: a URI representing the Creative Commons license that governs this data
- dataset: a URI representing the dataset of a source from which this edge was extracted
- context: not used in ConceptNet 5.2. The value is always "/ctx/all" for compatibility.
- features: a list of three identifiers for features, which are essentially assertions with one of their three components missing. They can be useful in machine learning for inferring missing data
- surfaceText: the natural language text where the statement was obtained from. Since not every statement was derived from natural language input, this may be null. The locations of the start and end concepts in the text will be surrounded by double brackets. An example of a surfaceText is "[[a parrot]] is [[a bird]]".

Some standard relations exist within ConceptNet and appear often throughout its content. Example of these are “RelatedTo”, “IsA”, “PartOf”, “MemberOf”, “HasA”, “UsedFor”, “CapableOf”, “AtLocation”, “Causes”, “HasProperty”, “MotivatedByGoal”, “Desires”, “CreatedBy”, “Synonym”, “Antonym”, “DefinedAs”...

Relations extracted from natural language text sources are not always contemplated within this set of pre-established edges. Sometimes relations can have more unusual names, but these generally appear often throughout multiple sources.

ConceptNet offers a REST API which is very easy to access from the project. All the documentation necessary to be able to use the API is contained in the ConceptNet wiki.

3.3 Text Sources

As for information enrichment, the sources where the triplets are extracted from must contain definitions and properties of concepts. The most appropriate resources for this purpose are dictionaries and encyclopedias. Dictionaries provide succinct definitions and a brief and usually more technical overview of the concept's most salient properties. Encyclopedias, on the other hand, contain more general information and in greater quantity. These were the types of sources used in this project, but any type of text written in natural language in English could be used if the user so wished.

The knowledge sources used were English Wikipedia, Simple English Wikipedia and Simple English Wiktionary (from now on referred to as Simple Wikipedia and Simple Wiktionary). All three are open source, free collaborative resources that are widely used and always expanding. For this reason they provide and excellent source of information for enriching the input data. A comparison between the results obtained from these three sources can be found in section 5.4.

3.4 Design

This work consists of two main parts, a filtering process and an enrichment process. The project is divided into individual modules which represent different tasks, and these modules can be activated or not as the user requires. If the user provides the system with input information, all modules can be used, but it is also possible to run the program with no input (the step where the input information is filtered would be ignored, and only the information enrichment and output parsing tasks would be carried out. Figure 3.1 shows the pipeline for the system.

The first module in the system is in charge of parsing the input data found in the input folder into the internal representation of the information that the system uses. In this case, the input data consists of text files provided by the University of Coimbra, each containing a set of triplets for a different concept. A full description of the input data for the system

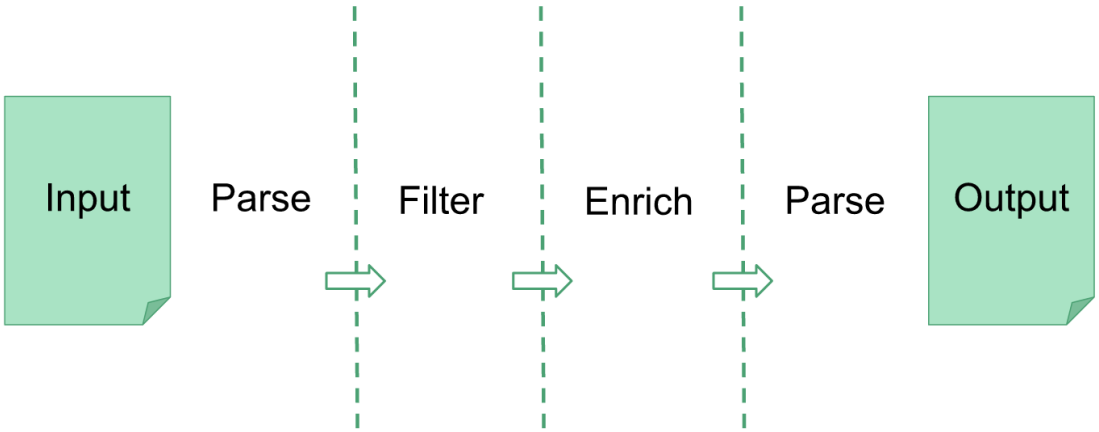

Figure 3.1: *System pipeline*

can be found in Section 4.1. If this module is ignored, the system can work without using input information, it can be used in order to simply find information online regarding a certain subject and store it.

Another module is in charge of information filtering. This module makes use of ConceptNet's API to compare the information from the input with the information available in the online knowledge base. Triplets which are considered incorrect or irrelevant to the topic are discarded and removed from the system. Just as in the previous module, this can be skipped if no input is provided for the system. A more in depth explanation of the process of information filtering can be found in Chapter 4.

One module executes queries using the different text sources in order to obtain more information on the topic, and stores this information in the system's internal format. The sources used in the project so far have been Wikipedia, Simple Wikipedia and Simple Wiktionary, but the user can introduce any online text source they want. There is also an extra step in this module that consists in cleaning the information obtained from the sources so that any kind of markup tags are removed and only the text remains. The final text obtained from each source is stored in the queries folder. There will be one text file for each concept from each source. So if the concept "dog" has been found in all three sources, there will be a dogWikipedia file, a dogSimpleWikipedia file, and a dogSimpleWiktionary file.

Another module takes the filtered data and enriches it further, using the information obtained from different knowledge sources by making use of the previous module (the user may specify which source they want to use or if they want to use them all at the same time). Triplets are extracted from the text files stored in the previous module, and are stored in JSON files in the triplets folder. Once again, each concept has its own file. The full process of information enrichment is explained in Chapter 5.

One last module unifies all the information and stores it in a file that can then be used as input for other systems, for instance a system that automatically generates abstracts on certain topics. This final result is stored in the output folder.

The programming language chosen for the system was Java. The previously existing project that incorporated FreeLing 2.2 was also written in Java, so it was easy to incorporate the tool into the new system using the same language.

Chapter 4

Information Filtering

The first step that must be taken in this project is checking that the information already available to work with is correct, and making sure to remove as much irrelevant or incorrect information as possible. In order to illustrate the process and to make it easier to understand, the example of the concept "hamster" will be used throughout the rest of the document.

4.1 Input triplets

The input for the system could be represented in many different formats, such as a graph, a set of triplets or simply as plain text. The system can be adjusted to work with different types of formats, but for this project only one has been used. I will be working with a set of triplets as input. They consist in sets of three words or groups of words: a source concept, a target concept, and a relation that exists between the two. These triplets have been provided by researchers from the University of Coimbra who work as part of the CONCRETE team for which this system was developed.

The triplets provided are contained in a JSON file where all of the triplets in the file refer to a specific topic, or *main entity* (which is specified in the file name). The triplets follow a *relation(source-concept, target-concept)* structure, and represent different relations that exist between two concepts. The following is an example of a text from which a series of triplets were extracted. The topic of the text was "Hamsters".

Hamsters are rodents belonging to the subfamily Cricetinae. The subfamily contains about 25 species in six or seven genera.They have become established as popular small house pets.They are a bit like a mouse. Wild hamsters live in the desert, but people all over the world keep domesticated hamsters as pets. In the wild, hamsters are crepuscular and stay underground during the day. They feed on seeds, fruits, and vegetation, and occasionally eat burrowing insects. Hamsters are distinguished by their large cheek pouches, and relatively short tail. They use their long cheek pouches (extending to their shoulders) to carry food back to their burrows.

There are six main types of hamsters: the Syrian hamster (the kind most people have as pets), winter whites, campbells, theRussian dwarf hamster (a hybrid of winter white and campbell hamsters), winter whites and campbells are two type of Russian (despite what many pet shops say), the Chinese hamster, and the Roborovski hamster. The Chinese hamster has a long tail. All Syrian hamsters are the descendants of 12 baby hamsters found in Syria in 1930.

Pet hamsters like to live in cages with wood shavings or recycled newspaper (only if non-toxic).Fluffy bedding is dangerous for hamsters. Plain toilet paper is also a cheap, safe option, but not straw as it is very dry and they may choke on it. Hamsters eat mostly hamster food sold at a pet shop, but they also eat almost any vegetables and fruits in very tiny portions. Hamsters store food in the sides of their mouths.

Hamster teeth grow constantly. Wooden blocks and some hard food can be used for this. Most hamsters also have a wheel to run on for exercise. A 6 inch (Syrian) wheel is recommended for most dwarf hamsters, and at least an eight inch wheel for Syrians. Some people get other kinds of exercise equipment for their hamsters, like an empty ball that the hamster can roll around the floor or a long network of tubes with air holes for it to crawl through. They should live

in an aquarium (without water) or cage. When letting your hamster out always keep an eye on it. Hamsters may look slow, but are truly very fast. Keep your hamster away from any electric wires, since they like to chew on them. Hamster are not recommended for young children.

Figure 4.1 shows the triplets extracted from this text by the University of Coimbra.

The triplets in Figure 4.1 present a problem that must be addressed. The triplets that contain the main entity as the source or target concept can be accepted as relevant, but some of the triplets seem unrelated to the main entity, even though they have all been extracted from the same text. Triplets such as *property(toxic, dangerous)* or *isa(contains, bit)* maintain no obvious relation with hamsters. For this reason, the input received must be filtered before it is included in the system. The information must be analyzed to check how strongly related it is to the main entity, and whether it should or should not be introduced in the system.

```
1  isa(contains, bit).
2  property(genus, seven).
3  isa(genus, bit).
4  isa(mouse, pet).
5  keep(people, pet).
6  isa(hamster, crepuscular).
7  stay(hamster, day).
8  fee_on(hamster, seed).
9  isa(vegetation, insect).
10 eat(vegetation, insect).
11 property(tail, short).
12 isa(tail, pouch).
13 property(pouch, long).
14 use(tail, pouch).
15 property(hamster, six).
16 isa(hamster, type).
17 property(hamster, long).
18 isa(hamster, tail).
19 property(tail, long).
20 have(hamster, tail).
21 property(toxic, dangerous).
22 be(bedding, dangerous).
23 eat(non, vegetable).
24 eat(toxic, vegetable).
25 eat(bedding, vegetable).
26 isa(syrian, aquarium).
27 shall_live_in(inch, aquarium).
28 shall_live_in(syrian, aquarium).
29 isa(cage, eye).
30 keep(when, eye).
31 may_look(hamster, slow).
32 be(hamster, fast).
```

Figure 4.1: *Triplets for "hamster" generated by the University of Coimbra*

4.2 Information filtering using ConceptNet

In order to filter the information contained in the input data and remove triplets which are unrelated to the main entity, an external knowledge source must be used to compare the concepts mentioned in the input to concepts which maintain a direct relation with the main entity.

The knowledge source chosen for this task was ConceptNet 5 [1]. It is a widely used, open source, free online semantic network. It contains *nodes* or *terms* that represent concepts, words or short phrases in natural language and the relations (represented by edges) that exist between them. ConceptNet provides a REST API which can be used to perform queries that allow the user to obtain information on certain nodes or the edges that connect them, obtain information given certain properties or measure the semantic distance between nodes to see how related they are.

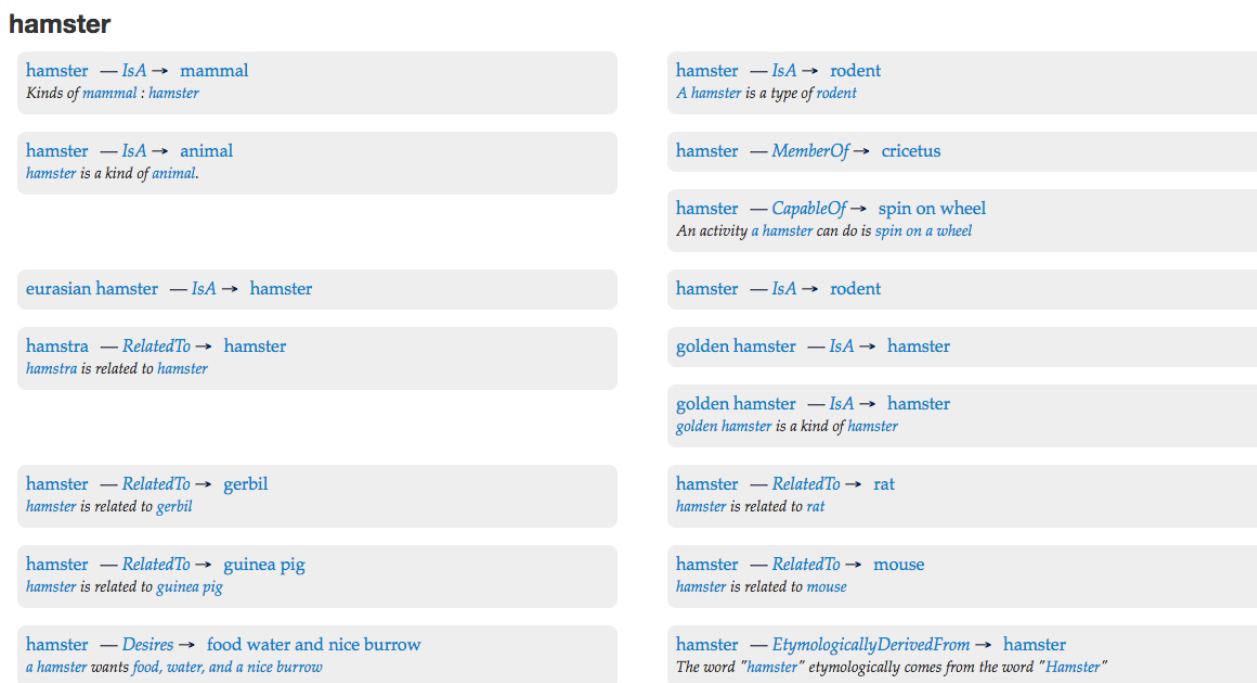

Figure 4.2: *Some nodes and relations for the concept "hamster" in ConceptNet*

Figure 4.2 displays some of the data that ConceptNet provides when searching for the concept "hamster". This information can be used to find out whether the concepts provided by the University of Coimbra hold any relation to the main entity that is being discussed.

