



FREIE UNIVERSITÄT BOZEN
LIBERA UNIVERSITÀ DI BOLZANO
FREE UNIVERSITY OF BOZEN · BOLZANO

Fakultät für
Informatik

Facoltà di Scienze
e Tecnologie informatiche

Faculty of
Computer Science



UNIVERSITÄT
DES
SAARLANDES

**Erasmus Mundus European Master in
Language & Communication Technology (LCT)**

Master of Science in Computer Science

Free University of Bozen-Bolzano

Master of Science in Language Science and Technology

Universität des Saarlandes

**Feasibility study on using Relational
Patterns in Entailment-based Question
Answering over Linked data.**

Thesis submission for a

Master of Science in Computer Science

Wanzare Lilian Diana Awuor

July 2011

Supervisor: Prof. Raffaella Bernardi

Co-Supervisor: Prof. Bernardo Magnini

Co-Supervisor: Prof. Manfred Pinkal

Declaration

I hereby confirm that the thesis presented here is my own work, with all assistance acknowledged.

Bozen-Bolzano, 7th July, 2011.

Wanzare Lilian Diana Awuor

Acknowledgment

The longest journey begins with a single step, and the peak of creativity is achieved in a team. Having come this far, I would like to first thank The Almighty Father for the gifts He gave me: the gift of life, time, His protection and the people who saw me through.

To my supervisor, Prof. Raffaella Bernardi, this thesis would not have been possible without you. Your guidance, dedication in reading my work and constant encouragements made it all possible. I would also like to thank my second supervisor, Prof. Bernardo Magnini, for the insights and feedback that streamlined my work. It has been an honor working with you. Not to forget my supervisor from Saarland, Prof. Manfred Pinkal, thank you for your time and feedback. The Saarland crew, Ms. Bobbye Pernice and PD Dr. Valia Kordoni, thank you for making my stay as smooth as possible. It was a pleasure meeting and working with all of you.

I am indebted to my colleagues whom we worked with in the project, Angelo Basile, Rahmed Mahendra and Silvana Bernaola. It was a pleasure working with you. To my new found friends, TzuYi Kuo, Nghia Pham, Diana Yagomba, Elisabeth Gatachew, Nick Ruiz, Gabriela Ferraro, Eberechi Innocenzia, Ario Santoso, Zakka Fauzan, Thi Ngoc Quynh, Ilyas Celik and Gang Tran Binh, thanks for making my stay a delight and for encouraging me to soldier on. Thanks Gabriela for reading and correcting my work. Thanks Elisabeth, you have been my sister all through.

I acknowledge the Erasmus Mundus program that gave me the opportunity to study and experience the awesome European culture. The time I have spent in Germany and Italy has been an experience of a lifetime. I also acknowledge Fondazione Bruno Kessler for giving me the chance to be part of their research group during my thesis preparation.

To the pillar of my life, my husband Alfred, you have always be there to support me, thank you. To my parents, brothers, sister and my extended family, it's a joy being part of you and knowing that you are always praying for me. To my Dad, Dr. Wanzare, thanks for reading my work and for pushing me to work harder.

Finally, I close by saying, I was once a caterpillar but now I'm a butterfly, thanks to all of you.

Dedication

This thesis is dedicated to my mother, Hellen Akinyi, nothing runs deeper than your love.

Contents

1	Introduction	1
1.1	Question Answering over linked Data	2
1.2	Objective and research questions	3
1.3	Structure of the thesis	4
2	Overview	5
2.1	Linked data	5
2.1.1	Wikipedia	6
2.1.2	DBpedia	7
2.2	Question Answering	12
2.2.1	Dimensions of QA task	12
2.2.2	Natural Language Interface to Database (NLIDB)	14
2.2.3	Entailment based QA	16
2.2.4	Related work on QA over Linked data	21
2.2.5	Challenges for QA over linked data	25
2.3	Relation extraction	27
2.3.1	Role of relational patterns in QA	28
2.3.2	Evaluating relation extraction systems	28
3	Feasibility study: Proposed work-flow	31
3.1	Problem statement	31
3.2	Proposed Work-flow	32
3.3	Building the gold-standard: Module 1	34
3.3.1	Annotation	34
3.3.2	Related annotation task	35
3.3.3	Building the corpus	36
3.3.4	Annotation scheme	36
3.3.5	Annotation guidelines	38
3.4	Inter-Annotator Agreement	38
3.4.1	Procedure	40
3.4.2	Challenges during annotation	41
3.4.3	Interpreting inter-annotator results	43
3.5	Building the gold-standard sentences	45
3.6	Chapter Summary	45

4	Relation extraction: Module 2	47
4.1	Extracting patterns from gold-standard sentences	47
4.2	Automatic extraction	48
4.2.1	Sentence extraction	48
4.2.2	Pattern extraction	49
4.3	Automatic extraction evaluation	51
4.3.1	Sentence extraction evaluation	51
4.3.2	Pattern extraction evaluation	52
4.4	Chapter Summary	52
5	Question Answering: Module 3	54
5.1	Acquiring user questions	54
5.2	Question-Pattern mapping	55
5.3	Question-Pattern mapping evaluation	56
5.4	Query generation and Answer retrieval	58
5.5	Chapter Summary	59
6	Conclusion and further research	60
6.1	Summary of work	60
6.2	Further research	62
A	Annotation guidelines	67
B	Codes for the developed modules	74
B.1	Description of each module	74
B.1.1	Inter-Annotator agreement	74
B.1.2	Pattern Extraction	74
B.1.3	Question-Pattern mapping module	74
B.1.4	Evaluation	75
B.2	Datasets developed	75

List of Figures

2.1	Redundancy of information both in the infobox and free text	7
2.2	Sample infobox showing relationship between domain, range and relation [http://en.Wikipedia.org/wiki/Men_in_Black_III]	8
2.3	Portion of DBpedia ontology for thing:work:film showing properties (relations), domains and ranges [http://mappings.dbpedia.org/server/ontology/classes/Film]	10
2.4	Sample DBpedia entity -MACS3 [http://dbpedia.org/page/MACS3]	11
2.5	From natural language to DB via ontologies [Lesmo and Robaldo, 2006]	16
2.6	TE-based Question Answering [Negri and Kouylekov, 2009]	17
2.7	Qall-me Framework [QALL-ME Consortium, 2009a]	19
2.8	PowerAqua flow diagram [Lopez <i>et al.</i> , 2010]	22
2.9	Freya work flow showing the validation of POC through the user interaction [Damljanovic <i>et al.</i> , 2011]	24
3.1	Feasibility study work-flow	33
3.2	Variations in Dice values across relations	42
3.3	Number of annotated pages vs number of gold-standard sentences and patterns .	43
5.1	Results showing percentage of correctly mapped questions per relation	57

List of Tables

2.1	Details of DBpedia ontology	9
2.2	Number of instances for sample classes in the ontology	9
2.3	Infobox datasets vs infobox ontology datasets (version 3.2)	11
2.4	Qall-me ontology	18
2.5	Qall-me evaluation results [Sacaleanu <i>et al.</i> , 2008]	21
2.6	QALD-1 challenge: results for DBpedia linked data source	22
2.7	Partial view of the EMTs for QT <person/org, plays, rock group nirvana> [Lopez <i>et al.</i> , 2009]	23
2.8	The TMT for OTs in ontologies that match the QTs [Lopez <i>et al.</i> , 2009]	23
3.1	Target relations showing aspects considered during selection	37
3.2	Average Dice similarity results	41
4.1	Number of pages, sentences and extracted relational patterns	47
4.2	Examples of patterns from gold-standard	48
4.3	Examples of automatically extracted pattern candidates	50
4.4	Number of training sentences and extracted patterns per relation	51
4.5	Evaluation results for automatic sentence extraction	52
4.6	Evaluation results for automatic pattern extraction	53
5.1	Questions correctly vs those incorrectly mapped	57
6.1	User questions with similar relational patterns	61

Abstract

Question Answering aims at allowing users to ask a computer arbitrary questions and get correct answers back quickly and concisely. The current QA trend is moving towards answering questions from linked data resources. The objective of this thesis is to evaluate relational patterns extracted from Wikipedia and to carry out a feasibility study on the use of these relational patterns, rather than those extracted from a corpus of questions, in entailment based QA over linked data. Entailment based QA uses Recognizing Textual Entailment (RTE) paradigm. RTE relies on relational patterns that represent the various ways in which a particular relation between entities could be expressed in surface form. Linked data is based on subject, predicate and object model and the relational patterns represent the various ways in which the predicate can be expressed.

To achieve this objective, Wikipedia is used as the source of free text from where the relational patterns are extracted and DBpedia is used as an example linked data resource. This is because DBpedia was built by extracting structured information from Wikipedia, especially from the infoboxes. 10 relations, from DBpedia ontology, are used in the experiment. To acquire sentences expressing the chosen relations, the domain and range values (from the infobox) were looked up in their corresponding Wikipedia articles and only those sentences with a mention of both the domain and range were extracted as the sentences expressing that particular relation. These sentences were annotated with tags for the unit of words expressing the target relation and tags for the domain and the range values. With this, a gold-standard of annotated sentences was created. The results of the inter-annotator agreement show a high level of agreement and consistency, with respect to the built annotation guidelines, between the annotators. The average results are 0.76 for sentence agreement, 0.85 for domain tag agreement, 0.82 for range tag agreement and 0.70 for relation tag agreement.

The gold-standard sentences are used to evaluate the automatically extracted sentences and relational patterns from Wikipedia. From the gold-standard sentences, gold-standard relational patterns were extracted i.e. each relational pattern is an ordered set of domain, range and relation tags found in an annotated sentence.

Sample uses questions expressing the target relations are also acquired from the Web. The gold-standard and auto-extracted relational patterns are used (separately) to evaluate the question-pattern mapping module. Smith-Waterman string similarity metric is used in the question-pattern mapping module to measure the similarity between a user question and the set of relational patterns so as to get the relational pattern that is most similar to the user question. The results of the pattern mapping are positive, with 79% of the questions being correctly mapped to the matching pattern when using the auto-extracted patterns and 90% of the questions being correctly mapped when using the gold-standard patterns. These results show that the idea of using relational patterns extracted from text in entailment based QA over linked data is worth pursuing further.

Keywords:

Linked data, Textual Entailment, Question Answering, Relational Patterns, Relation Extraction, Inter-annotator Agreement.

Chapter 1

Introduction

When one talks about Question Answering (QA) the key point is being able to ask a computer arbitrary questions and get correct answers back quickly and succinctly without the headache of knowing how or where the computer gets the answers: from some database, or documents or even from the web. The other point is that the question is asked in natural language, independent of whether the user query and the target source language are similar or not. Research on QA has evolved over the recent years from Natural language Interface to Databases (NLIDB), QA over text data, and now to the recent area of QA over Linked data.

Traditionally, data was stored in computers mostly in form of databases, and access to these databases was achieved via formal query languages like SQL. This meant that the users had to understand the underlying structure of the databases and the formal query languages if they were to benefit from the stored data. Natural language interfaces to databases were then introduced in order to map the user question posed in natural language onto the formal query language, allowing the user to express themselves more naturally. The mapping was challenging in that different words could be used to ask about the same information, and were to be mapped onto the same formal query. NLIDB was limited in that the natural language interfaces were only over databases and not free text, and the databases were closed domains.

With the advent of the Web and digital documents, research developed in QA over free text (unstructured data) with the help of information retrieval systems. This fostered open domain QA because of the vast amount of data that was freely available on the web. Also ontologies were introduced to help bridge the gap between the user question and the concepts used to represent the data in the databases. Ontologies were used to markup data and in expanding the user query with similar terms in order to improve the recall. Recently, there has been a paradigm shift in the way data is being published on the web. Data is being published in such a way that it is machine readable and its meaning is explicitly defined using the Linked data¹ paradigm.

Linked Data resources can be used to answer a number of sophisticated factual queries over a wide range of topics. Linked Data pertains to links between data and standards for connecting data on the web[Bizer *et al.*, 2009a]. This allows for published data to be interlinked and combined with many more new data sources because of the already agreed upon standards i.e.,

¹Linked Data is method of publishing structured data so that it can be interlinked and become more useful-Wikipedia

rules of connecting the data and identifiers for data. Linked data relies on Resource Description Framework (RDF)² triple model that connects two entities, a *subject* and *object*, with a *predicate*. The predicate represents the relation that holds between the subject and object (see Section 2.1). The idea is to offer user friendly interfaces for typical users to be able to gain from the linked data resources. A natural language interfaces to linked data would be of great advantage.

1.1 Question Answering over linked Data

QA over linked data is the current QA trend. Linked data idea is to allow machines to understand the meaning of the text that is published on the web. This means that data is published in such a way that it is machine readable, its meaning is explicitly defined, it is linked to other external datasets, and can in turn be linked to form external datasets [Bizer *et al.*, 2009a]. Traditionally, data on the Web was made available as raw formats like XML or markups like HTML. This was not expressive enough to enable individual entities described in the documents to be connected by typed links to related entities. Now, the Web has evolved to the extent that both documents and data are linked.