[1] http://conceptnet5.media.mit.edu/

4.2.1 Example of information filtering for the concept "hamster"

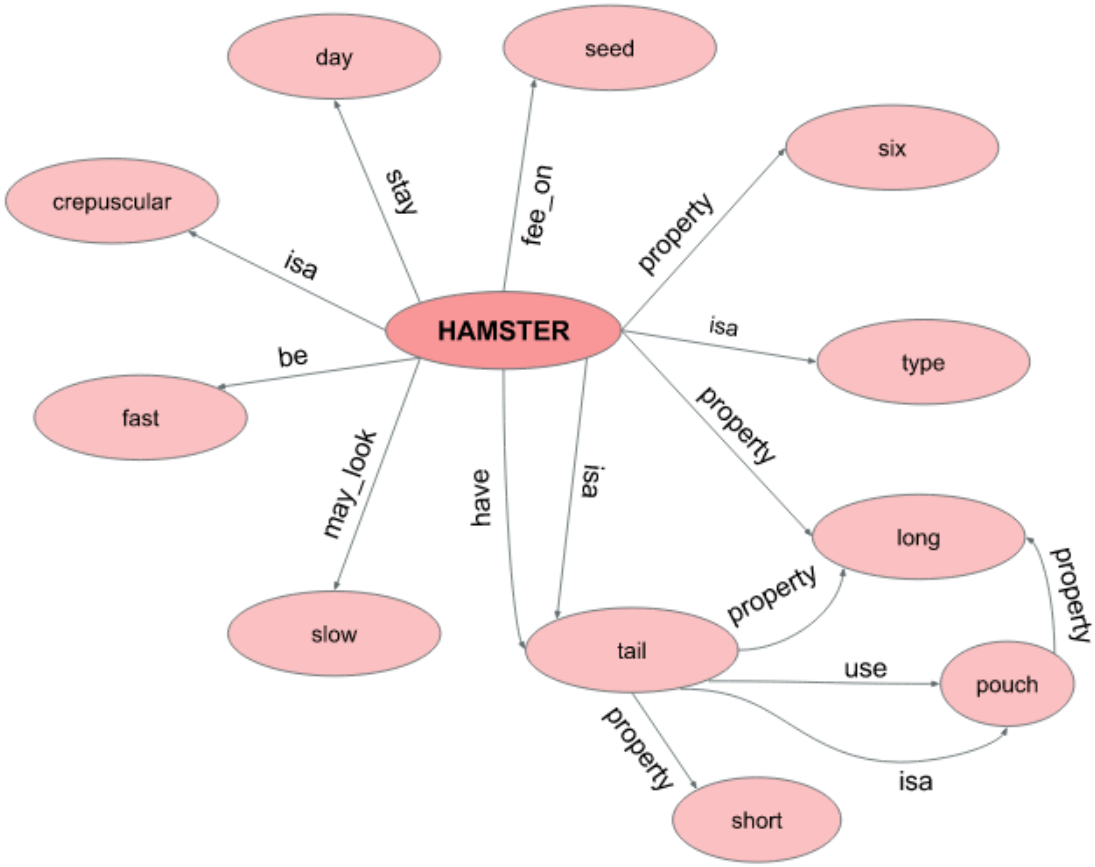

Figure 4.3: *Triplets containing the concept "hamster" from the University of Coimbra*

The graph in Figure 4.3 represents the triplets provided by the University of Coimbra which are directly connected to the concept "hamster" (they contain the concept in their source or target, in this case only in their source, or they are related to another concept which is connected to the main entity). These triplets will automatically be accepted as relevant and they will be included in the system. Figure 4.4 shows the triplets that are left with no direct connection to the main entity, referred to as spare triplets from now on.

Using a query to ConceptNet, a series of concepts will be obtained which are in some way related to "hamster". By using different parameters, a specific number of concepts can

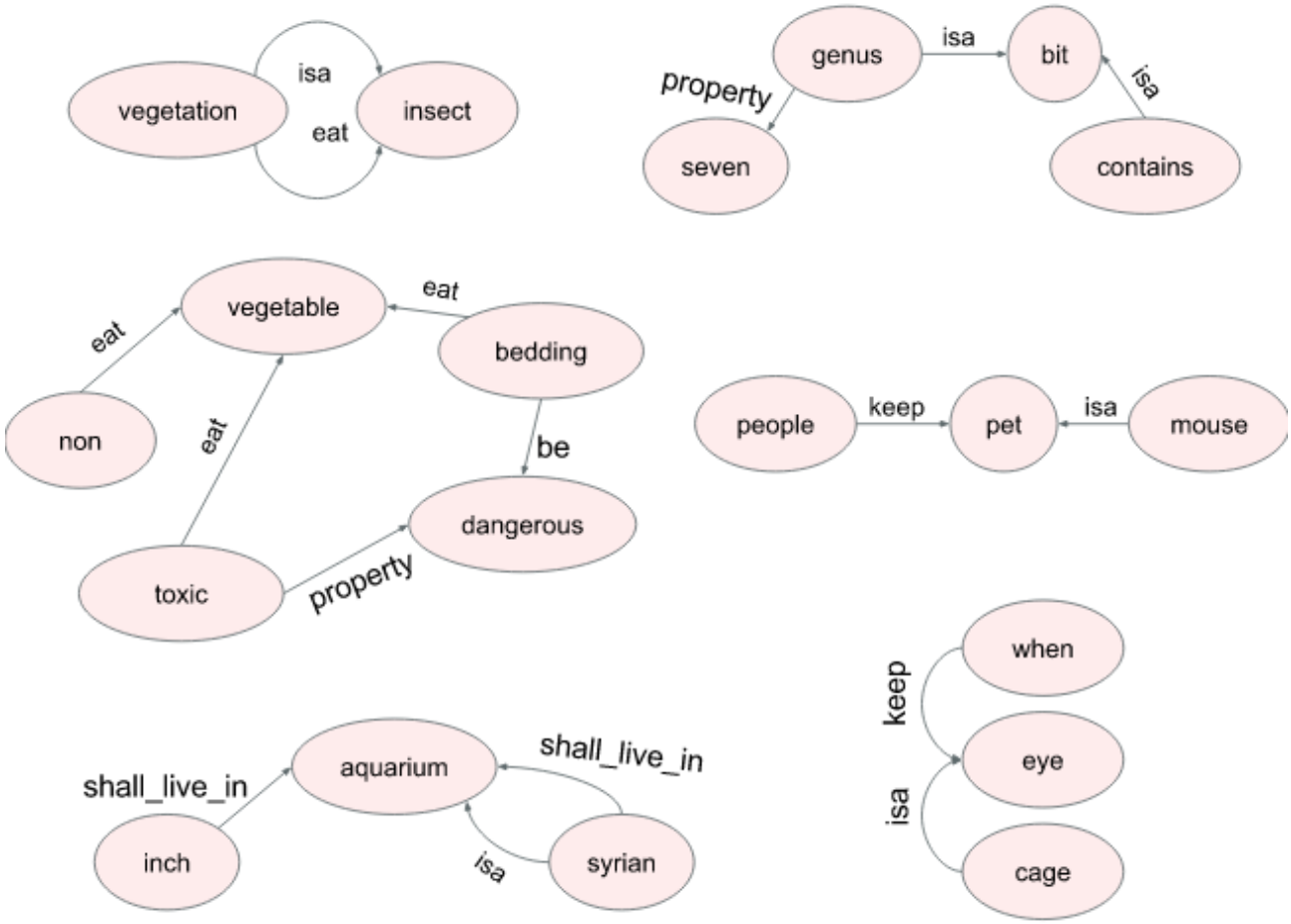

Figure 4.4: *Spare triplets for the concept "hamster" from the University of Coimbra*

be obtained instead of the full list. For this example I indicate that I wish to obtain the first 15 concepts that hold some kind of relation with the main entity. After discarding a couple of relations that appeared twice, the result is as shown in Figure 4.5. The concept "mouse" is the only one that appears both in the spare triplets from the input and in ConceptNet, so it is a good candidate to be considered as relevant information.

There is one more option for ConceptNet to help distinguish if one of the spare concepts is related to the main entity. By using the association option in the query, ConceptNet is able to show the degree in which two concepts are similar. It is possible to iterate through all of the spare concepts and check how strongly related they are to the main entity. The concept

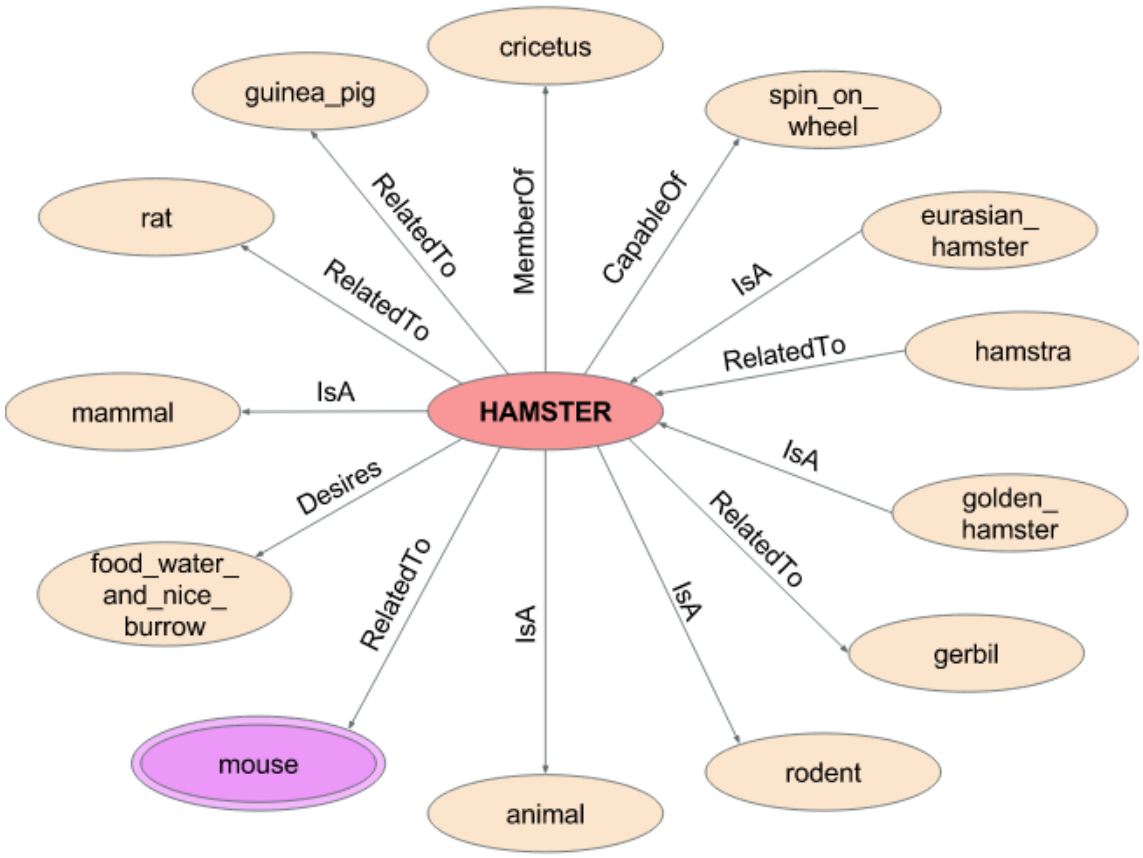

Figure 4.5: *Nodes and edges for "hamster" obtained from ConceptNet*

"mouse", for instance, shows a 70% degree of similarity to "hamster", whereas "aquarium" is only 7% similar to the main entity. A minimum percentage of similarity should be established to determine whether two concepts are strongly enough related. The problem with this approach is that it would be possible to find out how strongly related two concepts are, but not which type of relation they maintain. This information is useless if the relation is unknown, since there is no way to connect the spare concept to the main entity and store it in the system. For this reason, this approach was discarded.

Now that we know thanks to ConceptNet that *hamster* - *RelatedTo* - *mouse*, we can include this information in the system, together with the concepts that are related to "mouse"

according to the University of Coimbra. The final triplets that will be accepted into the system at this stage are shown in Figure 4.6.

```
isa(mouse, pet).
keep(people, pet).
RelatedTo(hamster, mouse).
isa(hamster, crepuscular).
stay(hamster, day).
fee_on(hamster, seed).
property(tail, short).
isa(tail, pouch).
property(pouch, long).
use(tail, pouch).
property(hamster, six).
isa(hamster, type).
property(hamster, long).
isa(hamster, tail).
property(tail, long).
have(hamster, tail).
may_look(hamster, slow).
be(hamster, fast).
```

Figure 4.6: *Filtered triplets for "hamster"*

Chapter 5

Information Enrichment

Although software applications could theoretically benefit from the huge amount of information in the Web, they usually face the problem of this information appearing in the form of unstructured data, like plain text. The possibility of automatically extracting the knowledge underlying this plain text is therefore becoming increasingly important. With this information, the system can be enriched with additional knowledge from the innumerable text sources available on the Internet.

5.1 Retrieving data for information enrichment

The first step of this task consisted in obtaining the text on the specified topic from any or all of the used text sources (English Wikipedia[1], Simple English Wikipedia[2] and Simple English Wiktionary[3]).

By performing a simple query, the full text from the article can be obtained in a JSON file. The text was parsed to remove the markup language, images, etc. and only the plain text was stored in a text file. The topic or main entity was stored in the file name. The information for each specified topic was obtained from the corresponding web page from each source.

[1] http://www.wikipedia.org
[2] http://simple.wikipedia.org
[3] http://simple.wiktionary.org

5.2 Text analysis for information enrichment using FreeLing

Once the full text of the article has been obtained and temporarily stored inside the queries folder, the next step is to parse it using FreeLing, splitting the text into sentences and storing them in the system's internal format which will be explained further on. Once the text has been split into sentences, each sentence is first morphologically analysed using the tool. FreeLing assigns a POS (Part Of Speech) tag, or grammatical tag, to each word. This consists in marking up each word based on its definition and on its context, it not only considers the word individually, but also takes into account the the other words that surround it in the sentence. Since words can be ambiguous and have a different meaning and function depending on the words that surround them, FreeLing assigns each one the possible tags that might apply to it along with a probability (between 0 and 1) of that being the correct tag. The tag with the highest probability is the one that is finally assigned to that word. Figure 5.1 shows the Penn TreeBank tags that FreeLing uses for English texts. These treebank tags are tags which offer semantic information about each word and are necessary for morphological analysis and dependency parsing. The Penn Treebank in particular was the first large-scale treebank to be published and it is widely used in computational linguistics.