The presence of linked data seems promising in really achieving open domain QA. One main goal of QA has been to be able to answer questions about any topic. The Web was good but not enough, linked data could be a break through in achieving high levels of openness. One example of successful attempts of linking data on the Web is DBpedia³, which has extracted structured information from Wikipedia and this information has been made available on the web. It has made possible the asking of sophisticated queries on Wikipedia and linking of other datasets on the Web to Wikipedia. Currently DBpedia has 672 million pieces of information RDF triples which could be explored to answer queries over a wide range of topics. For a typical Web user, this poses a challenge, how are they able to exploit this vast amount of information without the huddles of understanding its structure? To foster research in QA over linked data area, workshops like QALD-1⁴ have been started to bring together different researches interested in QA over linked data.

An example of QA technique that could be exploited to provide a natural language interface to linked data is entailment based QA that uses Recognizing Textual Entailment (RTE)⁵ techniques. As noted in [QALL-ME Consortium, 2009a], RTE techniques can allow one to 'deal with the variability expressed within the questions through semantic inferences at the textual level' and thus can be applied to languages with fewer linguistic resources. [Dagan and Glickman, 2004] define Textual entailment as 'the relationship between a coherent Text T and a language expression, which is considered as a hypothesis H '. The idea is that variable texts can be used to express the same kind of information, and thus one piece of text can be entailed in another, i.e. the Text T entails the Hypotheses H . This means that the meaning of H can be fully derived from the meaning of T . Textual entailment operates at the lexico-syntactic level and not full

²http://en.Wikipedia.org/wiki/Resource_Description_Framework

³<http://dbpedia.org/About>

⁴<http://www.sc.cit-ec.uni-bielefeld.de/qald-1>

⁵http://www.aclweb.org/aclwiki/index.php?title=Recognizing_Textual_Entailment

semantic interpretation. In a Question Answering scenario, T is the user question posed to the system, and H is the *relational pattern* representing the various ways in which a particular relation between entities could be expressed in surface form. For instance, to express the “invent” relation in text, one could say $[DOMAIN]$ *invented the* $[RANGE]$, $[DOMAIN]$ *created the* $[RANGE]$, $[DOMAIN]$ *discovered* $[RANGE]$ *e.t.c.*, where the $[DOMAIN]$ is the subject of the relation and the $[RANGE]$ is the object. These patterns are the hypothesis in entailment based QA. A connection can be made to the Linked data model; relational patterns represent how the *predicates* are expressed in surface form and thus once entailment is established between a user question and some relational pattern, one could infer the relation being expressed in the user question and possibly retrieve the entity being asked for as an answer. An example of state-of-art entailment based QA system that this thesis uses as a reference is Qall-me⁶ that applies the entailment paradigm in querying over structured data. In Qall-me, the relational patterns used as the hypotheses were extracted from a corpus of questions that represent the various ways users could pose questions in the cinema domain.

In the thesis, DBpedia will be used as the example linked data resource that could be queried using a natural language interface. In order to overcome the gap between natural language question and the linked data resource, Wikipedia is used to acquire the relational patterns that represent the various ways that the relation can be expressed. Wikipedia has been chosen because it has both free text and structured data in form of infoboxes, and DBpedia was created from it by extracting such structured data to form a rich resource of RDF triples. Also, the DBpedia ontology was manually constructed from the frequently used infoboxes in the English Wikipedia pages. In particular this thesis will focus on evaluating and using relational patterns extracted from Wikipedia in entailment based QA over linked data.

1.2 Objective and research questions

On one hand, acquiring user questions expressing different relations takes a lot of effort and are hard to come by for all relations. On the other hand, there exists on the Web large amounts of text that can be freely used to extract relational patterns. The objective of this thesis is to evaluate relational patterns extracted from Wikipedia and to carry out a feasibility study on the use of these relational patterns, rather than those extracted from a corpus of questions, in entailment based QA to query over linked data.

In line with the objective, this thesis would seek to answer the research questions highlighted below:

1. Is entailment based QA paradigm appropriate for QA over linked data?
2. Are relational patterns extracted from text scalable to the many relations that exist?

To answer the research questions, DBpedia and Wikipedia are used as case studies, DBpedia as an example linked data resource and Wikipedia for extracting the relational patterns. Chapter 3 presents the details of the proposed work-flow of the feasibility study in an attempt to answer the research questions.

⁶<http://qallme.fbk.eu>

1.3 Structure of the thesis

The rest of the thesis is structured as follows:

Chapter 2 gives an overview on linked data, DBpedia and Wikipedia. The chapter highlights on QA, with a special emphasis on entailment based QA and Qall-me as a case study, and gives the related work on development of QA, including the state-of-art systems in QA over linked data. An overview of relation extraction and related work on evaluating relation extraction systems is also presented.

Chapter 3 states the research problem and the experimental modules proposed in an attempt to answer the research questions. It explains the first module of building the gold-standard and the inter-annotator results.

Chapter 4 presents the second module on automatic extraction of sentences and relational patterns from Wikipedia. It also presents the evaluation results of the automatic extraction against the gold-standard sentences.

Chapter 5 presents the third module on testing the feasibility of using the relational patterns in entailment based QA over Linked data.

This thesis concludes with a brief summary and future work in Chapter 6, and provides, in the Appendix, the annotation guidelines developed and a link to where the codes and datasets used in the experiments can be downloaded.

Chapter 2

Overview

This chapter gives an overview on Linked data, DBpedia and Wikipedia. It highlights on QA, with a special emphasis on entailment based QA and Qall-me as a case study, and gives the related work on development of QA over linked data. An overview of relation extraction and related work on evaluating relation extraction systems is also presented.

2.1 Linked data

Linked Data is about using the Web to connect related data that wasn't previously linked, or using the Web to lower the barriers to linking data currently linked using other methods. More specifically, Wikipedia defines Linked Data as "a term used to describe a recommended best practice for exposing, sharing, and connecting pieces of data, information, and knowledge on the Semantic web using URIs and RDF"¹.

As defined by [Bizer *et al.*, 2009a], Linked data refers to data published on the Web in such a way that it is machine-readable, its meaning is explicitly defined, it is linked to other external data sets and can in turn be linked from external data sets. Linked data relies on documents containing data in RDF format, Uniform Resource Identifies (URIs)² technology and Hypertext Transfer Protocol (HTTP)³ technology. It uses RDF to make typed statements that link arbitrary things in the world. URI is an entity that exists in the world and uses the *http://* schema. RDF model encodes data in the form of subject, predicate, object triples. Thus RDF links take the form of RDF triples where the *subject* of the triple is a URI reference in the namespace of one dataset, while the *object* of the triple is a URI reference in another [Bizer *et al.*, 2009a]. The *predicate* connects the subject and object in a relation, i.e. that the *relation:predicate* exists between the subject and the object. The examples below show examples of RDF links. Example 1 describes that the object resource is a member of the subject resource; the *relation:member* exists between data#DIG and card#i, while example 2 describes that the subject is the same as the object; the *relation:sameAs* exists between 77 and Pulp_Fiction_%28film%29.

¹<http://linked.data.org/>

²<http://labs.apache.org/webarch/uri/rfc/rfc3986.html>

³<http://www.w3.org/Protocols/rfc2616/rfc2616.html>

Example 1:

Subject: <http://dig.csail.mit.edu/data#DIG>
Predicate: <http://xmlns.com/foaf/0.1/member>
Object: <http://www.w3.org/People/Berners-Lee/card#i>

Example 2:

Subject: <http://data.linkedmdb.org/resource/film/77>
Predicate: <http://www.w3.org/2002/07/owl#sameAs>
Object: http://dbpedia.org/resource/Pulp_Fiction_%28film%29

DBpedia [Bizer *et al.*, 2009b] is an example of a project which was aimed at creating a linked data resource from Wikipedia. Next, Wikipedia and its structure that facilitated the building of DBpedia is presented, followed by more details on DBpedia itself.

2.1.1 Wikipedia

Wikipedia is a free encyclopedia that has been developed through community effort where anyone can create and edit articles on a variety of formats using pre-specified templates for different domains. Wikipedia is structured in form of free texts, links in the free text that direct to other Wikipedia pages or web pages in general, infoboxes⁴ that hold summaries of the main ideas in articles, tables and categories (pages used to group Wikipedia pages of similar topics together).

Wikipedia has been a common source for acquiring world knowledge in various language technology tasks, a concept referred to as Wikipedia mining⁵. Wikipedia mining involves extracting useful information, patterns, links e.t.c that can be used to aid Natural language processing (NLP) tasks. The existence of structured information in Wikipedia has made it possible for it to be mined for various kinds of lexical semantic information [Zesch *et al.*, 2007]. It is useful for getting a broad coverage of entities and their relations with other entities. One can mine for links to get how entities are linked together, categories and category hierarchies, named entities, relational patterns among entities e.t.c.

In a Wikipedia infobox, the *domain* is the principle entity that a given Wikipedia page is talking about (normally the title of the Wikipedia page), and the *range*, is the secondary entity or property that is in relation with the domain. For instance in Figure 2.2 the Domain is “*Movie : Men in Black 3*”, the relations are “*directed by*”, “*produced by*”, “*written by*” e.t.c and the range is of type “*Person*” for each of the above mentioned relations. An *entity* is the topic that a Wikipedia page is describing and is normally the domain and title of the Wikipedia page. A *mention* of the domain or range means that an instance of either the domain or range is found in the Wikipedia text.

As noted in the objective of the study, this thesis seeks to build a resource of relational patterns extracted from Wikipedia text and use the resource in entailment based QA over linked data. The Wikipedia text can be mined to get text phrases that talk about some relation in the

⁴<http://en.Wikipedia.org/wiki/Help:Infobox>

⁵http://Wikipedia-lab.org/en/index.php/Wikipedia_mining

infobox. The phrase contains a mention of the domain and the range. Such phrases are referred to as relational patterns. *Relational patterns* are the linguistic variations or surface realizations through which the underlying relation between the domain and the range is expressed. In Wikipedia, there are several pages describing different entities belonging to the same category. It can be argued that different writers would express the same relation differently and therefore one is able to acquire different phrases that express the same relation. Figure 2.1 shows the structure of Wikipedia where one gets redundancy of information between the infobox and the Wikipedia text. Figure 2.2 shows the relationship between the domain, range and relation in a sample Wikipedia infobox.

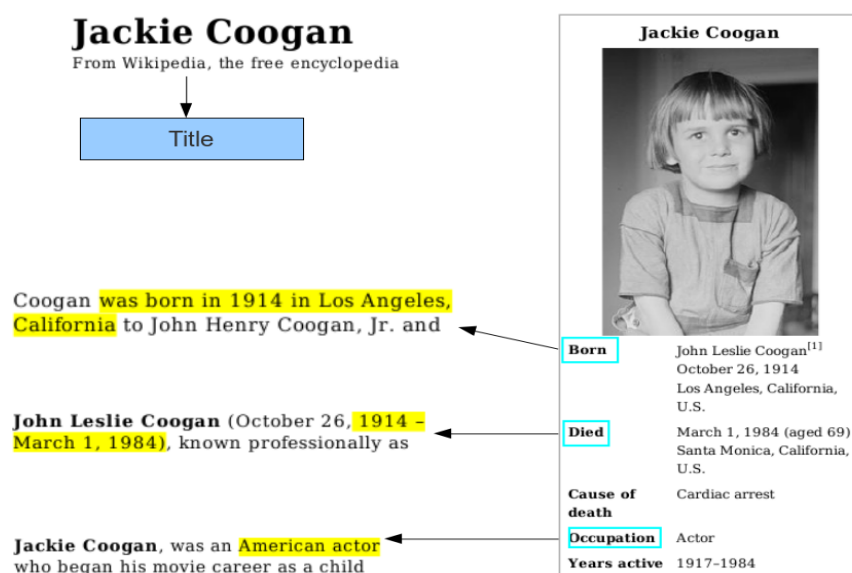


Figure 2.1 – Redundancy of information both in the infobox and free text

A further effort accomplished from Wikipedia is DBpedia. DBpedia is an extraction of structured information from Wikipedia articles and building a linked data resource from it. The structured information used to build DBpedia include infoboxes, links and tables. DBpedia is presented next.

2.1.2 DBpedia

Wikipedia has grown into one of the central knowledge sources of mankind and is maintained by thousands of contributors. Though it is mostly plain text, it has structured information in form of infoboxes, images, categorization information e.t.c. The DBpedia has been built by extracting data from such structured information, especially the infoboxes (see Figure 2.1). Infoboxes display an article's most relevant facts as a table of attribute-value pairs on the top right-hand side of the Wikipedia page. DBpedia dataset currently contains 3.5 million things (entities), 1.5 million of which have been described in a consistent ontology. The DBpedia dataset has several advantages over existing datasets: it covers many domains; it represents real community agreement; it automatically evolves as Wikipedia changes, and it is truly multilingual. A detailed description of the process of building DBpedia is found in [Bizer *et al.*, 2009b].