Below is an example of the morphological analysis that FreeLing would perform for the sentence "A hamster is a furry rodent a little larger than a mouse and with a very short tail." using POS tagging.

A a DT 0.333333 a NN 0.333333 a NNS 0.333333

hamster hamster NN 1

is be VBZ 1

a a DT 0.333333 a NN 0.333333 a NNS 0.333333

furry furry JJ 1

Tag	Description	Example
CC	conjunction, coordinating	*and, or, but*
CD	cardinal number	*five, three, 13%*
DT	determiner	*the, a, these*
EX	existential there	*there were six boys*
FW	foreign word	*mais*
IN	conjunction, subordinating or preposition	*of, on, before, unless*
JJ	adjective	*nice, easy*
JJR	adjective, comparative	*nicer, easier*
JJS	adjective, superlative	*nicest, easiest*
LS	list item marker	
MD	verb, modal auxillary	*may, should*
NN	noun, singular or mass	*tiger, chair, laughter*
NNS	noun, plural	*tigers, chairs, insects*
NNP	noun, proper singular	*Germany, God, Alice*
NNPS	noun, proper plural	*we met two Christmases ago*
PDT	predeterminer	*both his children*
POS	possessive ending	*'s*
PRP	pronoun, personal	*me, you, it*
PRP$	pronoun, possessive	*my, your, our*
RB	adverb	*extremely, loudly, hard*
RBR	adverb, comparative	*better*
RBS	adverb, superlative	*best*
RP	adverb, particle	*about, off, up*
SYM	symbol	*%*
TO	infinitival to	*what to do?*
UH	interjection	*oh, oops, gosh*
VB	verb, base form	*think*
VBZ	verb, 3rd person singular present	*she thinks*
VBP	verb, non-3rd person singular present	*I think*
VBD	verb, past tense	*they thought*
VBN	verb, past participle	*a sunken ship*
VBG	verb, gerund or present participle	*thinking is fun*
WDT	*wh*-determiner	*which, whatever, whichever*
WP	*wh*-pronoun, personal	*what, who, whom*
WP$	*wh*-pronoun, possessive	*whose, whosever*
WRB	*wh*-adverb	*where, when*

Figure 5.1: *Penn TreeBank Part Of Speech tags*

rodent rodent NN 1

a a DT 0.333333 a NN 0.333333 a NNS 0.333333

little little JJ 0.752817 little RB 0.245775 little DT 0.000704225 little PRP 0.000704225

larger large JJR 1

than than IN 1

a a DT 0.333333 a NN 0.333333 a NNS 0.333333

mouse mouse NN 1

and and CC 1

with with IN 0.999158 with RP 0.000842381

a a DT 0.333333 a NN 0.333333 a NNS 0.333333

very very RB 0.942516 very JJ 0.0574837

short short JJ 0.988095 short VB 0.0085034 short NN 0.00170068 short VBP 0.00170068

tail tail NN 0.833333 tail VB 0.0833333 tail VBP 0.0833333

. . Fp 1

The text or sentence to be analysed is split into words and the analysis for each word is displayed in a separate line of the morphological analysis file. The first word of each line is the word as it appears in the original input text, also called form. Next is the lemma which is the canonical form of that word. Then we can see each POS tag assigned to the word followed by a number which represents the probability of it being the correct tag.

This morphological analysis is stored in another text file, and it will be used as entrance later for the next module, when generating the dependency analysis file.

Using this morphological analysis file as entrance, Freeling generates a dependency parsing where the syntactic structure of each sentence is analysed, displaying relations and dependencies between the words in the sentence. Figure 5.2 shows the dependency parsing tree obtained for the previous sentence using the morphological analysis as input.

```
claus/top/(is be VBZ -) [
  sn-chunk/ncsubj/(hamster hamster NN -) [
    DT/det/(A a DT -)
  ]
  sn-chunk/dobj/(a a NNS -) [
    DT/det/(a a DT -)
    attrib/ncmod/(furry furry JJ -)
    NN/ncmod/(rodent rodent NN -)
  ]
  attrib/ncmod/(larger large JJR -) [
    adv/ncmod/(little little RB -)
  ]
  sp-coor/modnomatch/(and and CC -) [
    sp-chunk/conj/(than than IN -) [
      sn-chunk/dobj/(mouse mouse NN -) [
        DT/det/(a a DT -)
      ]
    ]
    sp-chunk/conj/(with with IN -) [
      sn-chunk/dobj/(tail tail NN -) [
        DT/det/(a a DT -)
        attrib/ncmod/(short short JJ -) [
          adv/ncmod/(very very RB -)
        ]
      ]
    ]
  ]
  st-brk/ta/(. . Fp -)
]
```

Figure 5.2: *Dependency parsing for "hamster"*

The first line, marked with *top* represents the parent node of the tree. Text inside brackets represents children nodes of the previous node. For each word, the syntactic function that it plays in the sentence is shown, followed by its morphological analysis with the se-

lected OS tag assigned to it (the one that had the highest probability in the morphological analysis).

Once the dependency parsing has been stored in a text file, this information must be transformed to match the internal representation used by the system. The dependency trees obtained from Freeling are stored as lists of nodes, each representing a word, with references to children nodes. The information contained in each node is as follows:

- id: a number that represents the position of the word in the sentence
- func: the function carried out by the word (such as ncsubj for the subject or dobj for the object)
- form: word
- lemma: lemma or canonical form of the word
- pos: Part Of Speech tag
- head: identifier of the parent node
- parent: reference to parent node
- children: list of references to children nodes

The full text is internally represented as a list of nodes, where each of these nodes is the root word of a sentence. There is the same number of nodes as sentences in the text. Each of these nodes in turn stores a list of children nodes which contain the rest of the words from their sentence.

Once this information has been saved in the system, it is ready to be analysed for triplets which might be candidates to represent IsA relations or properties of the main entity (HasProperty relations).

5.3 Extracting information from natural language texts using triplets

The triplets used will represent definitions and properties, concepts that establish a unidirectional relation with certain other concepts. For this project it was decided that the relations that should be extracted from the text were IsA or HasProperty. Even though these two relations are different, they can both be used to define a concept. The system's parameters can very easily be adjusted to find any type of relation (or all of them) rather than simply IsA and HasProperty. This type of output will be easily computable by machines and can be used to establish new relations between concepts. This can be achieved, for instance, by connecting triplets in which the second concept is the same as the first concept of the other triplet.

Triplets will be stored in JSON files, inside a list, and will follow the structure *{"sourceConcept":"hamster", "relation":"IsA", "targetConcept":"animal"}*.

The first step towards extracting triplets from the trees obtained using FreeLing is to iterate through each sentence examining its main verb. The verb of the sentence is the root of its tree, so iterating through each sentence's main verb does not have a high computational cost. If the user is only interested in extracting triplets with certain relations, then the appropriate verb must be found for the sentence to be a good candidate.

For instance, if the user wants definitions of the main entity, it is possible that they are only interested in IsA and HasProperty types of relations. In this case it may suffice with discarding sentences which have any verb other than "to be". Sentences with any form of the verb "to be" are likely to define the subject of the sentence, and the rest can be discarded. Forms of the verb "to have" may also belong to sentences which are good candidates to represent properties of the main concept. In order to find out if the verb matches the one that the user is looking for, the *lemma* of the verb node must be analysed and compared with the specified ones. Sentences that make use of a form other than the present tense have been taken into consideration because texts referring to historic events or characters

may use the past tense.

Once the sentences with the chosen verbs are found, we must make sure that the subject of the sentence matches the main entity. To find the subject we must look for the node with *func* = *ncsubj*. The subject's children are explored recursively to find out whether it contains the main entity or not. If it does, then this sentence is a good candidate to represent a triplet for the topic.

The last part of the process consists in finding the target concept of the relation. For this purpose, the object of the sentence is analysed (finding the node with *func* = *dobj*). At this point there are three possible scenarios:

1. When the root of the object is a noun or an adjective (its *func* parameter equals "NN", "NNS", "NNP" or "NNPS" for nouns and "JJ", "JJR" and "JJS" for adjectives), it can be saved directly as the target concept of the triplet. If it is a noun then the relation is stored as IsA, and if it is an adjective then it will be stored as a HasProperty relation. For example in the sentence "A hamster is a rodent", the object is "a rodent", and the root of the object is "rodent". Since this is a noun, the resulting triplet would be *(hamster - IsA - rodent)*. On the other hand, with a sentence such as " A hamster is furry", the root of the object is "furry", which is an adjective, so the triplet obtained from this sentence would be *(hamster - HasProperty - furry)*.

```
claus/top/(is be VBZ -) [
  sn-chunk/ncsubj/(hamster hamster NN -) [
    DT/det/(A a DT -)
  ]
  sn-chunk/dobj/(rodent rodent NN -) [
    DT/det/(a a DT -)
  ]
  st-brk/ta/(. . Fp -)
]
```

Figure 5.3: *First example of first scenario of triplet extraction*

```
claus/top/(is be VBZ -) [
  sn-chunk/ncsubj/(hamster hamster NN -) [
    DT/det/(A a DT -)
  ]
  attrib/ncmod/(furry furry JJ -)
  st-brk/ta/(. . Fp -)
]
```

Figure 5.4: *Second example of first scenario of triplet extraction*

2. If the root of the object is a noun and has among its children any modifiers which are adjectives (*func* = *ncmod*), they are also selected as possible information related to the concept. For instance in the phrase: "A hamster is a furry rodent", the root of the object ("rodent") has one modifier, "furry", so aside from the triplet that represents an IS_A relation *(hamster - IsA - rodent)*, the adjective is stored in an additional triplet with the HasProperty relation *(hamster - HasProperty - furry)*.

```
claus/top/(is be VBZ -) [
  sn-chunk/ncsubj/(hamster hamster NN -) [
    DT/det/(A a DT -)
  ]
  sn-chunk/dobj/(rodent rodent NN -) [
    DT/det/(a a DT -)
    attrib/ncmod/(furry furry JJ -)
  ]
  st-brk/ta/(. . Fp -)
]
```

Figure 5.5: *Example of second scenario of triplet extraction*

3. If the root of the object is the conjunction "and" or "or" instead of a noun, its children are searched for nouns and adjectives much like in the previous case. For example, in the sentence "A hamster is a small animal and a rodent" the triplets extracted would be *(hamster - HasProperty - small)*, *(hamster - IsA - animal)* and *(hamster - IsA - rodent)*.

```
claus/top/(is be VBZ -) [
  sn-chunk/ncsubj/(hamster hamster NN -) [
    DT/det/(A a DT -)
  ]
  sn-coor/dobj/(and and CC -) [
    sn-chunk/conj/(animal animal NN -) [
      DT/det/(a a DT -)
      attrib/ncmod/(small small JJ -)
    ]
    sn-chunk/conj/(rodent rodent NN -) [
      DT/det/(a a DT -)
    ]
  ]
  st-brk/ta/(. . Fp -)
]
```

Figure 5.6: *Example of third scenario of triplet extraction*

When this method is applied to the example we are working with, the triplets obtained from Wikipedia, Simple Wikipedia and Simple Wiktionary are as shown in Figure 5.7

5.3.1 Difficulties encountered

The above method is relatively simple to understand and to implement, but it has a few disadvantages.

As mentioned before, the version of Freeling used in this project is not the latest one, and there as some issues in the morphological analysis that can be appreciated in Figure 5.7. For instance the triplet *hamster - HasProperty - a* does not make sense since "a" is neither a noun nor an adjective. However, Freeling 2.2 detected "a" as an adjective when analysing the sentence.

As explained in the previous section, when the object's root is a noun with an adjective that refers to it, both noun and adjective are stored separately in different triplets. But in some cases the concept's definition only makes sense when the adjective and noun are used together. For example, when defining a foot, the sentence "anatomical structure" was obtained. This makes sense as a combination, but a person would not usually describe a

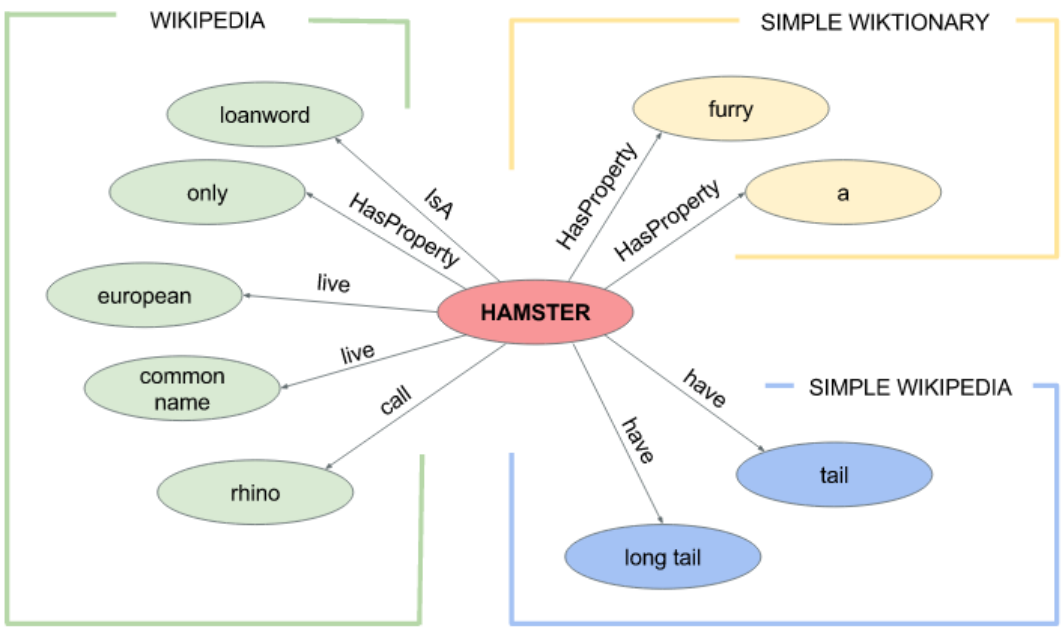

Figure 5.7: *Triplets extracted from the Wikis for "hamster"*

foot as a structure. When this situation arises, both words usually make sense separately as well as combined, but in some cases storing them separately renders one or both of them useless. The user may decide if they want to store the information in one longer triplet or in separate triplets by modifying the system's parameters.

Accepting any form of the verb "to be", including past tense, means that relevant information can be extracted from text regarding past events or historical characters. The problem is that this could also result in out of date information. For instance the sentence "In ancient times Germany was largely pagan" results in the triplet *(Germany - HasProperty - pagan)*. This is not true at present, and so this triplet is incorrect.

Sentences that use the Saxon genitive also tend to be problematic. The version of Freeling used in this project does not recognise it correctly, and assumes that the suffix -'s is an abbreviation of "is". This means that in a sentence such as "The star's radiation stops it from collapsing further under its own gravity" is transformed into "The star is radiation

stops it from collapsing further under its own gravity", and the resulting triplet is *(star - IsA - radiation)*.

An interesting phenomenon that occurs is when providing examples of a concept. Sentences such as "Examples of [concept] are..." or "A type of [concept] could be..." match the pattern recognised by the triplet extractor, so the sentence "A popular toy of this type is the Teddy Bear" will result in the triplet *(toy - IsA - teddy_bear)*. This represents information that is related to the concept, but since it does not match the IsA relation, it cannot be considered correct.

The use of free-content online resources poses a few obstacles in triplet extraction. When the concept name does not match an article name exactly, we are sometimes redirected to another article. This means that the sentences in this article might not use the exact concept name that was specified, and when trying to find the concept in the subject of a sentence, the triplet extractor will fail. For instance when searching for "clothes" in Wikipedia, the query is redirected to "clothing", and definitions such as "Clothing is an item or fabric which is made to cover part of the human body" are ignored because the main entity cannot be found.

Another problem presents itself in articles about people or characters. Sometimes they are referred to in different ways inside the article, for instance by their full name, just their first name, just their surname or even a nickname. When searching for "Bruce Willis" in the Wikipedia, he is referred to as "Walter Bruce Willis" and further ahead as just "Willis". In this case only the sentences that contain the concept written exactly as specified can be examined.

5.4 Text sources in common English versus Basic English

When extracting information from Natural Language text, the nature of the language used can greatly impact the quality of the information obtained. The three online text sources chosen for this step of the project were Wikipedia, Simple Wikipedia and Simple Wiktionary. The most commonly used of the three by the general public is Wikipedia, a rich and constantly growing source of articles written by users all over the world. The other two sources are written in Basic English.

5.4.1 Basic English

Basic English [Ogden, 1930] was created in 1930 by Charles K. Ogden. It is a controlled language, a simplified subset of English that can be used, for instance, to aid in teaching English to non-natives. It has a reduced vocabulary, with only 850 words which, according to Ogden, are sufficient for normal, everyday communication. The author suggests that an additional 100 words should be used specific to the field that the speaker works in, and another 50 words for a more specialised branch of the field. He also explains that around 200 English words have become international enough that most people, whatever language they speak, will understand them. This means that good communication should be achievable with a vocabulary of only 1200 words. Complex words can be broken into a combination of several simpler words and though texts written in this language will not have literary pretensions, they are easier to understand.

5.4.2 Wikipedia, Simple Wikipedia and Simple Wiktionary

Simple Wikipedia and Simple Wiktionary are two sources that make use of Basic English in their articles. Slightly less strict than Ogden on the amount of vocabulary allowed, they also ask users to write in active voice rather than passive, avoid contractions and use a simple sentence structure. The articles and topics described are no less complex, the language used is simply easier to understand.

Because of the simplified structure of the text and the clear vocabulary, these sources were good candidates to allow for better triplet extraction for this project. Simple Wikipedia, just like Wikipedia, is an encyclopedia-like source, offering articles which describe topics and concepts giving additional background information and going into detail. It contained 119,393 articles at the time that this document was written. Simple Wiktionary on the other hand is more similar to a dictionary, with 24,627 entries at the time that this was written. Both sources are smaller than the English Wikipedia, which had a total of 5,177,360 articles at the time this was written, but for a lot of common concepts, entries exist in two or more of the sources.

Below is an example of the type of language used in each of the Wikis:

- English Wikipedia:

  > Hamsters are rodents belonging to the subfamily Cricetinae. The subfamily contains about 25 species, classified in six or seven genera. They have become established as popular small house pets, and partly because they are easy to breed in captivity, hamsters are often used as laboratory animals.

- Simple Wikipedia:

  > Hamsters are rodents belonging to the subfamily Cricetinae. The subfamily contains about 25 species in six or seven genera. They have become established as popular small house pets. They are a bit like a mouse. Wild

hamsters live in the desert, but people all over the world keep domesticated hamsters as pets.

- Simple Wiktionary: "A hamster is a furry rodent a little larger than a mouse and with a very short tail. They are popular as pets."

Wikipedia is ranked as one of the top ten most popular websites at the time this article is written, so it provides a rich source of general reference information for this type of work. One of the main concerns when using a free-content resource is the quality of its content and language. Since the idea is not to extract very complex details of the concepts, the accuracy of these sources does not pose an impediment, because their general definitions tend to be correct. On the other hand, the structure of the text can be problematic when parsing the information. A simple grammatical error or an incorrectly structured sentence may lead to no triplets being extracted, or to triplets containing properties which are not definitions of the concept. This type of error is more likely to occur in sources where articles are longer and more complex.

Table 5.1 represents usage statistics for the chosen resources. Wikipedia has the highest number of articles and users, followed by Simple Wikipedia and finally Simple Wiktionary. It is to be expected that Wikipedia will produce triplets for a larger quantity of words than the other two sources. The data shows that Wikipedia users are less active, contributing less to the articles, whilst Simple Wikipedia and Simple Wiktionary users tend to contribute more. This may indicate that their community is more dedicated, in which case the quality of the articles seems more promising.

-	English Wikipedia	Simple Wikipedia	Simple Wiktionary
Articles	5,177,381	119,397	24,627
Users	28,495,359	526,306	16,507
Articles per user	0.18	0.23	1.49

Table 5.1: *Usage statistics of the used resources*

5.4.3 Evaluation of the extracted triplets

A selection of triplets extracted from Wikipedia, Simple Wikipedia and Simple Wiktionary were compared in order to find out whether information extracted from sources written in Basic English is more useful.

The evaluation criteria used to verify the quality of the extracted triplets is similar to the one used by Krawczyk [Krawczyk et al., 2015]. Every triplet generated for each concept is assigned a value based on how strongly related its property is to the concept and how well it respects the relation. The possible values are 1, 0.5 and 0.

- Triplets get the highest score when they correctly represent an IS_A or IS relation in which the property defines or is very strongly related to the concept. For instance the triplet *car* - *be* - *vehicle* would be considered a good triplet and it would be assigned 1 point.

- Mediocre triplets are assigned 0.5 points, when the property is a less accurate or informative definition of the concept, or when it represents a feature or quality of the concept. Note that the IS_A or IS relation must still be respected. A triplet such as *book* - *be* - *product* would have a score of 0.5 points.

- Triplets with properties which are related to the concept but do not respect the relation (for example *moon* - *be* - *crater*) or which are unrelated to the concept (*chocolate* - *be* - *iron*) are considered bad triplets and receive the lowest score (0).

The evaluation so far has been performed manually by four human annotators. The triplets generated for this evaluation were divided into four groups, where each annotator evaluated two groups and each triplet was evaluated by two annotators. The final statistics were obtained by using the average of the score given by all of the annotators, following an inter-annotator agreement using a popular metric, Fleiss Kappa [**?**]. This allows us to know the degree of agreement between the annotators.

A total of 62 concepts were chosen as input (e.g.: pinneaple, chocolate, Battle Royale...). The concepts used for testing belonged to different categories (animals, cities, celebrities, works of fiction, food, objects, abstract concepts, etc.) which were manually selected. The concepts were chosen without previously examining their articles in the text sources, but they were not generated randomly, they were chosen manually to ensure that they belonged to different categories. 49 of these concepts generated triplets for at least one of the knowledge sources. The absence of triplets for some concepts is due to texts with sentences defining the concept which do not match the required pattern accepted by the extractor. Both common nouns (water, yellow, chair...) and proper nouns (New York, Bruce Willis, Final Fantasy...) were used as input, and the latter produced less triplets (7 of the 13 concepts that did not generate any triplets were proper nouns). A total of 604 triplets were examined (428 from Wikipedia, 124 from Simple Wikipedia and 52 from Simple Wiktionary).

The results reflected in Table 5.2 show that sources with a large amount of content produce triplets for more concepts, as was expected. Consequently, Wikipedia is the source that offers the most good triplets (those assigned 1 point), followed by Simple Wikipedia and Simple Wiktionary. Note however that it also produces more mediocre triplets (0.5 points) and many more bad triplets (0 points) than the others. Even though the quantity of the triplets generated for sources using Basic English is compromised, their quality is much higher. Less than a third of the triplets extracted from Wikipedia can be considered good, and less than 10% are mediocre. This means that around 64% are bad triplets, representing

	Wikipedia	Simple Wikipedia	Simple Wiktionary
Concepts with triplets	46 (74.19%)	40 (64.52%)	26 (41.94%)
Triplets	428	124	52
Good triplets	119 (27.8%)	54.5 (43.95%)	28.5 (54.81%)
Mediocre triplets	36.5 (8.53%)	12.5 (10.08%)	9 (17.31%)
Bad triplets	272.5 (63.67%)	57 (45.97%)	14.5 (27.88%)
Average score	0.32	0.49	0.63
Inter-annotator agreement (kappa)	0.496	0.49	0.578

Table 5.2: *Results from the evaluation*

information that is not related to the specified concepts or that does not represent an IS_A or IS relation. Triplets extracted from Simple Wikipedia behave better, more than 40% of them are good, and less than half are bad.

As shown in Table 5.2, the degree of agreement between triplets extracted from Wikipedia and Simple Wikipedia is more or less the same. The Kappa score for Simple Wiktionary is better and shows that the annotators agree more on the quality of these triplets. Since the average score is higher for this source, this proves that triplets extracted from Simple Wiktionary have an overall better quality than the others.

The amount of concepts that generated triplets was similar for both Wikipedia and Simple Wikipedia, which means that the main difference between them was the content of the text. This proves that text expressed in Basic English yields more useful definitions for concepts than text written in common English.

Finally, the best results are achieved in Simple Wiktionary. Around 55% of the generated triplets are good definitions of the concepts, slightly less than 20% are mediocre, triplets which provide properties related to the concepts. Less than a third of the triplets are bad.

This seems to indicate that sources which contain less detailed and more specific content tend to result in higher quality triplets. Dictionaries are ideal, since they strive to define concepts briefly and do not offer additional background information. However, more research is needed to compare the results of knowledge extracted from encyclopedias against that extracted from dictionaries.

Figure 5.8 shows some of the scores awarded to triplets extracted from the three sources.

Apple	Wikipedia	a deciduous tree	0	Apple	SimpleWikipedia	a fleshy fruit	1	Apple	SimpleWiktionary	a sweet	0
Apple	Wikipedia	love	0	Apple	SimpleWikipedia	a very important species	0	Apple	SimpleWiktionary	a name	0
Banana	Wikipedia	an edible fruit	1	Banana	SimpleWikipedia	the common name	0	Book	SimpleWiktionary	a group of stamps	0
Book	Wikipedia	parchment	0	Book	SimpleWikipedia	a text	0,5	Card	SimpleWiktionary	often rectangular piece of paper or plastic	1
Book	Wikipedia	an e-book	0	Book	SimpleWikipedia	an industrial product	1	Cat	SimpleWiktionary	a short-form	0
Book	Wikipedia	an atlas	0	Cat	SimpleWikipedia	a queen	0	Cat	SimpleWiktionary	a short-form	0
Cat	Wikipedia	a symbol of good fortune	0,5	Cat	SimpleWikipedia	a short nickname	0	Cherry	SimpleWiktionary	a roundish small piece of fruit	1
Cat	Wikipedia	German_Katze	0	Cherry	SimpleWikipedia	a fruit	1	Cherry	SimpleWiktionary	the color of the fruit	1
Cat	Wikipedia	gastrointestinal tract	0	Desert	SimpleWikipedia	a dry biome	1	Cherry	SimpleWiktionary	a color	1
Cat	Wikipedia	meat-oriented physiology	0	Desert	SimpleWikipedia	the Sahara desert	0	Cherry	SimpleWiktionary	the flavour or smell	0
Cat	Wikipedia	tongue	0	Desert	SimpleWikipedia	the Sahara	0	Desert	SimpleWiktionary	a bare area of land	1
Cat	Wikipedia	a very vocal animal	0	Dessert	SimpleWikipedia	a type of food	1	Dessert	SimpleWiktionary	a sweet food dish	1
Cat	Wikipedia	ability	0	Grapefruit	SimpleWikipedia	a citrus	1	Dog	SimpleWiktionary	a domestic mammal	1
Cat	Wikipedia	a significant predator of birds	1	HarryPotter	SimpleWikipedia	the hero	0	Grapefruit	SimpleWiktionary	a fruit	1
Cherry	Wikipedia	the fruit of many plants of the genus Prunus	0,5	House	SimpleWikipedia	part	0	King	SimpleWiktionary	the male leader of a country whose son w	1
Desert	Wikipedia	a barren area of land where little precipitation	1	House	SimpleWikipedia	a building	1	Lemon	SimpleWiktionary	a yellow citrus fruit	1
Desert	Wikipedia	a region of land	1	King	SimpleWikipedia	a man who rules a count	1	Letter	SimpleWiktionary	a symbol which makes up part of a word	1
Dessert	Wikipedia	a course	1	Lemon	SimpleWikipedia	a small tree	1	Letter	SimpleWiktionary	a written message or note	1
Dessert	Wikipedia	Australia	0	Lemon	SimpleWikipedia	the common name	0	Lime	SimpleWiktionary	a green citrus fruit	1
Dog	Wikipedia	a	0	Lemon	SimpleWikipedia	a yellow citrus fruit	1	Lime	SimpleWiktionary	a material	1
Dog	Wikipedia	a Great_Dane	0	Lime	SimpleWikipedia	a fruit tree	1	Pear	SimpleWiktionary	an edible fruit	1
Dog	Wikipedia	senses	0	London	SimpleWikipedia	the largest city	0,5	Pineapple	SimpleWiktionary	a fruit	1
Dog	Wikipedia	limp ears	0	London	SimpleWikipedia	the biggest city	0,5	Strawberry	SimpleWiktionary	triangle-shaped fruit	1
Fun	Wikipedia	the enjoyment of pleasure	1	London	SimpleWikipedia	also_known_as Lunnain	0	Strawberry	SimpleWiktionary	a type	0
Fun	Wikipedia	an experience	1	London	SimpleWikipedia	a small city	0,5	Watermelon	SimpleWiktionary	a delicious fruit	1

Figure 5.8: *Some examples of scores for the triplets*

5.5 Information enrichment using ConceptNet

Since, as we saw in Section 4.2, ConceptNet offers triplets that have the main entity either in the source concept or the target concept, these triplets can also be included in the system.