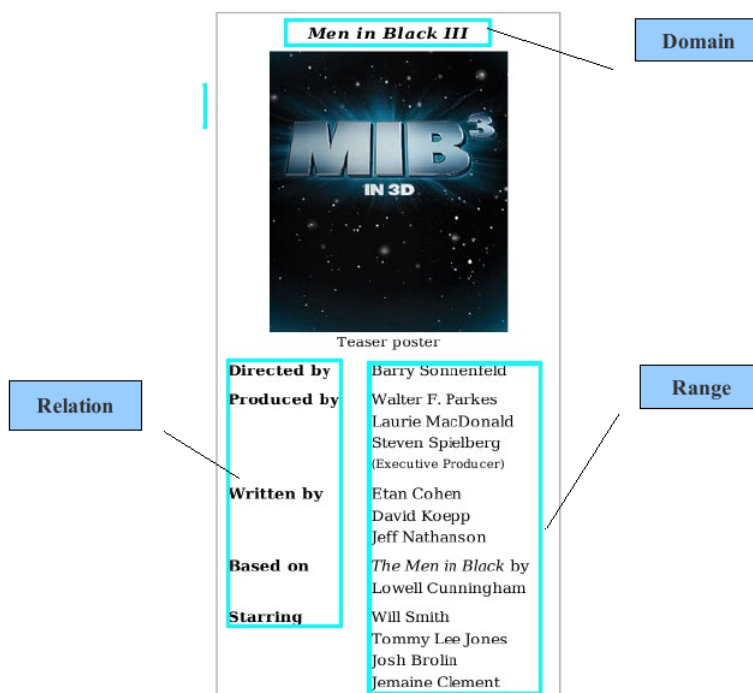


Figure 2.2 – Sample infobox showing relationship between domain, range and relation
[http://en.Wikipedia.org/wiki/Men_in_Black_III]

DBpedia is an example of a project which was aimed at creating a linked data resource from Wikipedia. For each of the entities described in DBpedia, a globally unique identifier (URI) is defined that can be dereferenced over the Web into a rich RDF description of the entity, relationships to other resources, classifications in concept hierarchies and various facts, as well as data-level links to other Web data sources describing the entity. Each resource is tied directly to an English language Wikipedia page. Over the last years, an increasing number of data publishers have begun to set data-level links to DBpedia datasets, making DBpedia a central interlinking hub for the emerging web of data. Currently, the web of interlinked data sources around DBpedia provides approximately 4.7 billion pieces of information and covers domains such as geographic information, people, companies, films, music, genes, drugs, books, and scientific publications [Bizer *et al.*, 2009b]. DBpedia dataset thus gives a promising resource to open domain Question Answering.

2.1.2.1 DBpedia ontology

DBpedia ontology is manually created based on the most commonly used infoboxes in Wikipedia. The ontology currently covers over 272 classes which form a subsumption hierarchy and are described by 1,300 different properties (relations). The DBpedia ontology is based on OWL⁶ and forms the structural backbone of DBpedia. Table 2.1 gives the number of classes, properties, datatypes and instances of the ontology. Examples of classes are person, city, country *e.t.c.*, properties are birth place, longitude *e.t.c.*, datatypes are string, integer *e.t.c.* and instances are

⁶OWL is a web Ontology Language designed for use by applications that need to process the content of information instead of just presenting it - W3C

existing things e.g. Italy, George W. Bush e.t.c. Table 2.2 gives a brief statistics of the number of instances for a sample of classes in the ontology.

	DBpedia Ontology
Classes	272
Properties	1300
Datatypes	55
Instances	1,667,000

Table 2.1 – Details of DBpedia ontology

Class	Instances
Resource (overall)	1,667,000
Place	462,000
Person	364,000
Work	355,000
Species	168,000
Organization	148,000

Table 2.2 – Number of instances for sample classes in the ontology

Compared to other ontologies which usually are created by relatively small groups of knowledge engineers, only cover specific domains and are very cost intensive to keep up-to-date as domains change, DBpedia has the advantage that it covers many domains and contains lots of instances; it represents real community agreement; and it (automatically) evolves as Wikipedia changes. The disadvantages of DBpedia ontology compared to hand-crafted ontologies are that DBpedia is less formally structured; the data quality is lower and there might be inconsistencies. To combine the advantages of both worlds DBpedia has been mapped to hand-crafted ontologies such as OpenCyc, SUMO, which enables applications to use the formal knowledge from these ontologies together with the instance data from DBpedia. DBpedia already contains 42,000 RDF links into OpenCyc and 2.4 million RDF links to Freebase. Figure 2.3 shows a section of the DBpedia ontology depicting relations or properties for Film class, with the domain and range for each relation.

DBpedia thus forms a rich resource that can be exploited, it grows as Wikipedia grows. Though not all infoboxes have been mapped to the ontology, the DBpedia team has introduced a public wiki⁷ for writing infobox mappings, editing existing ones as well as editing the DBpedia ontology. This allows external contributors to define mappings for the infoboxes they are interested in and to extend the existing DBpedia ontology with additional classes and properties.

2.1.2.2 DBpedia datasets

DBpedia project currently contains the following datasets:

1. Infobox dataset: This dataset has the coverage of all Wikipedia properties i.e. extracts from all properties from all infoboxes and templates within all Wikipedia articles. Extracted information is represented using properties in the *<http://DBpedia.org/property/>*

⁷http://mappings.dbpedia.org/index.php/Main_Page

Film ([Show in class hierarchy](#))

Label (en): film
Label (el): ταινία
Label (fr): film
Super class: [Work](#)

Properties on Film:

Name	Label	Domain	Range
amgid (edit)	amgId	Film	xsd:string
cinematography (edit)	cinematography	Film	Person
eTeatrId (edit)	e-teatr.pl id	Film	xsd:string
editing (edit)	editing	Film	Person
filmPolskiId (edit)	FilmPolski.pl id	Film	xsd:string
gross (edit)	gross	Film	Currency
imdbId (edit)	IMDB id	Film	xsd:string
narrator (edit)	narrator	Film	Person
author (edit)	author	Work	Person
basedOn (edit)	based on	Work	Work
completionDate (edit)	completion date	Work	xsd:date
creator (edit)	creator	Work	Person

Figure 2.3 – Portion of DBpedia ontology for thing:work:film showing properties (relations), domains and ranges
[\[http://mappings.dbpedia.org/server/ontology/classes/Film\]](http://mappings.dbpedia.org/server/ontology/classes/Film)

namespace. The names of these properties directly reflect the name of the Wikipedia infobox property. Property names are not cleaned or merged. Property types are not part of a subsumption hierarchy and there is no consistent ontology for the infobox dataset. Currently, there are approximately 8000 different property types. The infobox extractor performs only a minimal amount of property value clean-up, e.g., by converting a value like “June 2009” to the XML Schema format “2009-06”. This is useful only if ones application requires complete coverage of all Wikipedia properties and one is prepared to accept relatively noisy data.

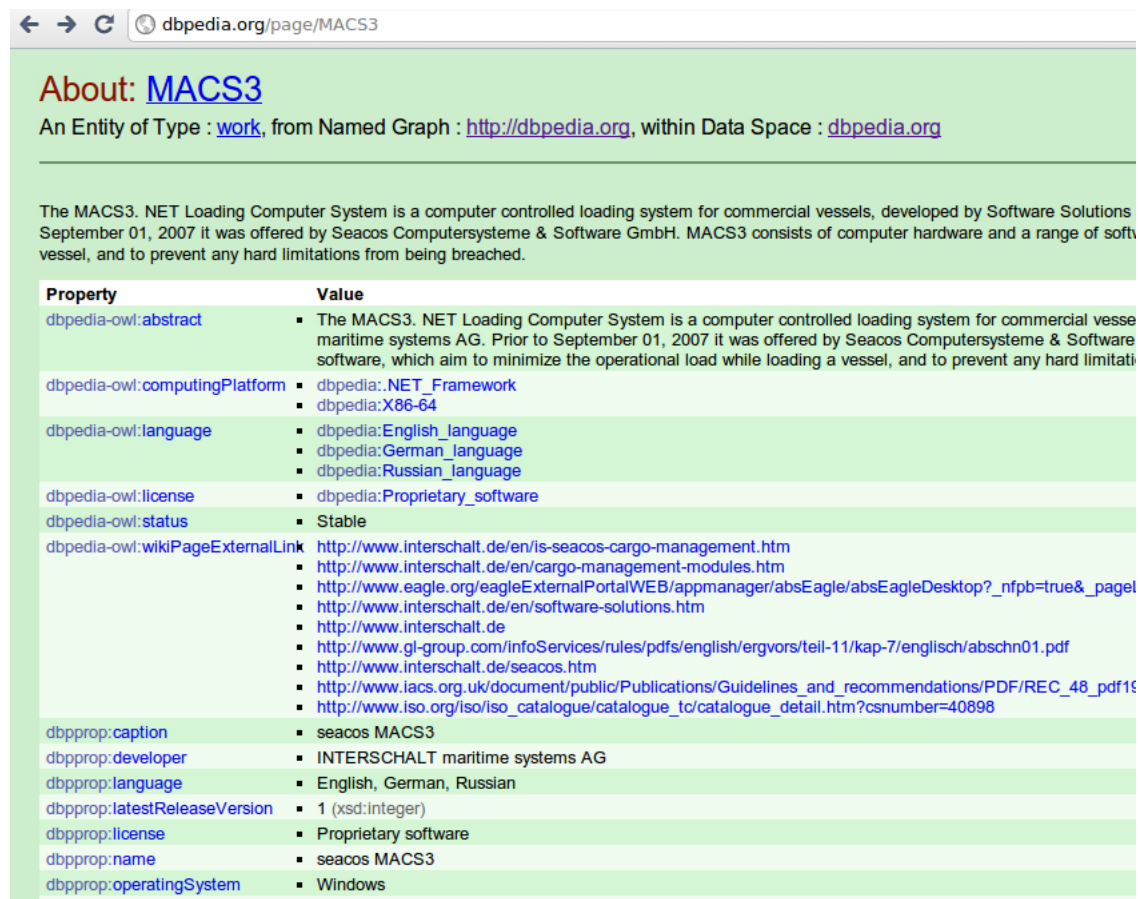
2. Infobox ontology dataset: This dataset is based on a new infobox extraction method which is based on hand-generated mappings of Wikipedia infoboxes/templates to the newly created DBpedia ontology. The ontology consists of 259 classes which form a subsumption hierarchy and have altogether 1300 properties. The mappings adjust weaknesses in the Wikipedia infobox system, like using different infoboxes for the same type of thing (class) or using different property names for the same property. Therefore, the instance data within the infobox ontology is much cleaner and better structured than the Infobox Dataset, but currently doesn’t cover all infobox types and infobox properties within Wikipedia.

Table 2.3 shows a comparison between the generic infobox dataset and ontology mapped dataset. DBpedia contains generic infobox data for 1,462,000 resources compared to 843,000 resources that are covered by the mapping-based approach. The mapping-based dataset contains 1300 different properties compared to 38,659 different properties for the generic dataset (including many synonymous properties) [Bizer *et al.*, 2009b]. To get the connected entities, all triples were removed from the datasets that did not point at a DBpedia entity, including all literal triples, all external links and all dead links.

	Resources covered	Connected Entities	Properties
Infobox dataset	1,462,000	1,029,712	38,659
Infobox ontology dataset	843,000	627,941	1300

Table 2.3 – Infobox datasets vs infobox ontology datasets (version 3.2)

As detailed by [Bizer *et al.*, 2009b], every DBpedia entity is described by a set of general properties and a set of infobox-specific properties, if the corresponding English Wikipedia article contains an infobox. The general properties include a label, a short and a long English abstract, a link to the corresponding Wikipedia article, (if available) geo-coordinates, a link to an image depicting the entity, links to external web pages, and links to related DBpedia entities. If an entity exists in multiple language versions of Wikipedia, then short and long abstracts within these languages and links to the different language Wikipedia articles are added to the description. A *resource* is a page describing a Wikipedia entity, showing the relations it holds with other entities or properties. Figure 2.4 shows an example of a DBpedia resource page showing the properties/relations and their values.