Section 4.2 discussed how ConceptNet 5 was used to filter the information received as input. In this stage of the work, I also used ConceptNet for information enrichment.

Going back to the "hamster" example, in Figure 5.9 I showed how "mouse" was the only concept that appeared both in the spare triplets and in ConceptNet, and so the triplets from the input file connected to "mouse" were included in the system. There was one more step taken after that which serves to further enrich the information available.

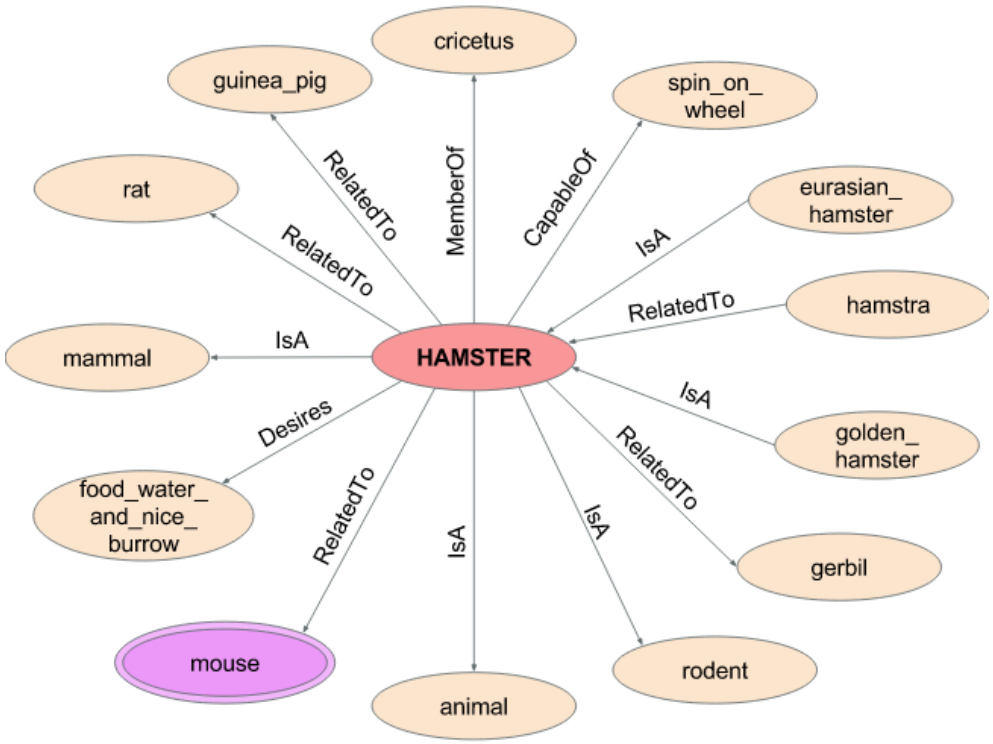

Figure 5.9: *Nodes and edges for "hamster" obtained from ConceptNet*

In order to fully justify the inclusion of the spare triplets connected to "mouse", the relation between that topic and the main entity can be explored further. According to ConceptNet, a hamster is related to a mouse. But in a text that is centered around information about hamsters, it may not be enough simply to say that hamsters and mice are related.

When this new concept is searched for in ConceptNet, some of the information retrieved for the topic is as shown in Figure 5.10.

If the data obtained is compared to the information available on "hamster", both from ConceptNet and from the input triplets, we can observe that the only properties that a mouse and a hamster have in common are that they are both an animal and they are both a rodent. This information can be included in the system so that the connection between mice and hamster is clearer. For instance one could say " A hamster is related to a mouse because

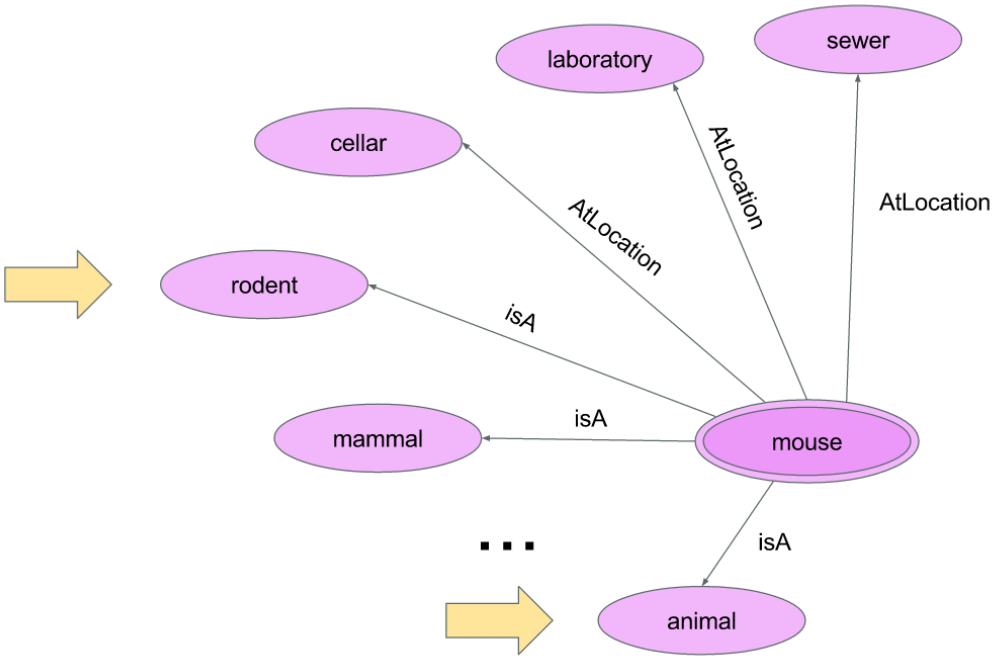

Figure 5.10: *Some nodes and edges for "mouse" obtained from ConceptNet*

they are both animals and they are both rodents". Figure 5.11 shows the new connections between "mouse" and "hamster" that will be included in the system.

It is also possible to enrich more triplets this way, for instance it can be done with all the concepts that have a "RelatedTo" relation with the main entity, since this relation is quite vague. In the current example, when performing this step we find that even though "hamster" and "hamstra" appear to hold a RelatedTo relation, they share no relation other than that one. For this reason the concept "hamstra" will be eliminated from the system, since no more connections can be found between it and the main entity.

Figure 5.12 shows the final triplets that were obtained from ConceptNet for the concept "hamster" and included in the system. Only 15 nodes were explored when examining the main entity in this example (two that appeared twice were discarded), but this number can be chosen by the user by modifying the system's parameters.

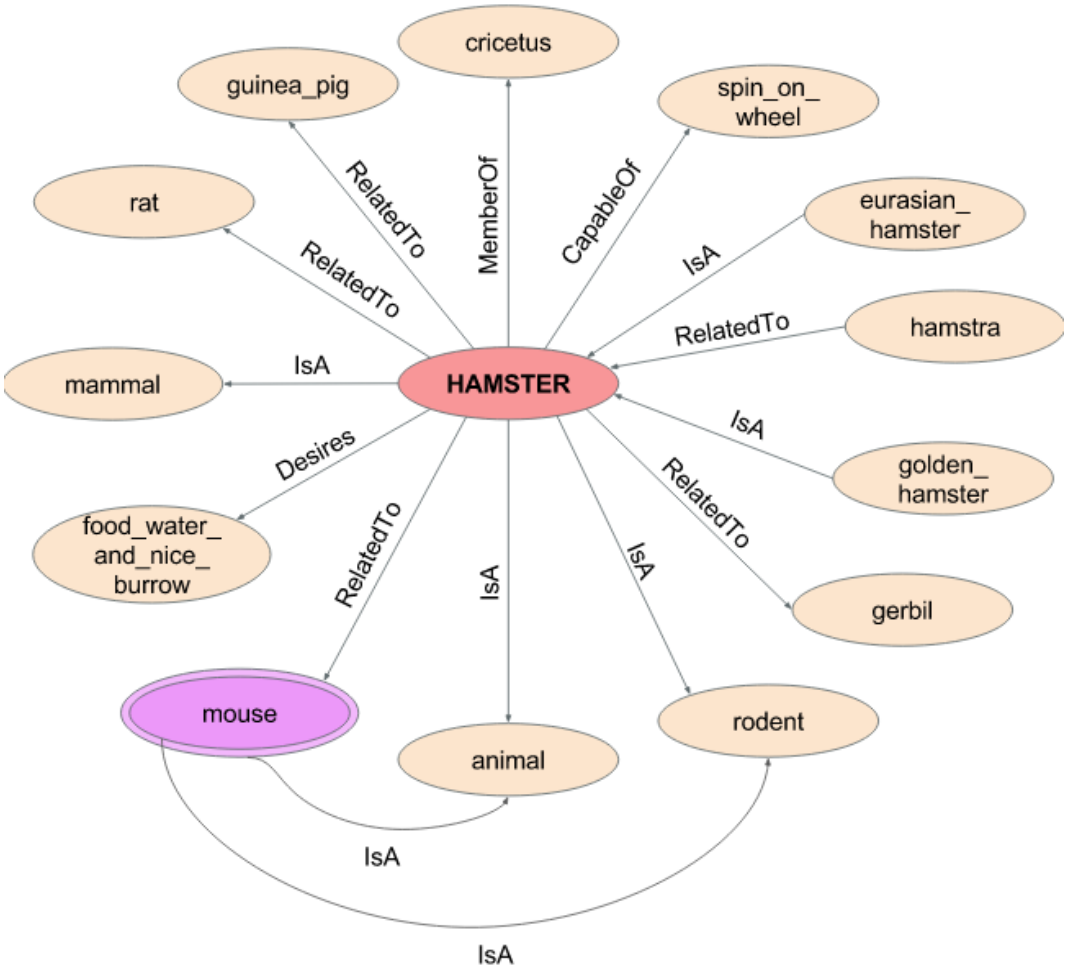

Figure 5.11: *Information obtained for "hamster" from ConceptNet*

5.6 Output triplets

As a last step towards the unification of the data, some relations from the input file will be slightly modified so that their names match the triplets extracted in the enrichment phase. The relation named "isa" will be transformed into "IsA", and the relation "property" now becomes "HasProperty". Relations from ConceptNet that are verbs will be stored in their infinitive form so that they match the triplets obtained from the Wikis. For instance the relation "Desires" will become "Desire". Underscores in triplets extracted from ConceptNet

will become blank spaces.

If the triplets obtained from ConceptNet, English Wikipedia, Simple Wikipedia and Simple Wiktionary are all introduced into the system, the final triplets for the concept "hamster" included in the system after the filtering phase and the enrichment phase are as shown in Figure 5.13.

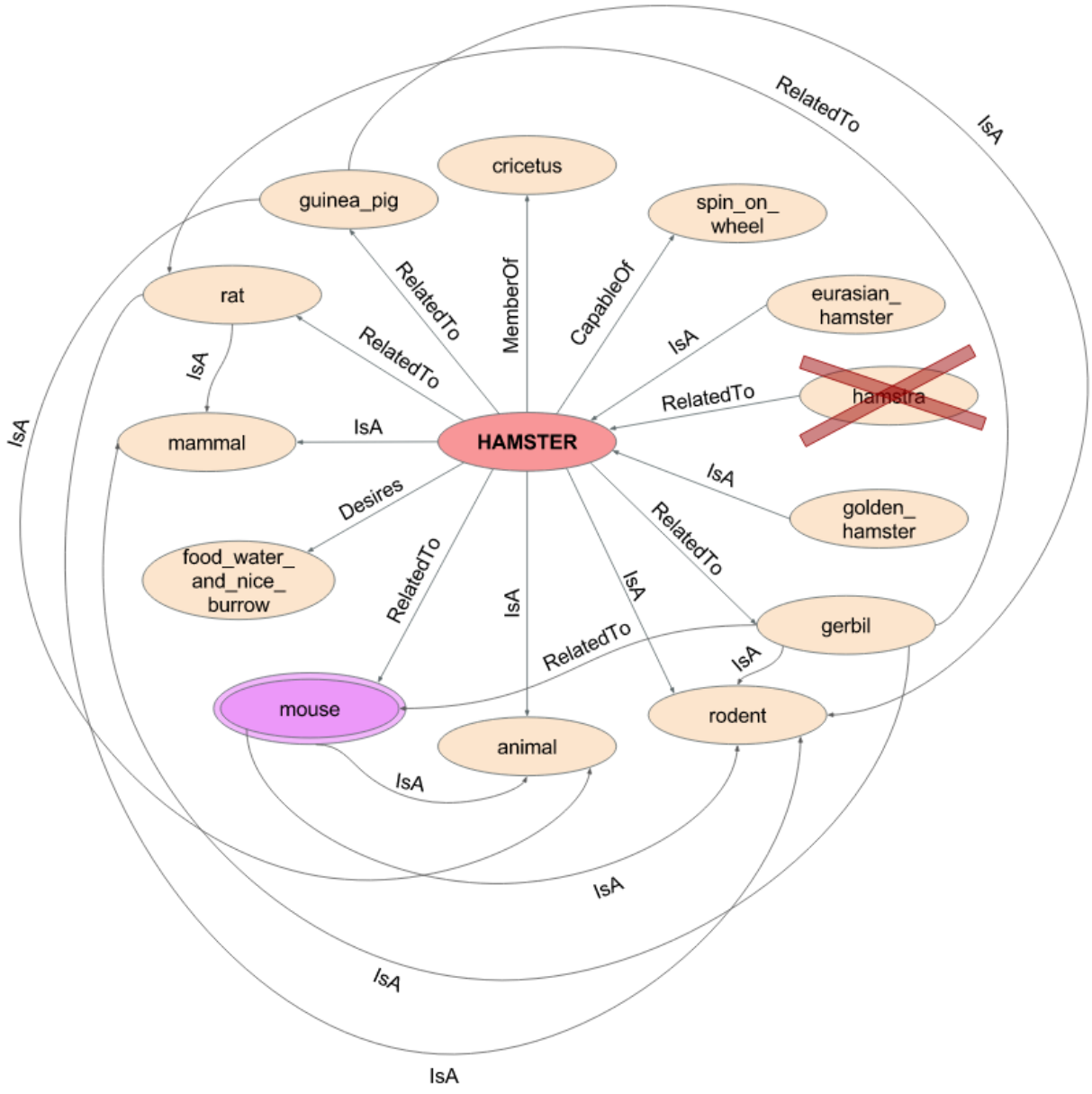

Figure 5.12: *Final triplets included for "hamster" from ConceptNet*

```
IsA(mouse, pet).
keep(people, pet).
RelatedTo(hamster, mouse).
IsA(hamster, crepuscular).
stay(hamster, day).
fee_on(hamster, seed).
HasProperty(tail, short).
IsA(tail, pouch).
HasProperty(pouch, long).
use(tail, pouch).
HasProperty(hamster, six).
IsA(hamster, type).
HasProperty(hamster, long).
IsA(hamster, tail).
HasProperty(tail, long).
have(hamster, tail).
may_look(hamster, slow).
be(hamster, fast).

IsA(hamster, loanword).
HasProperty(hamster, only).
live(hamster, european).
live(hamster, common name).
call(hamster, rhino).

have(hamster, tail).
have(hamster, long tail).

HasProperty(hamster, furry).
HasProperty(hamster, a).
```

```
MemberOf(hamster, cricetus).
CapableOf(hamster, spin on wheel).
IsA(eurasian hamster, hamster).
IsA(golden hamster, hamster).
RelatedTo(hamster, gerbil).
RelatedTo(gerbil, rat).
IsA(gerbil, mammal).
IsA(gerbil, rodent).
RelatedTo(gerbil, mouse).
IsA(hamster, rodent).
IsA(hamster, animal).
RelatedTo(hamster, mouse).
IsA(mouse, animal).
IsA(mouse, rodent).
Desire(hamster, food water and nice burrow).
IsA(hamster, mammal).
RelatedTo(hamster, rat).
IsA(rat, rodent).
IsA(rat, rodent).
RelatedTo(hamster, guinea pig).
IsA(guinea pig, rodent).
IsA(guinea pig, animal).
```

Figure 5.13: *Final triplets for "hamster" after filtering and enriching the information*

Chapter 6

Conclusions and Future Work

6.1 Conclusions

The task of Information Extraction for information enrichment is not an easy one. The quality of the text sources used plays an important role in the quality of the triplets extracted. Complex or incorrect sentence structures can lead to errors in morphological analysis and in dependency parsing, and to irrelevant information being introduced into the system. Spelling mistakes render the morphological analysis completely useless and can lead to a whole sentence being ignored. The quality of the text analysis tool is also extremely important.

Information filtering is also a difficult task. When two concepts appear to be related but the nature of the relation is not known, it may be hard to find and the concept could be mistakenly removed from the system.

The main goal of this project was to gather as much information as possible on a certain topic by exploring text sources written in natural language across the Internet, taking advantage of its potential to offer a huge amount of knowledge on any subject. The final results show that after filtering the input and enriching the information contained in it with the different online resources, the amount of knowledge on the specified topic more than doubles. If a user who is not part of ConCreTe were to use this application without any input triplets, they would be able to obtain a large amount of knowledge on their subject

of choice by combining the information extracted from the different text sources.

Considering the input triplets received, the information filtering phase was somewhat successful. The concepts which were completely irrelevant to the subject were eliminated. However, the concepts which were connected to the main entity and were included in the system are not necessarily 100% relevant to the topic. This aspect of the work could be improved.

The goals that were set for this project were the following:

- **Given a set of triplets that represents information regarding a certain topic, filter this information in order to discard as much incorrect or irrelevant information as possible, keeping only triplets related to the topic.**

  This part of the work was described in Chapter 4 of this document. The input triplets consisted of a main entity, a group of triplets that were connected to that main entity, and spare groups of triplets which had no apparent connection to the topic. The connected triplets were left untouched, and the spare triplets went through a process of filtering. By comparing them to an external knowledge base, ConceptNet 5, triplets that existed in both sources were considered relevant, and included in the system. The rest were discarded. After going through this phase of filtering, the resulting triplets are shown in Figure 6.1.

```
isa(contains, bit).
property(genus, seven).
isa(genus, bit).
isa(mouse, pet).
keep(people, pet).
isa(hamster, crepuscular).
stay(hamster, day).
fee_on(hamster, seed).
isa(vegetation, insect).
eat(vegetation, insect).
property(tail, short).
isa(tail, pouch).
property(pouch, long).
use(tail, pouch).
property(hamster, six).
isa(hamster, type).
property(hamster, long).
isa(hamster, tail).
property(tail, long).
have(hamster, tail).
property(toxic, dangerous).
be(bedding, dangerous).
eat(non, vegetable).
eat(toxic, vegetable).
eat(bedding, vegetable).
isa(syrian, aquarium).
shall_live_in(inch, aquarium).
shall_live_in(syrian, aquarium).
isa(cage, eye).
keep(when, eye).
may_look(hamster, slow).
be(hamster, fast).
```

```
IsA(mouse, pet).
keep(people, pet).
RelatedTo(hamster, mouse).
IsA(hamster, crepuscular).
stay(hamster, day).
fee_on(hamster, seed).
HasProperty(tail, short).
IsA(tail, pouch).
HasProperty(pouch, long).
use(tail, pouch).
HasProperty(hamster, six).
IsA(hamster, type).
HasProperty(hamster, long).
IsA(hamster, tail).
HasProperty(tail, long).
have(hamster, tail).
may_look(hamster, slow).
be(hamster, fast).
```

Figure 6.1: *Triplets before and after filtering*

- **Explore different online resources from which additional information could be extracted to enrich the existing triplets.**

  The main text sources used for information enrichment were Wikipedia, Simple Wikipedia, Simple Wiktionary and ConceptNet 5.

  The phase was described in Chapter 5 of this document. A comparison between the triplets extracted from the three wikis was covered in Section 5.4.2 and Section 5.4.3. Triplets extracted from Wikipedia, the source written in common English, were more abundant but proved to have less quality than those from sources written in Basic English. If the final triplets are aimed towards a task such as automatic abstracts generation, it may be useful to have less triplets knowing that their quality is higher. But this depends on what the system will be used for, so it should be decided by the user. Any combination of text sources can be used for information enrichment.

  The triplets extracted from ConceptNet are quite reliable and seem to provide useful information.

- **Build a tool that is able to extract triplets representing definitions or properties of a topic from any text written in Natural Language. These new triplets will be used to enrich the available information.**

  This part of the work was described in Chapter 5 of this document, specifically in Section 5.3. FreeLing was the text analysis tool used to analyse the text in order to extract triplets correctly. By performing a morphological analysis and dependency parsing, information related to the topic could be extracted from the texts and included in the system.

  ConceptNet was also used for this task.The spare triplets that were included in the information filtering phase were examined to find out what relations they had in common to the main entity, in order to strengthen their bond. Information extracted from ConceptNet was also enriched. Triplets connected to the main entity by “ReltaedTo”

relations were examined to find out if they could be connected to the subject in different ways.

The final triplets after going through filtering and enrichment are shown in Figure 6.2.

```
isa(contains, bit).
property(genus, seven).
isa(genus, bit).
isa(mouse, pet).
keep(people, pet).
isa(hamster, crepuscular).
stay(hamster, day).
fee_on(hamster, seed).
isa(vegetation, insect).
eat(vegetation, insect).
property(tail, short).
isa(tail, pouch).
property(pouch, long).
use(tail, pouch).
property(hamster, six).
isa(hamster, type).
property(hamster, long).
isa(hamster, tail).
property(tail, long).
have(hamster, tail).
property(toxic, dangerous).
be(bedding, dangerous).
eat(non, vegetable).
eat(toxic, vegetable).
eat(bedding, vegetable).
isa(syrian, aquarium).
shall_live_in(inch, aquarium).
shall_live_in(syrian, aquarium).
isa(cage, eye).
keep(when, eye).
may_look(hamster, slow).
be(hamster, fast).
```

```
IsA(mouse, pet).
keep(people, pet).
RelatedTo(hamster, mouse).
IsA(hamster, crepuscular).
stay(hamster, day).
fee_on(hamster, seed).
HasProperty(tail, short).
IsA(tail, pouch).
HasProperty(pouch, long).
use(tail, pouch).
HasProperty(hamster, six).
IsA(hamster, type).
HasProperty(hamster, long).
IsA(hamster, tail).
HasProperty(tail, long).
have(hamster, tail).
may_look(hamster, slow).
be(hamster, fast).

IsA(hamster, loanword).
HasProperty(hamster, only).
live(hamster, european).
live(hamster, common name).
call(hamster, rhino).

have(hamster, tail).
have(hamster, long tail).

HasProperty(hamster, furry).
HasProperty(hamster, a).
```

```
MemberOf(hamster, cricetus).
CapableOf(hamster, spin on wheel).
IsA(eurasian hamster, hamster).
IsA(golden hamster, hamster).
RelatedTo(hamster, gerbil).
RelatedTo(gerbil, rat).
IsA(gerbil, mammal).
IsA(gerbil, rodent).
RelatedTo(gerbil, mouse).
IsA(hamster, rodent).
IsA(hamster, animal).
RelatedTo(hamster, mouse).
IsA(mouse, animal).
IsA(mouse, rodent).
Desire(hamster, food water and nice burrow).
IsA(hamster, mammal).
RelatedTo(hamster, rat).
IsA(rat, rodent).
IsA(rat, rodent).
RelatedTo(hamster, guinea pig).
IsA(guinea pig, rodent).
IsA(guinea pig, animal).
```

Figure 6.2: *Input triplets versus output triplets*

6.2 Future Work

This system has granted satisfactory results but it can be greatly improved.

- The first problem that should be solved is the version of FreeLing being used. The text analysis tool should be updated to its latest version in order to avoid errors during the analysis.

- The issue of storing nouns and adjectives that refer to them separately or together should be studied further. Storing them together will make it harder to search for them in other sources in order to further enrich the information, but storing them separately can lead to incorrect or hard to understand triplets.

- It would be interesting to carry out a study comparing triplets extracted from encyclopedias to triplets extracted from dictionaries, in common English as well as Basic English.

- Another possible improvement could be programming the system to recognise patterns in natural language text rather than isolated verbs. This might improve information extraction from the sources and more complex sentences could be examined.

- Due to the nature of the text sources used, if the concept that the user is searching for does not match the exact name of the article, they might be redirected to another page where the topic is referred to with another name. In that case no triplets would be extracted. This should be corrected by storing the name that the site redirects the user to and using it as an alternative topic name.

- There is also a problem of different names being used throughout an article to refer to the main entity, especially when the topic is a person. The main entity can sometimes be referred to with different names throughout the text, but only the name that the user introduced would be used. There are sources such as DBpedia that can provide

pseudonyms and alternative names for people. This could be used in order to not lose this type of information.

- Synonyms could also be used to improve the system's behavior. If specific verbs are being searched for in the text, including synonyms for these verbs in the search will probably result in more information being extracted. Synonyms could also be used to ensure that concepts with the same meaning are not stored separately.

- Antonyms could be used for information filtering. When the main entity is connected by the same relation to two concepts which are antonyms, one of them is bound to be incorrect. An example of this can be observed in the input triplets from the University of Coimbra, where the triplet *(tail - property - long)* appears alongside the triplet *(tail - property - short)*.

Bibliography

[Akbik and Bross, 2009] Akbik, A. and Bross, J. (2009). Wanderlust: Extracting semantic relations from natural language text using dependency grammar patterns. In *Proceedings of the 2009 Semantic Search Workshop at the 18th International World Wide Web Conference*, pages 6–15, Madrid, Spain.

[Carreras et al., 2004] Carreras, X., Chao, I., Padró, L., and Padró, M. (2004). Freeling: An open-source suite of language analyzers. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*.

[Cunningham et al., 2011] Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M. A., Saggion, H., Petrak, J., Li, Y., and Peters, W. (2011). *Text Processing with GATE (Version 6)*.

[Dale and Haddock, 1991] Dale, R. and Haddock, N. (1991). Content determination in the generation of referring expressions. *Computational Intelligence*, 7(4):252–265.

[Dalvi et al., 2013] Dalvi, B. B., Cohen, W. W., and Callan, J. (2013). Websets: Extracting sets of entities from the web using unsupervised information extraction. *CoRR*, abs/1307.0261.

[Delgado et al., 1998] Delgado, J., Naohiro, I., and Tomoki, U. (1998). Content-based collaborative information filtering: Actively learning to classify and recommend documents. In *International Workshop on Cooperative Information Agents*, pages 206–215. Springer.

[Etzioni et al., 2008] Etzioni, O., Banko, M., Soderland, S., and Weld, D. S. (2008). Open information extraction from the web. *Commun. ACM*, 51(12):68–74.

[Hanani et al., 2001] Hanani, U., Shapira, B., and Shoval, P. (2001). Information filtering: Overview of issues, research and systems. *User Modeling and User-Adapted Interaction*, 11(3):203–259.