Property	Value
dbpedia-owl:abstract	<ul style="list-style-type: none"> The MACS3. NET Loading Computer System is a computer controlled loading system for commercial vessels, developed by Software Solutions September 01, 2007 it was offered by Seacos Computersysteme & Software GmbH. MACS3 consists of computer hardware and a range of software, which aim to minimize the operational load while loading a vessel, and to prevent any hard limitations from being breached.
dbpedia-owl:computingPlatform	<ul style="list-style-type: none"> dbpedia:NET_Framework dbpedia:X86-64
dbpedia-owl:language	<ul style="list-style-type: none"> dbpedia:English_language dbpedia:German_language dbpedia:Russian_language
dbpedia-owl:license	<ul style="list-style-type: none"> dbpedia:Proprietary_software
dbpedia-owl:status	<ul style="list-style-type: none"> Stable
dbpedia-owl:wikiPageExternalLink	<ul style="list-style-type: none"> http://www.interschalt.de/en/is-seacos-cargo-management.htm http://www.interschalt.de/en/cargo-management-modules.htm http://www.eagle.org/eagleExternalPortalWEB/appmanager/absEagle/absEagleDesktop?_nfpb=true&_pagel http://www.interschalt.de/en/software-solutions.htm http://www.interschalt.de http://www.gi-group.com/infoServices/rules/pdfs/english/ergvors/teil-11/kap-7/english/abschn01.pdf http://www.interschalt.de/seacos.htm http://www.iacs.org.uk/document/public/Publications/Guidelines_and_recommendations/PDF/REC_48_pdf19 http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40898
dbpprop:caption	<ul style="list-style-type: none"> seacos MACS3
dbpprop:developer	<ul style="list-style-type: none"> INTERSCHALT maritime systems AG
dbpprop:language	<ul style="list-style-type: none"> English, German, Russian
dbpprop:latestReleaseVersion	<ul style="list-style-type: none"> 1 (xsd:integer)
dbpprop:license	<ul style="list-style-type: none"> Proprietary software
dbpprop:name	<ul style="list-style-type: none"> seacos MACS3
dbpprop:operatingSystem	<ul style="list-style-type: none"> Windows

Figure 2.4 – Sample DBpedia entity -MACS3
[http://dbpedia.org/page/MACS3]

The datasets and ontologies can be queried using Sparql⁸ formal query language over DBpedia Sparql endpoints like:

- Public SPARQL⁹ endpoint over the DBpedia data set.
- The Leipzig¹⁰ query builder.
- The SNORQL¹¹ query explorer.

It would be of interest to many if such a rich resource of linked data could be queried using natural language.

The next section takes a look at QA, its dimensions, entailment based QA and related work on QA over linked data.

2.2 Question Answering

QA goes beyond retrieving pages or snippets where the answer is but involves concisely giving an answer to a user question. Question answering seeks to provide natural language interfaces to data sources where users can pose questions to the system and get concise answers back without the hurdle of understanding the underlying structure or complexity. This research area has been driven by emerging trends over the years from databases, digital documents, the Web and until recently linked data. The next sections takes a look at the evolution process that has been driving QA, and the current research trends.

2.2.1 Dimensions of QA task

Question Answering necessitates more than just a bag of words key word search of the terms in the question. The computer seeks to try and “understand” the interest of the user. This process can be called question interpretation [Frank *et al.*, 2007]. The information content in the question is what will reliably help the system to retrieve or find the relevant answer. A slack in this process would lead to a complete system failure in retrieving the answer no matter how good the answer extraction is. As detailed by [Webb and Webber, 2008], information in the question is what helps in retrieving the relevant documents where the answer is, and also extracting it and returning it to the user more reliably. Since in most cases the answer to a user query cannot be directly derived from the underlying documents or database, it is important that semantic interpretation of the question is done. This could involve shallow or deep linguistic and semantic analysis of the user query in order to get the semantic interpretation.

How the question is posed depends on the user, their level of expertise in the field and language of use. Though one can argue that a user question has the potential of being arbitrarily subtle, complex and rife with ambiguity, they can also be simple, straight forward and clear [Popescu *et al.*, 2003]. It is more often the case that a user will ask a question to the system,

⁸<http://www.w3.org/TR/rdf-sparql-query/>

⁹<http://dbpedia.org/sparql>

¹⁰<http://querybuilder.dbpedia.org/>

¹¹<http://dbpedia.org/snorql/>

on failure, he or she would rephrase the question with the hope that the system would then understand. This can be viewed as the query cycle and can have more than one iteration and ideally, the system serves as a cooperative partner in the information search process. This notion has some what lead to Interactive Question Answering.

Sometimes, a user would like to get information but it is available in a language different from the users. CLEF¹² addresses this issue of multilingual QA where the query and target source documents are in different languages. It is often an good idea to have this in mind when one is considering developing a QA system.

Differences in QA systems come depending on the type of question they deal with, the data sources and the scope [Lopez *et al.*, 2011]. This can be summarized as:

- Type of question - questions with different levels of sophistication and processing difficulty
 - Systems based on factoids (Wh-queries—who, which, what, how many)
 - Systems that deal with temporal reasoning (queries with start and end dates)
 - Systems that deal with commands or lists (name all, list all)
 - Interactive systems (able to engage in clarification dialogs with user)
 - Systems that deal with definition or opinions which requires analogical reasoning (what, why, how)
- Data sources - the source of the answer plays a role in the development of QA systems. Some can only answer questions from one type of answer source or capable of aggregating answers from different types of answer sources.
 - Databases
 - Unstructured data - Text data
 - Semi-structured - Wikipedia infoboxes
 - Structured semantic data - ontologies and RDF triples
- Scope - this defines the extent of what the system is capable of handling
 - Closed domain - systems handling only questions based on a particular pre-specified domain or small set of domains.
 - Open domain - these are domain independent systems able to exploit heterogeneous content i.e. different data sources and domains.

When designing any QA system, a couple of fundamental issues apply across the board. Thus QA techniques should be able to handle these issues. Some of these are spelled out below:

- *Open vs restricted domain* - There is demand for both angles, with the Web and linked data offering a solution for open domain, while still specific organizational obligations that require customization and defined vocabulary wanting closed domain QA.

¹²<http://www.clef-campaign.org/>

- *Deep vs shallow linguistic processing* - The computer should understand the user query and this could be enhanced by performing preprocessing and annotations, e.g. named entities, temporal expressions e.t.c on the input query. Linguistic processing helps in trying to understand the semantics of the user query. The key issue is to ensure that the answer retrieval time, from the moment the user submits the request, is within an acceptable range. Not all languages have deep NLP tools for processing text and thus shallow statistical techniques still play a crucial role.
- *Structured vs unstructured data sources* - The data source, to a good extent affects how the QA system is designed. A relational database is different from a collection of documents, web pages or linked data. The different data sources each have unique formal query languages. Adaptation is thus an onus when moving from one data source to another.
- *Single vs multiple ontologies* - The idea of multiple ontologies is to enable open domain QA. Multiple ontologies poses the issue of first having to select appropriate ontologies that would help in answering the user query. Systems like Watson [D'Aquin *et al.*, 2007] are able to collect linked data from the Web and indexes ontologies (approx. 8,300 ontologies).
- *Understanding of the information in the user query* - This is generally an Artificial Intelligence task, enabling the computer to understand natural language. All in all, for QA systems to reach their optima, the system has to try and interpret the content and context of the user query.
- *Giving some form of higher level representation of the user query depicting its semantics* - The representation is an expression of what the system understands to be the information content of the query. To support multilingualism, it should be abstracted away from any language dependence.
- *Mapping the question to a formal query language for answer retrieval* - Each data source has a unique formal query language for retrieving data from it and hence, the abstract representation is mapped at runtime to a corresponding formal query in order to retrieve the answer.
- *Presenting the answer in an acceptable format* - The answer returned to the user should be presented in a readable and pleasant format according to human computer interaction recommendations.
- *Multilinguality* - Information is available in many different languages. Current QA systems support multilingualism by developing language specific modules that support different languages.

2.2.2 Natural Language Interface to Database (NLIDB)

With the advent of computers, data began to be stored in them mostly in form of databases (DB). As early as the 60's, NLIDBs emerged with systems like LUNAR, as mentioned in [Androutsopoulos *et al.* 1995], which tried to answer questions over a database containing chemical analysis of moon rocks. Traditionally, access to such kind of database were achieved by

formal query languages like SQL. The natural language interface was used to map natural language questions to the formal query language, retrieve the result and present it back to the user. This was to alleviate the user from having to understand the structure of the database and the specified formal language to query it. At least the users would express themselves more naturally, and possibly be able to have some interactivity with the system.

The main idea of NLIDB is to take a users input query and try to understand the information content and context. This is done by first performing some linguistic analysis to the question, then further enriching the output with semantic annotations. The next phase transforms the user query in an intermediate language independent representation which gives the systems understanding of the semantic content of the user query. This representation is then mapped onto a formal query language SQL¹³ which would retrieve the answer from the database.

Some challenges faced by NLIDB's as presented by [Androutsopoulos *et al.* 1995] include linguistic vs conceptual failures, i.e. when no answer is returned back by the system, is it as a result of the system not having enough linguistic coverage or that the user query is out of the conceptual coverage of the system (i.e. the concepts that are in the domain of the system). This is a problem of mapping the words used in the user query with DB structure. Different word units could be used to ask for the same information, and the challenge existed in mapping the different ways onto the same formal query. This could also be viewed as an inherent challenge of QA as a whole. NLIDB is limited in that it majors only on interfaces over databases and not free text, a fact that tends to limit NLIDB to closed domain rather than open domain.

Over the years, research has developed pinpointing various data sources from where the answers could be retrieved. General QA in contrast to NLIDB is a broader term that is not just restricted to DBs. The data source can be structured data (e.g. database), semi-structured or unstructured (free text). After QA over databases, there was a surge in QA systems over text data due to the emergence of the Web and digital documents. Such systems were based on paradigms that build QA over Information retrieval¹⁴ systems. An example is by [Harabagiu *et al.*, 2003] where they used the words in the query as key words to retrieve relevant documents, then they identified snippets from the documents that contained possible answers and finally used an answer extraction module to extract the answers from the snippets.

Simple word search was not effective enough, and understanding the meaning of the words in the user query became important. Ontologies were introduced to help bridge the gap between the user query and the concepts used to represent the data source –database in this case. [Frank *et al.*, 2007] proposes a system that works on retrieving answers from restricted domains over databases. They perform deep linguistic analysis, HPSG¹⁵, of the question relative to the domain, in this case, they have two application domains: Nobel prizes and Language Technology WORLD¹⁶. A similar approach is taken by [Lesmo and Robaldo, 2006], but instead of HPSG, they use a dependency parser that gives dependency trees¹⁷, each node being a word from the input question. In both systems, they perform a mapping of the words in the user question to

¹³<http://www.sql.org/>

¹⁴http://en.Wikipedia.org/wiki/Information_retrieval

¹⁵http://en.Wikipedia.org/wiki/Head-driven_phrase_structure_grammar

¹⁶<http://www.lt-world.org/>

¹⁷http://en.Wikipedia.org/wiki/Dependency_grammar

concepts (nodes) in the ontology. The result is an ontological query, i.e. a representation (the user query expressed as ontological concepts) that specifies the meaning of the user question in the terms of the way the computer “thinks” about the domain. Finally the ontological query is mapped onto SQL query for answer retrieval from the DB. Figure 2.5 shows the system architecture. The ontologies are being used to mark-up data, both in free text and in structured data sources, and in query expansion to enrich the user query for better answer retrieval.

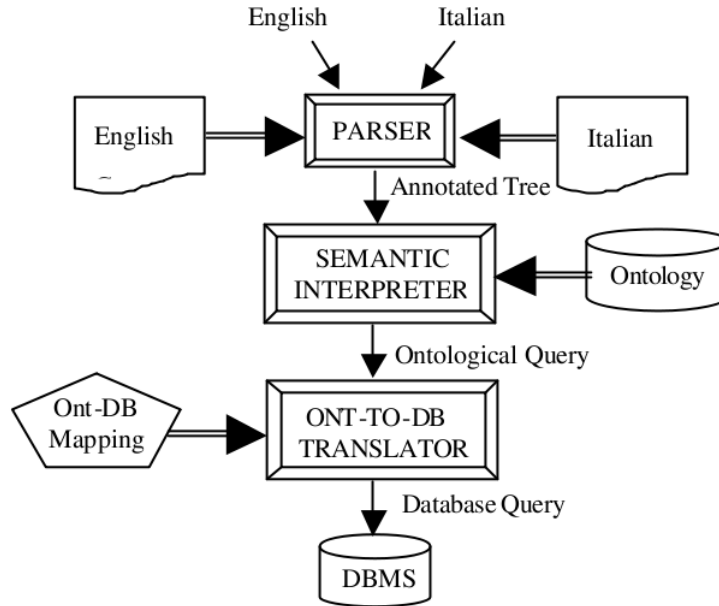


Figure 2.5 – From natural language to DB via ontologies [Lesmo and Robaldo, 2006]

Next, entailment based QA paradigm is presented, with Qall-me as a case study, to give insights on its architecture in relation to how it can be applied to linked data.