[Hearst, 1992] Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th Conference on Computational Linguistics - Volume 2*, COLING '92, pages 539–545, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Krawczyk et al., 2015] Krawczyk, M., Rzepka, R., and Araki, K. (2015). Extracting conceptnet knowledge triplets from japanese wikipedia. In *21st Annual Meeting of The Association for Natural Language Processing (NLP-2015)*, pages 1052–1055, Kyoto, Japan.

[Mititelu, 2008] Mititelu, V. B. (2008). Hyponymy patterns. In *International Conference on Text, Speech and Dialogue*, pages 37–44. Springer.

[Mooney, 1999] Mooney, R. (1999). Relational learning of pattern-match rules for information extraction. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, volume 334.

[Nivre et al., 2006] Nivre, J., Hall, J., and Nilsson, J. (2006). Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of LREC*, volume 6, pages 2216–2219.

[Ogden, 1930] Ogden, C. K. (1930). *Basic English: A General Introduction with Rules and Grammar*. Paul Treber & Co., Ltd, London.

[Reiter et al., 2000] Reiter, E., Dale, R., and Feng, Z. (2000). *Building natural language generation systems*, volume 33. MIT Press.

[Rusu et al., 2007] Rusu, D., Dali, L., Fortuna, B., Grobelnik, M., and Mladenic, D. (2007). Triplet extraction from sentences. In *Proceedings of the 10th International Multiconference Information Society*, pages 8–12.

[Sheth and Maes, 1993] Sheth, B. and Maes, P. (1993). Evolving agents for personalized information filtering. In *Artificial Intelligence for Applications, 1993. Proceedings., Ninth Conference on*, pages 345–352. IEEE.

[Shinyama and Sekine, 2006] Shinyama, Y. and Sekine, S. (2006). Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 304–311. Association for Computational Linguistics.

[Shinzato and Torisawa, 2004] Shinzato, K. and Torisawa, K. (2004). Acquiring hyponymy relations from web documents. In *HLT-NAACL*, pages 73–80.

[Sripada et al., 2001] Sripada, S. G., Reiter, E., Hunter, J., and Yu, J. (2001). A two-stage model for content determination. In *Proceedings of the 8th European Workshop on Natural Language Generation - Volume 8*, EWNLG '01, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Sumida and Torisawa, 2008] Sumida, A. and Torisawa, K. (2008). Hacking wikipedia for hyponymy relation acquisition. In *IJCNLP*, volume 8, pages 883–888. Citeseer.

[Weld et al., 2009] Weld, D. S., Hoffmann, R., and Wu, F. (2009). Using wikipedia to bootstrap open information extraction. *SIGMOD Rec.*, 37(4):62–68.

[Wu and Weld, 2010] Wu, F. and Weld, D. S. (2010). Open information extraction using wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 118–127, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Yates et al., 2007] Yates, A., Cafarella, M., Banko, M., Etzioni, O., Broadhead, M., and Soderland, S. (2007). Textrunner: Open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, NAACL-Demonstrations '07, pages 25–26, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Yu et al., 2002] Yu, K., Schwaighofer, A., and Tresp, V. (2002). Collaborative ensemble learning: Combining collaborative and content-based information filtering via hierarchical bayes. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pages 616–623. Morgan Kaufmann Publishers Inc.

Appendix A

Published paper for LREC 2016: Improving Information Extraction from Wikipedia Texts using Basic English

Improving Information Extraction from Wikipedia Texts using Basic English

Teresa Rodríguez-Ferreira, Adrián Rabadán, Raquel Hervás, Alberto Díaz
Facultad de Informática
Universidad Complutense de Madrid
teresaro@ucm.es, arabadan@ucm.es, raquelhb@fdi.ucm.es, albertodiaz@fdi.ucm.es

Abstract

The aim of this paper is to study the effect that the use of Basic English versus common English has on information extraction from online resources. The amount of online information available to the public grows exponentially, and is potentially an excellent resource for information extraction. The problem is that this information often comes in an unstructured format, such as plain text. In order to retrieve knowledge from this type of text, it must first be analysed to find the relevant details, and the nature of the language used can greatly impact the quality of the extracted information. In this paper, we compare triplets that represent definitions or properties of concepts obtained from three online collaborative resources (English Wikipedia, Simple English Wikipedia and Simple English Wiktionary) and study the differences in the results when Basic English is used instead of common English. The results show that resources written in Basic English produce less quantity of triplets, but with higher quality.

Keywords: Information Extraction, Triplets, Basic English

1. Introduction

Although software applications could theoretically benefit from the huge amount of information in the Web, they usually face the problem of this information appearing in the form of unstructured data like plain text. The possibility of automatically extracting the knowledge underlying this plain text is therefore becoming increasingly important.

Information Extraction (IE) is the process of automatically extracting structured data from unstructured texts. There are different ways to represent data extracted from text, such as in the form of graphs or by using triplets in the form *(concept$_1$, verb, concept$_2$)* to express relations between concepts extracted from the text. Although there are many IE approaches, in this paper we are only interested in unsupervised techniques that are able to extract information from plain text. For this kind of technique, the characteristics of the source text from which the information is going to be extracted play an important role in the obtained results.

In this paper we will evaluate whether the use of Basic English instead of common English leads to the extraction of more accurate data by implementing an experiment that compares triplets extracted from the English Wikipedia[1], Simple English Wikipedia[2] and Simple English Wiktionary[3] (from now on referred to as Simple Wikipedia and Simple Wiktionary). Basic English is a simplification of the English Language created by Ogden (1930) which defends that full communication can be achieved by using only 850 English words. In addition to using Basic English, Simple Wikipedia and Simple Wiktionary also ask users to write in shorter sentences, use active voice over passive voice and provide guidelines to help users write sentences with simple structures.

The triplets used will represent definitions and properties, concepts that establish a unidirectional IS_A or IS relation with certain other concepts. Even though these two relations are different, they can both be used to define a concept, so they have not been considered separately in the final results. This type of output will be easily computable by machines and can be used to establish new relations between concepts. This can be achieved, for instance, by connecting triplets in which the second concept is the same as the first concept of the other triplet.

The paper will address questions such as:

- Are triplets obtained from text written in Basic English more useful?
- How does information obtained from dictionaries compare to information obtained from encyclopedias?

The goal of this work is not to provide a new IE technique that improves previous work results, but to demonstrate that texts written using simplified vocabulary and grammar will lead to better triplet extraction.

In Section 2 we discuss previous work that is relevant to the field of Information Extraction. In Section 3 we describe the sources used and the results we expect to obtain from them, and we cover implementation details. In Section 4 we explain the evaluation criteria for the quality of the triplets obtained, we present the final results and we cover the issues encountered during this research. Section 5 is a discussion of the results. Finally, Section 6 describes future work that will improve the triplet extraction system.

2. Related work

Information Extraction (IE), the process of automatically extracting structured information from unstructured texts, has progressed substantially over the last few decades (Etzioni et al., 2008). Although the ambiguous nature of plain text makes the task an arduous one, it is possible to find many systems that have obtained quite good results. TextRunner (Yates et al., 2007), one of the pioneers in Open Information Extraction (OIE), is able to obtain high-quality information from text in a scalable and general manner.

[1] http://www.wikipedia.org
[2] http://simple.wikipedia.org
[3] http://simple.wiktionary.org

Rusu et al. (2007) present an approach to extracting triplets from sentences by relying on well known syntactic parsers for English.

Wikipedia is considered an excellent source of texts for IE systems due to its broad variety of topics and advantageous characteristics such as the quality of the texts and their internal structure. Therefore there are some IE systems that work with Wikipedia texts and/or their structured metadata, like Wanderlust (Akbik and Bross, 2009) or WOE (Wikipedia-based Open Extractor) (Wu and Weld, 2010). Weld et al. (2009) restrict their process to infoboxes, tabular summaries of an article's salient details which are included in a number of Wikipedia pages. Wanderlust (Akbik and Bross, 2009) is an algorithm that automatically extracts semantic relations from natural language text. The procedure uses deep linguistic patterns that are defined over the dependency grammar of sentences. Due to its linguistic nature, the method performs in an unsupervised fashion and is not restricted to any specific type of semantic relation. The applicability of the algorithm is tested using the English Wikipedia corpus. WOE (Wikipedia-based Open Extractor) (Wu and Weld, 2010) is a system capable of using knowledge extracted from a heuristic match between Wikipedia infoboxes and corresponding text. In particular, Krawczyk et al. (2015) present a method of acquiring new ConceptNet triplets automatically extracted from Japanese Wikipedia XML dump files. In order to check the validity of their method, they used human annotators to evaluate the quality of the obtained triplets.

3. Using Basic English for improving Information Extraction from texts

Our goal is to extract triplets which represent definitions or properties of a given concept established by a unidirectional IS_A or IS relation. Many other relations can be considered, but they are out of the scope of this experiment.

3.1. Textual knowledge sources

The sources where the triplets are extracted from must contain definitions and properties of concepts. The most appropriate resources for this purpose are dictionaries and encyclopedias. Dictionaries provide succinct definitions and a brief and usually more technical overview of the concept's most salient properties. Encyclopedias, on the other hand, contain more general information and in greater quantity.

We have chosen to use Wikipedia, Simple Wikipedia and Simple Wiktionary as sources for Information Extraction. All three are free-access and free-content collaborative Internet encyclopedias or dictionaries. This type of resource is fast-growing, with content created by users from all over the world (refer to Table 1).

Wikipedia is ranked as one of the top ten most popular websites at the time this article is written, so it provides a rich source of general reference information for this type of work. One of the main concerns when using a free-content resource is the quality of its content and language. Since we are not going to attempt to extract complex details of the concepts, the accuracy of these sources does not pose an impediment, because their general definitions tend to be correct. On the other hand, the structure of the text can be problematic when parsing the information. A simple grammatical error or an incorrectly structured sentence may lead to no triplets being extracted, or to triplets containing properties which are not definitions of the concept. This type of error is more likely to occur in sources where articles are longer and more complex.

Below is an example of a fragment of text extracted from the same article for each of the different sources:

1. Wikipedia: "Chocolate is a typically sweet, usually brown, food preparation of Theobroma cacao seeds, roasted and ground, often flavored, as with vanilla. It is made in the form of a liquid, paste, or in a block, or used as a flavoring ingredient in other foods."

2. Simple Wikipedia: "Chocolate is a food made from the seeds of a cacao tree. It is used in many desserts like pudding, cakes, candy, and ice cream. It can be a solid form like a candy bar or it can be in a liquid form like hot chocolate."

3. Simple Wiktionary: "Chocolate is a candy made from cacao beans and often used to flavour other foods such as cakes and cookies. A chocolate is an individual candy that is made of or covered in chocolate. Chocolate is a dark brown colour."

3.2. Triplet extraction

In order to extract relevant semantic information from the text, it must first go through a process of morphological analysis and dependency parsing. The analyser used was Freeling 2.2 (Carreras et al., 2004), an open source language analysis tool suite that supports several languages, including English.

The information for each specified concept was obtained from the corresponding web page from each source. For example, for the concept *pinneaple* and the source Simple Wikipedia the wiki page used was *https://simple.wikipedia.org/wiki/Pineapple*. This information was parsed into plain text, and then morphologically analysed using Freeling 2.2 (Carreras et al., 2004). This was in turn used as input for the dependency parsing, producing a final output of a tree containing all the semantic information. After this, the objective was to extract only IS_A or IS relations from the texts, so only sentences which had as their root any form of the verb "to be" were considered. Assertions that make use of a form other than the present tense were taken into consideration because texts referring to historic events or characters may use the past tense. Once the relevant sentences had been collected, the next step was to find the ones referring to the specified concept. Since the aim is to extract IS_A or IS relations, the third element of the triplets is always a definition or a property of the first element, so the triplets follow this structure: *concept - verb - property*.

In order to obtain definitions of the concept or related information from the text, the object of the chosen sentences has been studied. There are three possible scenarios depending on the root of the object (refer to Table 2):

1. When the root of the object is a noun, it is considered as a possible definition of the concept. For instance

-	English Wikipedia	Simple Wikipedia	Simple Wiktionary
Articles	4,977,081	115,138	24,309
Users	26,395,232	470,736	14,981
Articles per user	0.19	0.24	1.62

Table 1: Usage statistics of the used resources

in the sentence "A pineapple is a fruit", the object is "a fruit" and its root is "fruit", which is a noun, so it is saved in a triplet (pineapple - be - fruit). This represents an IS_A relation.

2. If the noun has any modifiers which are adjectives, they are also selected as possible information related to the concept. For instance in the phrase: "Chocolate is a dark brown colour", the root of the object ("colour") has two modifiers, "dark" and "brown", so aside from the triplet that represents an IS_A relation (chocolate - be - colour), both adjectives are stored in additional triplets (chocolate - be - dark, chocolate - be - brown). This type of information represents a property of the concept, an IS relation.

3. If the root of the object is the conjunction "and" or "or" instead of a noun, its children are searched for nouns and adjectives much like in the previous case, for example in the sentence "Battle Royale is a novel and a film" (Battle_Royale - be - novel, Battle_Royale - be - film). This represents an IS_A relation when the child is a noun or an IS relation when it is an adjective.

As an example, we can observe the differences between the properties extracted for the concept "wine":

- From Wikipedia, the extracted properties for the triplets were *cabernet_sauvignon*, *gamay*, *merlot*, *part*, *tradition* and *red*.
- From Simple Wikipedia, the properties were *drink*, *alcoholic* and *popular*.
- From Simple Wiktionary, only one property was extracted: *drink*.

4. Evaluation

The evaluation criteria used to verify the quality of the extracted triplets is similar to the one used by Krawczyk et al. (2015). Every triplet generated for each concept is assigned a value based on how strongly related its property is to the concept and how well it respects the relation. The possible values are 1, 0.5 and 0.

- Triplets get the highest score when they correctly represent an IS_A or IS relation in which the property defines or is very strongly related to the concept. For instance the triplet *car - be - vehicle* would be considered a good triplet and it would be assigned 1 point.
- Mediocre triplets are assigned 0.5 points, when the property is a less accurate or informative definition of the concept, or when it represents a feature or quality of the concept. Note that the IS_A or IS relation must still be respected. A triplet such as *book - be - product* would have a score of 0.5 points.
- Triplets with properties which are related to the concept but do not respect the relation (for example *moon - be - crater*) or which are unrelated to the concept (*chocolate - be - iron*) are considered bad triplets and receive the lowest score (0).

The evaluation so far has been performed manually by four human annotators. The triplets generated for this evaluation were divided into four groups, where each annotator evaluated two groups and each triplet was evaluated by two annotators. The final statistics were obtained by using the average of the score given by all of the annotators, following an inter-annotator agreement using a popular metric, Fleiss Kappa (Fleiss, 1981). This allows us to know the degree of agreement between the annotators.

4.1. Results

A total of 62 concepts were randomly chosen as input (e.g.: pinneaple, chocolate, Battle Royale...), 49 of which generated triplets for at least one of the knowledge sources. The absence of triplets for some concepts is due to texts with sentences defining the concept which do not match the required pattern accepted by the extractor. Both common nouns (water, yellow, chair...) and proper nouns (New York, Bruce Willis, Final Fantasy...) were used as input, and the latter produced less triplets (7 of the 13 concepts that did not generate any triplets were proper nouns). A total of 604 triplets were examined (428 from Wikipedia, 124 from Simple Wikipedia and 52 from Simple Wiktionary).
The results reflected in Table 3 show that sources with a large amount of content produce triplets for more concepts, as was expected. Consequently, Wikipedia is the source that offers the most good triplets (those assigned 1 point), followed by Simple Wikipedia and Simple Wiktionary. Note however that it also produces more mediocre triplets (0.5 points) and many more bad triplets (0 points) than the others. Even though the quantity of the triplets generated for sources using Basic English is compromised, their quality is much higher. Less than a third of the triplets extracted from Wikipedia can be considered good, and less than 10% are mediocre. This means that around 64% are bad triplets, representing information that is not related to the specified concepts or that does not represent an IS_A or IS relation. Triplets extracted from Simple Wikipedia behave better, more than 40% of them are good, and less than half are bad. As shown in Table 3, the degree of agreement between triplets extracted from Wikipedia and Simple Wikipedia is more or less the same. The Kappa score for Simple Wiktionary is better and shows that the annotators

Sentence	Freeling V2.2 tree	Triplets
A pineapple is a fruit	claus/top/(is be VBZ -) [n-chunk/ncsubj/(Pineapple pineapple NN -) sn-chunk/dobj/(fruit fruit NN -) [DT/det/(a a DT -)]]	Pineapple - be - fruit
Chocolate is a dark brown colour	claus/top/(is be VBZ -) [n-chunk/ncsubj/(Chocolate chocolate NN -) sn-chunk/dobj/(colour colour NN -) [DT/det/(a a DT -) attrib/ncmod/(dark dark JJ -) attrib/ncmod/(brown brown JJ -)]]	Chocolate - be - dark Chocolate - be - brown Chocolate - be - colour
Battle Royale is a novel and a film	claus/top/(is be VBZ -) [n-chunk/ncsubj/(Royale royale NNP -) [NN/ncmod/(Battle battle NN -)] sn-coor/dobj/(and and CC -) [sn-chunk/conj/(novel novel NN -) [DT/det/(a a DT -)] sn-chunk/conj/(film film NN -) [DT/det/(a a DT -)]]]	Battle_Royale - be - novel Battle_Royale - be - film

Table 2: Triplet extraction scenarios

agree more on the quality of these triplets. Since the average score is higher for this source, this proves that triplets extracted from Simple Wiktionary have an overall better quality than the others.

The amount of concepts that generated triplets was similar for both Wikipedia and Simple Wikipedia, which means that the main difference between them was the content of the text. This proves that text expressed in Basic English yields more useful definitions for concepts than text written in common English.

Finally, the best results are achieved in Simple Wiktionary. Around 55% of the generated triplets are good definitions of the concepts, slightly less than 20% are mediocre, and less than a third of the triplets are bad. This seems to indicate that sources which contain less detailed and more specific content tend to result in higher quality triplets. Dictionaries are ideal, since they strive to define concepts briefly and do not offer additional background information.

	Wikipedia	Simple Wikipedia	Simple Wiktionary
Concepts with triplets	46 (74.19%)	40 (64.52%)	26 (41.94%)
Triplets	428	124	52
Good triplets	119 (27.8%)	54.5 (43.95%)	28.5 (54.81%)
Mediocre triplets	36.5 (8.53%)	12.5 (10.08%)	9 (17.31%)
Bad triplets	272.5 (63.67%)	57 (45.97%)	14.5 (27.88%)
Average score	0.32	0.49	0.63
Inter-annotator agreement (kappa)	0.496	0.49	0.578

Table 3: Results from the evaluation

4.2. Detected errors in triplet extraction

The above method is relatively simple to understand and to implement, but it has a few disadvantages. When the text does not have any sentences that match the required pattern exactly, no triplets can be extracted. For instance, if a definition uses a verb other than "to be", but equivalent to it, the sentence will be ignored. The definition of "purple" extracted from the Wikipedia ("Purple is defined as a deep, rich shade between crimson and violet [...]") cannot be processed because "defined" is the main verb and "is" is an auxiliary verb. If the word "is" had been used by itself, the triplets *purple - be - shade, purple - be - deep* and *purple - be - rich* could have been extracted.

As explained above, when the object's root is a noun with an adjective that refers to it, both noun and adjective are

stored separately in different triplets. In some cases the concept's definition only makes sense when the adjective and noun are used together. For example, when defining a foot, the sentence "anatomical structure" was obtained. This makes sense as a combination, but a person would not usually describe a foot as a structure. When this situation arises, both words usually make sense separately as well as combined, but in some cases storing them separately renders one or both of them useless. The final decision was to keep the information separately in the triplets, ensuring that the results will be more easily computable, at the expense of having triplets which are more general and less precise.

Accepting any form of the verb "to be", including past tense, means that relevant information can be extracted from text regarding past events or historical characters. The problem is that this could also result in out of date information. For instance the sentence "In ancient times Germany was largely pagan" results in the triplet *Germany - be - pagan*. This is not true at present, and so this triplet is incorrect.

An interesting phenomenon that occurs is when providing examples of a concept. Sentences such as "Examples of [concept] are..." or "A type of [concept] could be..." match the pattern recognised by the triplet extractor, so the sentence "A popular toy of this type is the Teddy Bear" will result in the triplet *toy - be - teddy_bear*. This represents information that is related to the concept, but since it does not match the IS_A or IS relation, it cannot be considered correct.

Another problem presents itself in articles about people or characters. Sometimes they are referred to in different ways inside the article, for instance by their full name, just their first name, just their surname or even a nickname. When searching for "Bruce Willis" in the Wikipedia, he is referred to as "Walter Bruce Willis" and further ahead as just "Willis". In this case only the sentences that contain the concept written exactly as specified can be examined.

5. Conclusions

The results discussed in section 4.1. reveal that sources written in Basic English produce less quantity of triplets for a given concept than those written in English, but the triplets display much higher quality. Overall, the triplets extracted from Simple Wiktionary are twice as good as those extracted from Wikipedia. Generally, longer articles which tend to be more detailed and provide background information about the concept result in more incorrect triplets. This can be observed especially in articles concerning very general topics or articles on historical events and characters, for instance in the article regarding the Earth. For this reason, articles from Wikipedia, which are usually longer than those in Simple Wikipedia, produce more triplets per concept, but a large portion are incorrect. On the other hand, certain types of articles do not produce any triplets, especially articles regarding proper nouns (such as countries, cities, books, films, games or names of people). In our evaluation 15 concepts which are proper nouns were introduced, and roughly half of them (7) did not generate triplets for any of the sources.

The precise and succinct style of dictionaries seems more useful in the extraction of IS_A and IS relations between concepts and their properties. The triplets extracted from this type of source are also more easily evaluated by human annotators, since the information they contain is more objective. More research is needed, however, in order to correctly compare results extracted from encyclopedias against results extracted from dictionaries.

6. Future work

In this research, our goal was to compare triplets obtained from sources written in common English with those from sources written in Basic English. For this reason Wikipedia and Simple Wikipedia were the first two options to be considered. While analysing the results obtained, it seemed likely that Simple Wiktionary might be an even better source than Simple Wikipedia. This was on the grounds that aside from using Basic English and simpler sentence structure, its content is more precise and focuses solely on definitions, which was the goal of this study. We did not, however, evaluate results obtained from the English Wiktionary. It would be interesting to compare Simple Wiktionary against Wiktionary to examine the effect of IE from dictionaries written in Basic English, and to compare Wikipedia against Wiktionary to further observe the differences between data extracted from dictionaries and from encyclopedias. However, these resources could also be combined since the information contained in each one complements the others.

The extracted triplets follow a simple structure: *concept - verb - property*. In this work the verb used is always "to be", but this could be extended to also include relations such as HAS_A or RELATED_TO.

Having encountered the errors discussed in section 4.2., it would be useful to detect the patterns that lead to these errors and address them before saving the triplets. The matter of storing nouns and the adjectives that apply to them separately or together should also be explored further. When stored separately they lead to a larger amount of simpler triplets, but some information can be lost in the process, leaving either the noun or the adjective meaningless without its partner. Finally, the use of synonyms can aid in the recognition of additional triplets in the content. When searching for a concept, definitions that refer to it with a synonym (or a nickname or alternative name in the case of a person) are currently ignored. Using synonyms for common names, or alternative names found for people in sources such as DBpedia could produce richer results.

In order to reduce the time employed in the evaluation of the generated triplets, an automatic or semi-automatic criteria for evaluation should be implemented. By using existing triplets or relations similar to ours from sources such as ConceptNet, we could compare the results with others that are accepted as correct to automatically approve the common triplets.

7. Acknowledgements

This work is funded by ConCreTe. The project ConCreTe acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European

Commission, under FET grant number 611733.

8. References

Akbik, A. and Bross, J. (2009). Wanderlust: Extracting semantic relations from natural language text using dependency grammar patterns. In *Proceedings of the 2009 Semantic Search Workshop at the 18th International World Wide Web Conference*, pages 6–15, Madrid, Spain.

Carreras, X., Chao, I., Padró, L., and Padró, M. (2004). Freeling: An open-source suite of language analyzers. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*.

Etzioni, O., Banko, M., Soderland, S., and Weld, D. S. (2008). Open information extraction from the web. *Commun. ACM*, 51(12):68–74, December.

Fleiss, J. (1981). *Statistical methods for rates and proportions*. Wiley, New York, USA.

Krawczyk, M., Rzepka, R., and Araki, K. (2015). Extracting conceptnet knowledge triplets from japanese wikipedia. In *21st Annual Meeting of The Association for Natural Language Processing (NLP-2015)*, pages 1052–1055, Kyoto, Japan.

Ogden, C. K. (1930). *Basic English: A General Introduction with Rules and Grammar*. Paul Treber & Co., Ltd, London.

Rusu, D., Dali, L., Fortuna, B., Grobelnik, M., and Mladenic, D. (2007). Triplet extraction from sentences. In *Proceedings of the 10th International Multiconference Information Society*, pages 8–12.

Weld, D. S., Hoffmann, R., and Wu, F. (2009). Using wikipedia to bootstrap open information extraction. *SIGMOD Rec.*, 37(4):62–68, March.

Wu, F. and Weld, D. S. (2010). Open information extraction using wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 118–127, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yates, A., Cafarella, M., Banko, M., Etzioni, O., Broadhead, M., and Soderland, S. (2007). Textrunner: Open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, NAACL-Demonstrations '07, pages 25–26, Stroudsburg, PA, USA. Association for Computational Linguistics.

Appendix B

Configuration guide

This application was created by Teresa Rodríguez-Ferreira for the NIL research group as part of the CONCRETE project, Universidad Complutense de Madrid, Spain.

This analyser can extract triplets from online resources such as Wikipedia, Simple English Wikipedia, Wiktionary or Simple Wiktionary (English versions). You can also input text manually to submit it to morphological analysis and dependency parsing using FreeLing, a text analysis tool suite.

INSTRUCTIONS

In order to run this project on Windows you should:

1. Copy all the files into a folder in your computer

2. Modify at least the following files in iAnalyze/freeling-2.2/bin (this works in English and Spanish, if you need another language you will have to create these files for your language) tag-english.bat, tag-english1.bat, tag-english2.bat, tag-spanish.bat, tag-spanish1.bat, tag-spanish2.bat by replacing the existing paths with your own for example: C:/Users/user-0/Teresa/CONCRETE/iAnalyze/freeling-2.2/bin will become C/myFolder/iAnalyze/freeling-2.2/bin

3. Modify at least the following files in iAnalyze/freeling-2.2/data/config (this works in English and Spanish, if you need another language you will have to create these files for

your language) en.cfg, en1.cfg, en2.cfg, es.cfg, es1.cfg, es2.cfg by replacing the existing paths with your own for example: C/Users/user-0/Teresa/CONCRETE/iAnalyze/freeling-2.2/data will become C/myFolder/iAnalyze/freeling-2.2/data

4. Modify the configuration.properties file in iAnalyze/iAnalyzePRJ replace the existing paths with your own set the language you wish to use

Appendix C

Final application

The final application combines the different modules explained throughout this document and allows the user to choose which they want to activate depending on their needs.

If the user has input files in the form of triplets following the same format as the input obtained from Coimbra, they should include them in the *input* folder, and have each text file containing the triplets for one subject. The file name should be the name of this main subject. If the user desires to include input that has another format, they should configure the parsing module so that it can accept this new form of input and translate it into the system's internal format. If the user does not wish to include any input of their own, they can leave the *input* folder empty and the module that parses the input will not be activated. If this is the case, the module that filters the input triplets will also not be activated.

When using this tool, the user will be asked to chose whether they want to extract information from English Wikipedia, Simple English Wikipedia, Simple English Wiktionary, ConceptNet or all of them together. They will also be asked whether they wish to extract simple triplets (these contain only one word in the target concept) or complex triplets (may contain several words in the target concept). The *queries* folder may be consulted if the user wishes to see the triplets extracted for a concept from one particular text source.

The final triplets are contained in the *output* folder. Each file contains the final triplets (after filtering and enriching using the specified text sources) for each specified topic.