2.2.3 Entailment based QA

[Dagan and Glickman, 2004] defines Textual entailment as ‘the relationship between a coherent Text T and a language expression, which is considered as a hypothesis H ’. The idea is that variable texts can be used to express the same kind of information, and thus one piece of text can be entailed in another, i.e. the *Text* T entails the *Hypotheses* H . This means that the meaning of H can be fully derived from the meaning of T . As noted in [QALL-ME Consortium, 2009a], Recognizing Textual Entailment (RTE)¹⁸ techniques can allow one to ‘deal with the variability expressed within the questions through semantic inferences at the textual level’. Operating on the lexical-syntactic level alleviates the issues of needing deep NLP tools that might not be available for all languages. In Question Answering, T is the user question posed to the system, and H is the minimal forms or relational patterns.

A relational pattern is a text string associated to an ontological relation. It has fixed elements i.e. words expressing the relation and typed variables representing the entities in relation. For instance to express the “invent” relation in text, one could say *[PERSON:DOMAIN] invented the*

¹⁸http://www.aclweb.org/aclwiki/index.php?title=Recognizing_Textual_Entailment

$[THING:RANGE]$, $[PERSON:DOMAIN]$ created the $[THING:RANGE]$, $[PERSON:DOMAIN]$ discovered $[THING:RANGE]$ *e.t.c.*, where the $[DOMAIN]$ is the subject of the relation whose type is $[PERSON]$ and the $[RANGE]$ is the object of the relation whose type is $[THING]$. In linked data scenario, the patterns represent the various ways in which the predicate can be expressed in surface form, and thus fit in the triple model of linked data.

Figure 2.6 gives the architecture of how RTE is used in entailment based QA. For each ontological relation, there exists a set of patterns that represent the various ways in which that particular relation is expressed lexicographically. The set of relational patterns need to be acquired in advance, thus through RTE, the QA system can handle all reformulations of relational patterns in the user input question. At runtime, the user question which is the text T would entail one or more of the relational patterns. The output of the RTE gives the list of relations that are expressed in the user query. In Qall-me, the relational patterns are extracted from a corpus of user questions. Patterns too could be extracted from text data, a point that this thesis studies its feasibility.

A detailed example of patterns and how the entailment paradigm works is explained in the following section with the help of a running example from the Qall-me RTE based system.

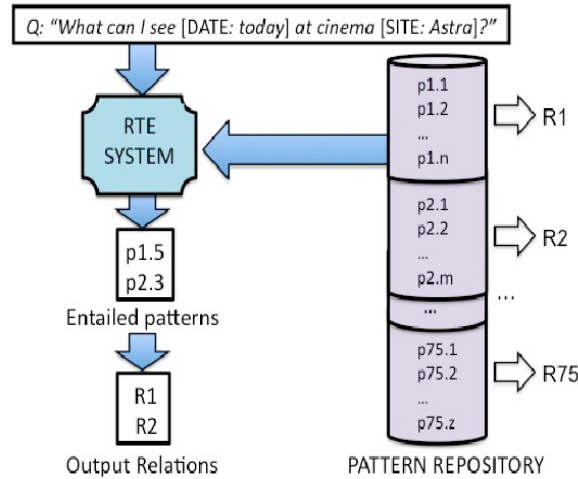


Figure 2.6 – TE-based Question Answering [Negri and Kouylekov, 2009]

2.2.3.1 Qall-me

Qall-me, **Q**uestion **A**nswering **L**earning technologies in a multi**L**ingual and **M**ultimodal **E**nvironment, is a state of art open source framework, that has a reusable architecture skeleton for building multilingual QA systems that answer questions with the help of structured answer data sources from freely specifiable domains [Ferrández *et al.*, 2011]. The framework is characterized by multilinguality, the data sources which are majorly structured data sources in form of RDF triples or simply XML documents with a specialized structure and the ontology [QALL-ME Consortium, 2009c] containing descriptions of both concepts for the target domain and possible relations between these concepts. The ontology is used as a schema for representing the structure of the answer and to cross the language barrier in multilingual QA i.e. the answer

is described by means of ontology vocabulary that gives a representation that is independent from the original language of the data; therefore, using the ontology vocabulary to create a mapping from the original question to a query, then applying the query to the answer data, one can surpass the language barrier. Table 2.4 gives the size of the ontology for the cinema domain used in Qall-me system.

	Qall-me Ontology
Classes	261
Properties	55
Datatypes	55
Instances	160000

Table 2.4 – Qall-me ontology

The system’s main idea is to view QA as an entailment problem in that, there is an entailment relationship between the user query and the Minimal Relational Patterns (MRP). A MRP is a natural language question string containing variables that correspond to concepts of the ontology. First the notion of minimal needs to be defined when it comes to relational patterns. In [QALL-ME Consortium, 2009b], it is said that a relational pattern P expresses a relation $R(arg1, arg2)$ in a certain language L if it can be agreed by speakers of L that from the meaning of P one can infer the relation between $arg1$ and $arg2$. Formally “ Given a set $P = p_1...p_n$ of relational patterns for a relation R , a pattern p_k belonging to P is minimal for the relation R if $\forall p_i \in P, p_k \Rightarrow p_i = \phi$. A pattern p_k is minimal if none of the other relational patterns contained in P can be derived from p_k .” Every pattern is associated to one or more ontological relations that it expresses [Negri and Kouylekov, 2009]. Recognizing Textual Entailment technique does the mapping between the MRP and user query. The MRPs are the Hypotheses H while the user question is the Text T that entails some MRPs. A finite bijective mapping between the MRPs and the corresponding (Sparql) query patterns are pre-defined. The query patterns in turn retrieve the desired answer from the answer data source.

Example:

MRP : “Where can I see the movie [MOVIE]?”

R1: HASMOVIESITE(MOVIE:?,SITE:?)

Corresponding Sparql query

```
SELECT ?cinemaName
WHERE ?movie qmo:name 'MOVIE' .
?cinema qmo:showsMovie.
?cinema qmo:name
?cinemaName.
```

In Qall-me the average number of relational patterns per relation used in the Cinema domain was set to 15 patterns per relation [QALL-ME Consortium, 2009b]. The effect of this number is still an open issue when it comes to an open domain scenario.

2.2.3.2 System components

The system components can either be language specific, location specific or system wide. Figure 2.7 shows the system architecture and how each component relates to one another in a pipeline fashion. To give a deeper understanding of the various components, a running example is used. We start with a user query asked in Italy at 1600hrs “Where can I watch the movie *Matrix* today?”. The context information from the question are: Location: “Bolzano”, Time: “2011-01-14T16:00:00”.

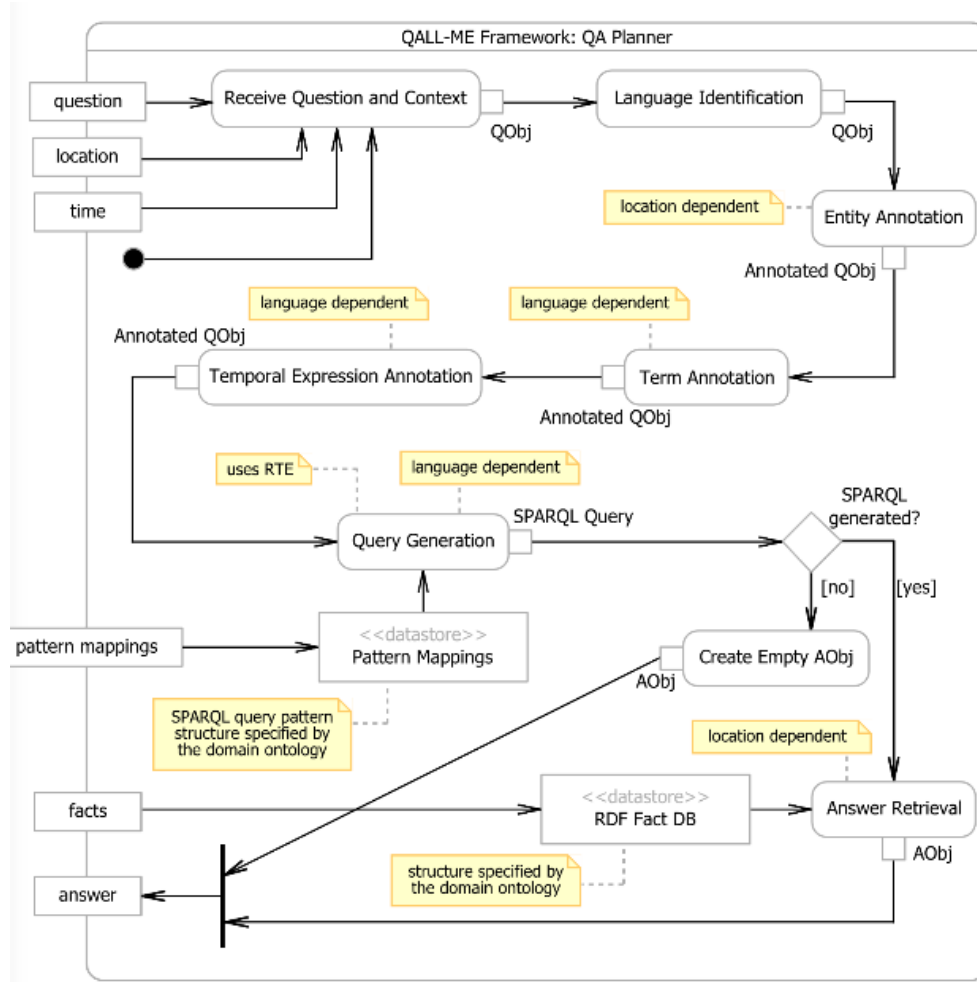


Figure 2.7 – Qall-me Framework [QALL-ME Consortium, 2009a]

1. *QA Planner* - handles the whole QA process, i.e. it is responsible for receiving the input question and selecting the appropriate components that would lead to getting the answer.
2. *Question Analysis*: Transforms the input question to NL pattern P_Q
 - (a) *Language identification component* - As the name suggests, it identifies the language of the user query. In this case, it detects the language as English.
 - (b) *Entity annotation component* - This is a location-dependent component and thus the Italian annotator is chosen. It creates an annotated question object regarding the

entities found.

“Where can I watch the movie [MOVIE:Matrix] today?”

- (c) *Term annotation component* - This is a language-dependent component and thus the English version is chosen. It recognizes term expressions relevant for answer retrieval e.g. hotel facilities like swimming pool, hair dryer e.t.c that are represented differently in different languages. This example has no relevant terms.
- (d) *Temporal expression annotation component* - This is also a language-dependent component. It detects and normalizes temporal aspects.

P_Q = “Where can I watch movie [MOVIE:Matrix] [2011-01-14]?”

- 3. *Query generation component* - This component is also language-dependent. It uses the appropriate language dependent RTE engine to determine the MRP(s) from the repository that corresponds to the user question by testing whether the MRPs are entailed in P_Q . In this case, the MRP “Where can I see [MOVIE]” is chosen and its corresponding Sparql query instantiated to reflect the entity in the user question.

Sparql Query:
 SELECT ?cinemaName
 WHERE ?movie qmo:name 'Matrix' .
 ?cinema qmo:showsMovie.
 ?cinema qmo:name
 ?cinemaName.

- 4. *Answer retrieval component* - This component is location dependent. It takes the Sparql query and retrieves the answer. It uses information about terms and temporal expressions stored in the question object to filter and retrieve correct answers. In the case that the query generation component selects more than one MRP entailed by different parts of the question, the associated queries are composed into a single query for answer retrieval. If no query is generated, it is assumed that the user probably posed an out of domain question.

Qall-me combines the expressive power of ontologies together with RTE to answer user questions. From the architecture (see Figure 2.7), the modules that this thesis is interested in are the source of the *patterns* and how the *pattern mapping* is done in order to get the relations expressed in the user query, generate the sparql and finally get the answer.

The results of the experiments carried out using Qall-me in the tourism domain for four languages English, German, Spanish and Italian, are given in Table 2.5. The evaluation was not on whether the system could retrieve the correct answer but it measured instead whether or not the system could retrieve the correct sparql pattern. This is because once the correct pattern is found, the answer extraction only requires retrieval of the answer from the linked data source. The results were based on patterns extracted from a corpus of user questions on tourism domain.

The question this thesis seeks to answer is the feasibility of entailment based QA, using relational patterns extracted from Wikipedia text rather than a corpus of questions, to provide

	Questions	Correct	Wrong
English	167	74 (44.31%)	93 (55.68%)
German	214	120 (56.04%)	94 (43.92%)
Spanish	58	50 (86.20%)	8 (13.79%)
Italian	99	46 (46.46%)	53 (53.53%)

Table 2.5 – Qall-me evaluation results [Sacaleanu *et al.*, 2008]

a natural language interface to DBpedia. Relation extraction techniques are used to acquire the relational patterns. The notion of relation extraction is explained in the following section.

Next, an overview of QA over linked data and example QA systems providing natural language interfaces to linked data are presented.

2.2.4 Related work on QA over Linked data

The expressive power of ontologies has enabled them to be exploited in marking-up data sources and expanding user queries. Ontologies have helped in the system portability problems with their independent nature of storing knowledge thus one could potentially replace one ontology with another in a QA system in order to move to a new domain. Ontologies have also played a role in bridging the gap between user query and system vocabulary through query expansion, handling ambiguities in user query and ontology “neighborhood” look-up. Such advances could help give insight when moving to QA over linked data. This section gives a description of some systems that exploit ontologies to query over linked data.

In order to explore different approaches for QA over linked data, Question Answering Over Linked Data (QALD)[QALD-1 proceedings, 2011] workshop has been recently formed. The aim of the workshop is to explore different systems and to facilitate comparison between different approaches. Several systems participated in the shared task challenge, and it was noted that the main challenge faced in QA over linked data is dealing with heterogeneous, distributed and huge amount of interlinked data. The goal of the challenge was to get an overview of the strengths and shortcomings of QA systems and how they deal with huge, heterogeneous and distributed linked data. DBpedia and musicbrainz¹⁹ linked data sources were used in the challenge.

Table 2.6 shows the results of the two state-of-art systems that participated, powerAqua [Lopez *et al.*, 2009] and Freya [Damjanovic *et al.*, 2010], using DBpedia linked data source. The systems are presented next.

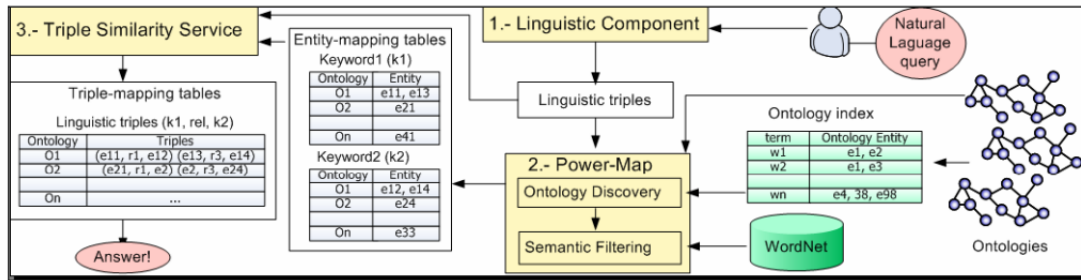
¹⁹<http://wiki.musicbrainz.org/LinkedBrainz>

	total	processed	right	wrong	recall	precision	f-measure
FREyA	50	43	27	16	0.54	0.63	0.58
PowerAqua	50	46	24	22	0.48	0.52	0.5

Table 2.6 – QALD-1 challenge: results for DBpedia linked data source

2.2.4.1 PowerAqua

PowerAqua[Lopez *et al.*, 2009] takes as input a natural language query and is able to return answers drawn from relevant semantic sources from anywhere on the semantic web. PowerAqua retrieves information in an open-domain multi-ontology architecture as shown in Figure 2.8.

Figure 2.8 – PowerAqua flow diagram [Lopez *et al.*, 2010]

The main issues around open-domain QA with multiple ontologies is first to identify the relevant ontologies for the given input query, perform ontology mapping and word sense disambiguation in order to avoid potentially incoherent constructions and lastly integrate the multi-sources to give an answer.

In their system first, the *linguistic component* takes as input a natural language question and translates it into a set of query-triples(QTs), by identifying associations that relate terms (entities) together. For instance, the question “Who plays in the rock group nirvana?” can be translated into QT <person/organization, plays, rock group nirvana>, i.e. <term, relation, term>. This approach did not give any work on questions that cannot be modeled in the triple format, e.g. why-questions. The QT model is in line with linked data model of subject, predicate, object, and its role is to provide an easy way to manipulate the input. Next there is the *powerMap* module that is responsible for identifying the semantic resources that may answer the given query and for producing the initial element-level mappings between the QT terms and entities in these sources. PowerMap uses the Watson Semantic Gateway²⁰ as the entry point to the semantic web. The output of the powerMap is *Entity Mappings Tables* (EMTs) where each table associates each QT term with a set of entities found on the semantic web as shown in Table 2.7. After validation and filtering, the *Triple Similarity Service* (TSS) takes as input the EMTs and returns a set of *Triple Mapping Tables* (TMTs) which specify complete mappings between a set of Query Triples and the appropriate Ontology Triples (OT) as shown in Table

²⁰<http://kmi-web05.open.ac.uk/Overview.html>

2.8. Finally the merging and ranking component generates the final answers from the returned ontology triples.

Rock group Nirvana, rock group, group nirvana		\emptyset
Nirvana	<i>Music</i> : Nirvana (type: group); <i>TAP</i> : MusicianNirvana (type: person); <i>SWETO</i> : Nirvana Meratnia (type: researcher); <i>KIM</i> : Eden (synonym); <i>Spiritual</i> : Nirvana; <i>Magnatune</i> : Passion of Nirvana (type: “track”), ...	
Rock	<i>Music</i> : rock (as a type of genre); <i>SWETO</i> : Michael_Rock, Sibyl Rock, etc; <i>ATO</i> : rock (as a type of substance), Ayers_Rock (as a place); ...	
Group	<i>Music ontology</i> : group, ...	
Person	<i>Music ontology</i> : musicians (as a hyponym of person according to TAP), <i>TAP</i> : person, <i>KIM</i> : person, <i>Magnatune</i> : musicArtist (hyponym),...	
Play	<i>KIM ontology</i> : sport (as synonym of “play”)...	

Table 2.7 – Partial view of the EMTs for QT <person/org, plays, rock group nirvana> [Lopez *et al.*, 2009]

<person / organization, play, Nirvana>	
<i>SWETO</i>	<Nirvana Meratnia, IS_A, person>
<i>Magnatune</i>	<MusicArtist (hyponym), maker (ad-hoc), Passion of Nirvana>
<i>Music</i>	<Musician (hyponym), has_members (ad-hoc), Nirvana>
<i>TAP</i>	<Person, hasMember (ad-hoc), MusicianNirvana>
<rock, ?, nirvana>; <group, ?, nirvana>	
<i>Music</i>	<Nirvana, has_genre, rock>; <nirvana, is-a, group>

Table 2.8 – The TMT for OTs in ontologies that match the QTs [Lopez *et al.*, 2009]

This work moves away from the typical NLIDBs and tries to answer questions from the semantic web using multiple ontologies. It is able to achieve a significant level of open domain QA by exploiting a range of ontologies. The building of QTs from user question, using linguistic analysis, is a positive step in trying to map the user question in the linked data model. In the Qall-me approach, the user question textually entails one or more patterns in order to discover the relations in the question. The patterns similarly fit in the linked data model of indicating the subject, predicate and object of the question. The two approaches both aim at representing the user question in a way that facilitates answer extraction from linked data.

2.2.4.2 Freya

Freya [Damljanovic *et al.*, 2010] – **F**eedback, **R**efinement and **E**xtended Vocabulary **A**ggregation – also seeks to tap the vast amount of linked data found on the Web by providing a natural language interface for querying over it. Their work flow can be broken down into three main steps: identification and verification of ontological concepts found in the user query, generation of Sparql, and presentation of the retrieved answer to the user. Figure 2.9 gives the work-flow of the Freya system depicting the interactive nature of the system in getting clarifications from the user.

First the system takes as input the user question and does syntactic parsing (Stanford parser²¹) and analysis, plus ontological reasoning, the output being the user question annotated with *Potential Ontology Concepts* (POC). POC are terms found in the user question which have the probability of being linked to an *Ontology Concept* (OC). OCs are either instances/individuals, classes, properties, or datatype property values. Several heuristic rules are used in order to identify the POC from the user question. For instance noun phrases and adjectives are identified as POCs. For better performance, the system engages the user into a dialog to precisely identify the user's desire and identify the POC. Next the system does a consolidation where it tries to map the POC to the OC without considering any grammar used in the question. If there are ambiguous POCs or no overlap between POC and OC, the system generates suggestions, either disambiguation or mapping dialogs with the user, seeking for clarification. For each ambiguous or non overlapping POCs, the systems generates suggestions to the user. After every POC is confirmed and resolved, the system identifies the answer type, then combines the OCs to triples, and finally from the triples generate the sparql query.

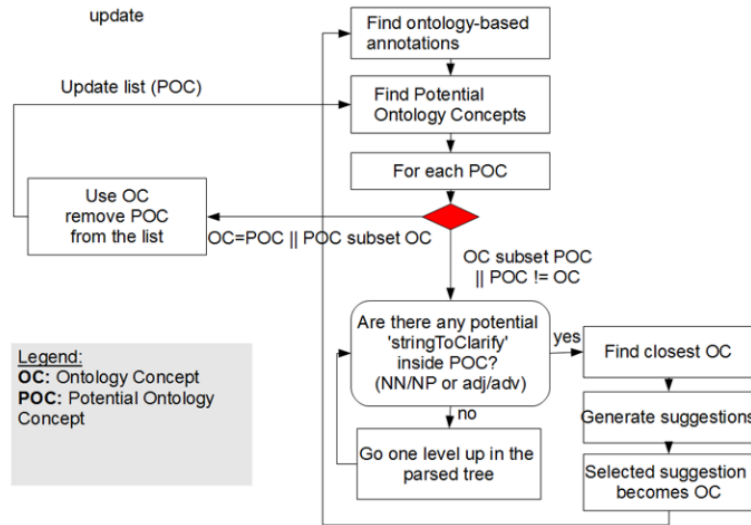


Figure 2.9 – Freya work flow showing the validation of POC through the user interaction [Damjanovic *et al.*, 2011]

The question “*Who plays in rock group nirvana?*” could be translated into the following OCs:

ont^a:playIn - *PROPERTY*
ont:rockBand - *CLASS*
ont:nirvana - *INSTANCE*

^a*ont* is used as a generic name to represent some ontology

If the first two OCs derived from the question are referring to a property and a class respectively, one *joker* is added before them. The elements are then transformed into:

²¹<http://nlp.stanford.edu/software/lex-parser.shtml>

? - JOKER1 <i>ont:playIn</i> - <i>PROPERTY1</i> <i>ont:rockBand</i> - <i>CLASS1</i> <i>ont:nirvana</i> - <i>INSTANCE</i>

Next the system generates a set of triples from the OCs. Two triples are generated from this example:

<i>?joker1</i> - <i>ont:playIn</i> - <i>ont:rockBand</i> <i>ont:nirvana</i> <i>ont:typeOf</i> <i>ont:rockBand</i>
--

The generated triples are then combined and used to build up the desired Sparql query for answer retrieval from the linked data source.

The results of QALD-challenge are presented in table 3.1. The main challenge they encountered while using DBpedia was on deciding the property(relation) to use to link a question term onto an OC, due to the large number of system suggestions. For example, the sentence “*Who created English Wikipedia?*” could be mapped to *dbp:created* or *dbo:author*. The right answer is only got after using *dbo:author* ²² This problem is caused by the heterogeneous nature of the DBpedia ontology.

Freya, though applying one ontology at a time, tries to map the user query to ontology concepts for triple generation, which in turn are used to generate the Sparql to retrieve the desired answer. It can also be seen here the attempt to transform the user question into a triple model in line with linked data models.

2.2.5 Challenges for QA over linked data

Though linked datasets literally may contain the answer to millions of questions, the major challenge revolves around locating and exploiting the relevant information in order to extract the desired answer. Currently, the state of art user interfaces that explore linked data include SPARQL endpoints like the Leipzig query builder, Public Faceted Web Service Interface ²³. As detailed by [Lopez *et al.*, 2010], the existing query approaches for linked data are restricted and only facets and query builder interfaces provide an efficient way to pose complex and expressive queries. The problem with the existing interfaces is that the user, and not the application has the responsibility of formulating the query in the way that the system can retrieve the desired answer. Some challenges that researchers have faced when querying over linked data sources are presented below.

1. *Scalability: size of ontology and number of instances.*

DBpedia is very large, the data set currently describes 3.5 million “things” with over half a billion “facts” (November 2010). It is important to note that the links between the entities are more rich in terms of context than the entities themselves. Exploring these connections between entities is the crucial aspect and is where the challenge is. For a user, the response time to a query is quite important and the time the system takes to retrieve

²² *dbp* stands for <http://dbpedia.org/property> while *dbo* stands for <http://dbpedia.org/ontology> namespaces.

²³ <http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VirtuosoFacetsWebService>

the desired answer is directly proportional to the size of the ontology and the instantiated data. Optimized semantic search algorithms are needed. Scalability goes beyond a single ontology to the fact that there are thousands of ontologies currently indexed online. If we take a look at semantic search engines, Falcon²⁴ indexes 7 million of RDF documents and 4,400 ontologies, Swoogle²⁵ indexes over 10,000 ontologies while Watson²⁶ indexes approx. 8,300 ontologies. With these kind of numbers, performance and scalability challenges are still open for more research.

2. *Heterogeneity of the ontology*

One has to move from different specific domain ontologies to one huge ontology with a large number of relationships from various domains. DBpedia ontology is multi-domain and even for a potentially unambiguous word, it would still have in the order of thousands number of mappings. Thus analyzing the ontological context of all potentially relevant hits in order to select the ones containing the answer would be unacceptably slow, therefore new filtering heuristics are required. Across ontologies, similar concepts might be expressed differently, and as highlighted in [Lopez *et al.*, 2011], the various data sources exhibit differences in granularity as the data is not centrally maintained and managed.

3. *Quality of data source*

The effectiveness of any query which uses ontological semantics to perform basic light-weight inferences based on the taxonomy and relationships, also relates to the quality of the sources they are querying. As mentioned in section 2.1, DBpedia contains two datasets, dataset mapped to the ontology and the generic dataset covering all infoboxes. Although the percentage of properties pointing to other DBpedia entities is much higher in the dataset mapped to ontology (53%) than in the generic dataset (23%) [Bizer *et al.*, 2009b], the coverage is lower (see Table 2.1). Also, although in the generic approach, coverage of all infoboxes is complete, synonymous attribute names are not resolved and there are errors in determining the data type of a value. This has an impact on the effectiveness of finding answers, and in the general query performance. The same argument can be extended to the thousands of other ontologies found in the semantic web, they vary in quality and their trustworthiness have to be analyzed. It is common for errors and inconsistencies to appear across the different ontologies, and thus quality of the data is an inherent challenge for QA over linked data.

Lessons learned: Linked data is based on the RDF subject, predicate and object model and QA systems over linked data like [Lopez *et al.*, 2009, Damjanovic *et al.*, 2010] attempt to transform the user input question into this kind of model to ease query manipulation and answer retrieval. The relational patterns used in entailment based QA show the different ways in which the predicates can be expressed in surface form, and gives the type of the domain and range entities.

²⁴<http://ws.nju.edu.cn/falcons/objectsearch/index.jsp>

²⁵<http://swoogle.umbc.edu/>

²⁶<http://kmi-web05.open.ac.uk/WatsonWUI/>

The next section sheds more light on relation extraction, and provides examples of relational patterns that can be extracted from sentences.

2.3 Relation extraction

Relation extraction involves automatic or semi-automatic –with human assistance–, acquisition of relations between concepts from text data. Such texts could include published documents, newspapers or just the world wide web. It has various application in the field of language technology from information extraction, ontology population, semantic linking to QA among others. Relation extraction can be viewed as a task of filling range slots, where given an entity and a target relation, one finds a corresponding range value that fills the slot. It also involves extracting from raw texts, existing relations between entities that are not known before hand. Another task, which is the one described in this thesis, involves getting surface patterns of how a relation between two entities can be expressed with the aim of applying such patterns to get more instances of the same relation or to answer questions about a given entity. One of the major challenges in the field is getting labeled data for testing and optimizing system performance.

As pointed out earlier, relational patterns are of uttermost importance in a number of applications. However, the ability to automatically collect large-scale relational patterns for an high number of relations, and with a good quality, is currently limited by two main factors: (i) as the state-of-art of Information Extraction techniques are based on automatic learning from training data, which are currently available for very few relations, the potential coverage for such approaches is very poor; (ii) on the other hand, non supervised approaches suffer from lack of precision, as potential relational patterns still contain too much noise to ensure good performance. It is in this respect that we seek to create labeled data from Wikipedia to improve automatic relation extraction.

Wikipedia has been chosen as the source of textual data because of its diversity and maintenance. There are many texts describing different instances of the same relation and thus, one could exploit the distributional properties of these relations and the various ways in which different writers express them. The presence of DBpedia is an advantage for getting the instances of domain and range of the relations.

Examples of potential patterns from sentences:

1. Relation "birthDate" - the range is a property

- Sentence - *Mozart was born in 1756.*
 - Potential Pattern - [PERSON] born in [birthDate]
- Sentence - *Nalla Reddi Naidu was born on January 13, 1917.*
 - Potential Pattern - [PERSON] born on [birthDate]
- Sentence - *Johnnie Brannigan (born June 11, 1982).*
 - Potential Pattern - [PERSON] (born [birthDate])

2. Relation "writer" - the range is also an entity

- Sentence - *Cigarette Girl* is a 2009 film written and directed by Mike McCarthy.

- Potential Pattern - [WORK] written by [PERSON]
- ⋈ The film was Bergman’s first feature as director and *he* also *wrote* the *screenplay*.
- [PERSON] wrote [WORK]
- ⋈ "*Eventful*" is the second single *released by Ami Suzuki* under the label Avex Trax.
- [WORK] released by [PERSON]

2.3.1 Role of relational patterns in QA

Research has been done previously on the use of surface text patterns for Question Answering, [Roth *et al.*, 2002, Ravichandran and Hovy, 2002, Soubotin and Soubotin, 2001], presented work on automatically extracting surface text patterns for various relations from the web. Some example patterns for the “inventor” relation include: $\langle \text{ANSWER} \rangle$ *invented the* $\langle \text{NAME} \rangle$ *in*, $\langle \text{ANSWER} \rangle$ ’ *s invention of the* $\langle \text{NAME} \rangle$, $\langle \text{ANSWER} \rangle$ *invents the* $\langle \text{NAME} \rangle$ e.t.c. [Ravichandran and Hovy, 2002]. The patterns were used to retrieve answers from text. From the user input question, first the system identifies the question type and the *entities* that the question is asking about. Next it creates a query from the entities and performs information retrieval to find snippets in text data where the entities are mentioned. Next the entities are replaced with question tags, in this case the “NAME” tag for the “inventor” relation. Using the patterns, the “ANSWER” value is matched and retrieved. Sorting and ranking is done to get the answer from the patterns with highest precision. The highlighted scenario shows the use of surface relational patterns for Question Answering using information retrieval approach that has been quite common in the recent past.

Further to this idea, [Shen *et al.*, 2005], instead of just surface patterns, they used syntactic relational patterns extracted from dependency trees. They defined a syntactic relation pattern as the smallest sub-tree that spans an answer candidate node and one question key word node in the dependency tree. The systems highlighted above used the patterns in the context of information retrieval, i.e. using the patterns to retrieve potential answers from snippets returned by the information retrieval component to be containing the answer. Though the patterns perform well on some type of questions like “birthDate”, for others like definition questions, the performance is dismal. Factors for this include lack of external knowledge to guide the patterns, long distance dependencies and varieties in the way the answer terms is expressed e.g. for date.

With this kind of approach, the answer data source for answer extraction is limited to free text and such patterns could be extended or modified for use in answering questions from linked data sources. In entailment based QA, like Qall-me, the patterns have been used to act as the hypotheses to which the user query will entail, in order to retrieve the desired answer. This paradigm has been tested using question patterns learned from a corpus of questions and this thesis presents a feasibility study on the use of patterns extracted from text rather than from a corpus of questions for use in entailment based QA.

2.3.2 Evaluating relation extraction systems

Automatic extraction of relational patterns is a non trivial task and there is need for evaluation and optimization mechanisms. Various researchers performing relation extraction have employed

different evaluation schemes depending on the extraction technique. Some of them who perform relation extraction on Wikipedia are highlighted next.

[Nguyen *et al.*, 2007] aims at extracting binary relations between entities in Wikipedia articles. They defined a binary relation as a triple (e_p, rel, e_s) where e_p is the domain (which they call principal entity) and e_s is the range (which they call secondary entity) of the relation, and rel indicates the directed relationship between the entities e.g. (Microsoft, Founder, Bill Gates). The system predicts only the relations between the domain and each mentioned range in the article, and makes an assumption that the relationship between an entity pair can be completely expressed in one sentence. For this reason, only sentences with both the domain and range entities are analyzed and those that express more than one relation are eliminated. They exploited the relations and ranges found in the infoboxes to create training data which are sentences expressing the relation between the domain and the range. Sentences that express more than one relation were excluded from the training set. They used the dependency paradigm, such that the shortest path from the domain to the range gives the syntactic structure expressing the relationship between the pair. To evaluate the extraction algorithm, they utilized the relations and range values found in the infobox to automatically build a gold-standard of sentences expressing the target relations, with the confidence that the infoboxes were created by humans and therefore are correct. They used standard precision and recall metrics for evaluation against the built gold-standards and the best system result was an F-measure of about 0.5.

WRAP [Vila *et al.*, 2010] extracted paraphrasing patterns from Wikipedia pages. The paraphrases are the wordings used to express the relations. An example of possible paraphrasing pattern would be $\{text\}/X/\{text\}/Y/\{text\}/Z/\{text\}$ where X is the domain entity (author and person in this case) and Y is the range entity or property (work, birth or death information). For “person” experiments only $[X]$ and $[Y]$ were considered. For “authorship”, $[Z]$ represents the work creation year. Such patterns are useful for understanding the various way in which a particular relation can be expressed and can be applied to summarization or language generation tasks. Some examples of patterns include *X continues to write the novel Y*, *X known as PERSON was born in Y*, *X directed “Educando a Rita” (YEAR) and Y e.t.c.* For the evaluation, for instance of “birthDate” relation, they made an assumption that at least every Wikipedia page that talks about a person has the birth date information in the text. This penalized the system even for pages that actually did not contain a mention of the relation. For “authorship” relation, they were not able to build a baseline to evaluate the system. They calculated precision, recall and F1-measure for the patterns, the top 8 patterns for “birthDate” had F1 of 0.83, while the top 3 patterns for “dateOfDeath” had F1 of 0.58.

[Tanaka *et al.*, 2010] also extracted relational patterns from Wikipedia text, which they called query patterns. They defined a query pattern as a lexical pattern that contains a slot to be filled by an entity. The query pattern is instantiated with an entity and used to collect concepts related to that entity. For instance a query pattern *X was born in*, when used with entity *Dustin Hoffman* to fill the slot X may return the birth place of the entity from the Web using an information retrieval system. They obtained the bridging phrases between the entity mentions, domain and range, as candidate query patterns. The results for *Opera* category using 10 query patterns was an F-score of 0.29.

Lessons learned: It was also noted that quick gold-standards can be built with the help of

Wikipedia’s infoboxes to get instances of domain, relation, and range values and use them to extract sentences containing the mentions of the domain and range. The extracted sentences are used as candidates for obtaining the relational patterns. The idea of using infobox instances for getting sentences expressing the target relations for pattern extraction was borrowed when developing the gold-standard for evaluating relational pattern extraction (see chapter 3) and adapted for the target relations that this thesis deals with. This thesis also uses DBpedia resource pages to get the relations and their instances to be used in the acquisition of sentences expressing the relations and further, annotation guidelines (see Appendix) are written for building the gold-standard for testing the relational pattern extraction algorithm. There was no mention of guidelines for the gold-standards developed by the related works. Standard precision, recall and F-measure measurements are used for evaluating the automatic sentence and pattern extraction against the built gold-standard.

Chapter 3

Feasibility study: Proposed work-flow

3.1 Problem statement

As highlighted in Chapter 2, RTE has been applied in QA and uses relational patterns that depict the various ways a relation is expressed in surface form. The acquisition of patterns is thus an intrinsic aspect of entailment based QA. Every ontological relation is associated to a set of patterns. This implies that the size of the ontology plays a role because it determines the number of relations and instances that exist. This is an important factor in that more relations means more patterns to be acquired, resulting into higher possibilities of overlap and ambiguity among patterns. A high number of instances could also lead to ambiguities in typing entities. Currently, the existing state of art entailment based QA system is Qall-me, that is able to query over structured data, in a closed domain scenario, with the help of a domain ontology. First, a comparison is made in terms of the sizes of the ontology and linked data source between DBpedia, and Qall-me.

DBpedia is an example of linked data resource that is large and heterogeneous. Taking a look at Tables 2.1 and 2.4, in Chapter 2, that show the ontology sizes for DBpedia and Qall-me respectively, one notices that the main difference is in the number of properties and instances; 55 vs 1300 for number of properties and 16000 vs approx 1.7 million for number of instances. Each relation in Qall-me is associated to a set of relational patterns that express that relation. The user query is mapped to one or more patterns at run time, and the corresponding Sparql instantiated and used to retrieve the desired answer. As detailed earlier, Qall-me works with handcrafted patterns, derived from a corpus of questions. To query over linked data, the source of the corpus of questions (or sentences) from where the patterns are extracted is a factor worth studying. The study would foster open domain QA exploiting the benefits of linked data to achieve openness.

Source of corpus of questions/sentences

Qall-me uses a corpus of domain questions already annotated with named entities and corresponding relations to acquire relational patterns. Using the relationship between Wikipedia and DBpedia, sentences containing the mentions of the domain and the range can be acquired for use in relational pattern extraction. The general idea is to search in the Wikipedia text for the mentions of the range and/or domain found in the infobox or DBpedia resource page, and extract those sentences where the mentions occur. See Chapter 2, Figure 2.1 for Wikipedia structure of information redundancy both in the text and in the infobox. In exploiting this, the extracted sentences would thus form a corpus of various ways of expressing some target relation, and used to acquire the relational patterns. The issue of robustness could arise, i.e. some relations and instances in the infoboxes not found the Wikipedia text. This can be complemented by the presence of different articles within the same category.

Extraction of relational patterns

Once the corpus of sentences are acquired, the next phase is to extract the relational patterns from the sentences. What number of relational patterns are enough to cover a large ontology like DBpedia, and its ever growing nature? As mentioned earlier, in Qall-me an average of 15 patterns per relation was set for the cinema domain. The effect of this number is not known when it comes to a large ontology. Each pattern belonging to some relation R has to be minimal as per the description of minimal relational pattern (see Section 2.2). The minimal test for the relational patterns was not handled in this thesis and left for future work.

Relational patterns - Sparql correspondence

A set of relational patterns belonging to a particular relation is associated to a corresponding Sparql query for answer retrieval. Work on automatic learning of the mappings between the extracted patterns and Sparql queries has so far not been reported with respect to Qall-me. Using the minimum of 15 patterns per relation and for 1300 relations in DBpedia, an extrapolation would arrive at 19,500 patterns to cover all relations. An automatic learning of mappings between the relational patterns and Sparql queries is ideal, though this is not covered, and could be an interesting future work. For the experiment, sparql queries have been manually pre-defined and each query associated with a set of relational patterns belonging to a target relation.

3.2 Proposed Work-flow

Having outlined the problem statement, Figure 3.1 shows the work-flow of the study that was carried out to achieve the thesis' objective of studying the feasibility of using relational patterns extracted from sentences for entailment based QA as a natural language interface to linked data.

The study was divided into three modules. Module 1 was on the annotation process of building the gold-standard. To begin, a number of Wikipedia articles were sampled, according to the initial target relations that were considered for the feasibility study (see Table 3.1). The relations were selected based on their frequency of mention in DBpedia (both high and low

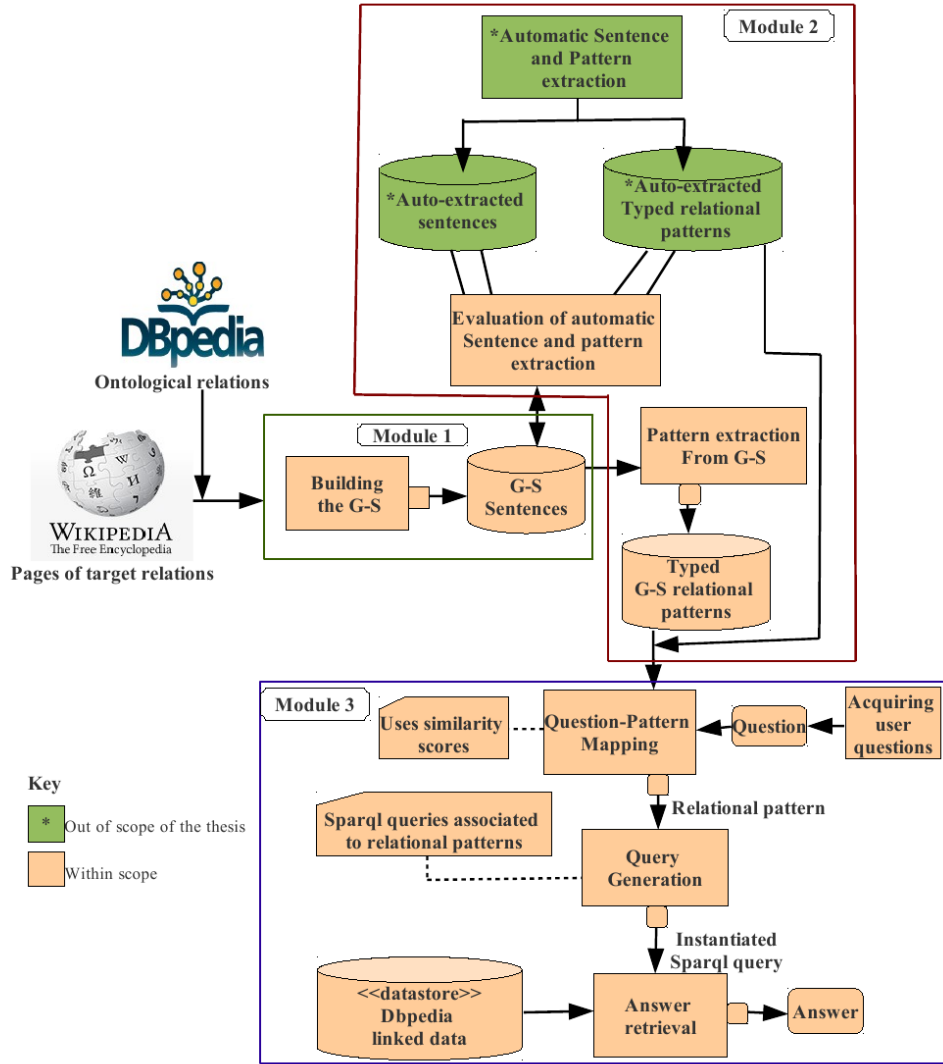


Figure 3.1 – Feasibility study work-flow

frequency relations), their level in DBpedia ontology (both those higher and those lower in the ontology) and whether the range is an entity or a literal property like date, string e.t.c (both relations with entities and properties). The sentences in the Wikipedia pages containing a mention of the domain and range of the target relations were then annotated (with tags for the domain, range and unit of words expressing the relation) (see Section 3.3) and extracted to build a gold-standard of annotated sentences. For reproducibility testing, inter-annotator agreement is carried out on the work done by the two annotators involved in the process (see Section 3.4). Section 3.5 details how the gold-standard sentences were assembled from the annotated data.

Module 2 was on relational pattern extraction. From the gold-standard, relational patterns are extracted for each of the target relations to form a resource of wikipatterns (see Section 4.1). A brief description of how the automatic sentence and pattern extraction was done is given in Section 4.2. The gold-standard sentences were used to evaluate the automatic extraction of sentences and relational patterns from Wikipedia using standard measurements of Precision,

Recall and F-measure (see Section 4.3 for evaluation results).

Module 3 was on the feasibility of using the relational patterns in entailment based Question Answering over Linked data. Sample user questions are acquired from the Web (see Section 3.7) and pattern mapping is done against the extracted patterns (from the gold-standard sentences and automatically from Wikipedia) using similarity based metrics to get the pattern that is most similar to the user question (see Section 3.8). Using Sparql associated with the patterns, the answer is retrieved from the DBpedia Sparql endpoint.

The next sections describe first module. The other modules are explained in the subsequent chapters.

3.3 Building the gold-standard: Module 1

This section presents the process of building the gold-standard, the annotation scheme and guidelines that were developed for use during the annotation process. The results of the inter-annotator agreement are also presented.

3.3.1 Annotation

Annotation is a commonly used term in computational linguistic community and is defined as 'the practice of adding interpretative *linguistic information* to a corpus' [Leech, 2005]. The term linguistic information is quite general and incorporates a wide range of annotation tasks. During annotation, one adds tags, or labels, to the corpus indicating that a particular word, phrase or sentence belongs to some predefined given category. Most researchers view tagging as a means of adding value to the original raw corpus. Annotation is both a time-consuming and laborious work, but the usefulness outweighs this. Such efforts can be beneficial to many if they are well done and documented.

There are various kinds of annotation tasks each applicable to different or complementary research areas. Some examples of annotation tasks include named entity recognition (NER), part of speech (POS) tagging, anaphora resolution, dialog act tagging, prosody e.t.c. These tasks vary in magnitude and complexity, and the results of such annotation tasks have been quite helpful in the computational linguistic community. Much still needs to be done to develop gold-standards for other specific tasks. [Artstein and Poesio, 2008, Leech, 2005] describe different NLP tasks and the kind of gold-standards that have been built to improve research in those areas. For example, Automatic content extraction (ACE) [Doddington *et al.*, 2000] for named entity recognition where the task is to identify mentions of individuals, organizations, locations, for English language. [Magnini *et al.*, 2006] developed the Italian content annotation bank for Italian named entities. For POS, some examples are the Brown corpus¹ for English and Tiger corpus² for German. Special tasks like tagging learners tags [Granger *et al.*, 2002], which consists of writings or speech produced by learners of a second language annotated with "error tags", showing the places where learners have most difficulty and the kinds of errors that

¹<http://www.comp.leeds.ac.uk/ccalas/tagsets/brown.html>

²<http://www.ims.uni-stuttgart.de/projekte/TIGER/>

they make. For dialog act we have [Carletta *et al.*, 1997], which is a dialog act annotated spoken language corpora.

3.3.2 Related annotation task

An annotation task that is related to the task that this thesis describes is done by [Hendrickx *et al.*, 2010]. They describe an annotation scheme and guidelines for semantic relations in the SemEval task-8³. Their task involved annotating a pair of nominals that are in relation with each other. The nominals (relation arguments) were only noun phrases with common noun heads. Only the semantic heads of the nominals were annotated, which were in most cases just one word. Each pair of nominals in context were to be mapped onto only one relation. They excluded instances where the relation arguments occurred in separate sentential clauses. The semantic relations included cause-effect (*those <e1>cancers</e1> were caused by radiation <e2>exposures</e2>*), content-container (*a <e1>bottle</e1> full of <e2>honey</e2> was weighted*), compound-whole (*my <e1>apartment</e1> has a large <e2>kitchen</e2>*), message-topic (*the <e1>lecture</e1> was about <e2>semantics</e2>*) among others. They described both general and relation specific guidelines. Two annotators were involved, and the inter-annotator agreement computed on the sentence level as the percentage of sentences for which the two annotations were identical, i.e. exact-match accuracies. It can be seen that annotation and building of gold-standards is a central part of computational linguistics towards promoting research in a number of research areas.

Annotation is both a time-consuming and laborious work, but such efforts can be beneficial to many if they are well done and documented. Some advantages of annotation as highlighted by [Leech, 2005] include:

- One is able to examine and extract different information from an annotated corpus e.g. from a corpus with POS tagging, one can extract the various senses of a word, cluster words that occur in similar contexts e.t.c.
- Annotation allows automatic analysis of a corpus e.g getting frequencies of a given phenomena.
- An annotated corpus is a sharable resource.
- Because its a sharable resource, every researcher might use it differently to accomplish various tasks making the annotated corpus a multifunctional resource.

To foster sharability and multi-functionality, there needs to be put in place standards for annotation. The annotation task needs to be well detailed and have explicit documentation. A couple of questions could be asked while developing the guidelines i.e. who are the annotators?, what is the annotation scheme?, what tools are used?, what guidelines are followed?, what were the reproducibility results?. If these questions are well answered, then in most cases, the documentation would be explicit enough for others to replicate the task. The annotation scheme could follow various encoding from simple underscore-label patters (`_ POS`) to XML or SGML encoding schemes. This thesis adapts the XML tagging format described in the succeeding

³<http://semeval2.fbk.eu/semeval2.php?location=tasks&area=Semantic%20relations>

subsections. The annotation guidelines explain the annotation scheme and what is acceptable in accordance to the task, and in most cases evolve as the annotation process unfolds.

3.3.3 Building the corpus

The gold standard is built from Wikipedia pages. To do this, first, the pages were preprocessed. The preprocessing included first selecting only the pages that contain an infobox and converting them into text files with one sentence per line. It was noted that some Wikipedia pages have identical titles with only a difference in the case of some character in the title. Such files have been concatenated into one file with tags marking the beginning and end of each unique file. An example is provided below. All other files represent a single wikipedia. The tagging scheme used is also shown below. When evaluating the sentences extracted by the system against the gold standard, the comparison is based on sentence identifiers thereby avoiding any noise that might have been introduced as a result of the annotation.

```
File_name= Wolfgang_Amadeus_Mozart
<page id="200 " title = "Wolfgang_Amadeus_Mozart">
<s id="200-1"> Mozart was born in 1756. </s>
<s id="200-2"> He was a great composer. </s>
</page>

<page id="201" title = "wolfgang_amadeus_Mozart">
<s id="201-1"> Mozart was born in 1989. </s>
<s id="201-2"> He is a great swimmer. </s>
</page>
```

Next, the target relations were identified and selected. Table 3.1 shows the target relations used in the experiment, and the aspects considered in selecting them i.e., the domain and range values, the level of the domain in DBpedia ontology, the range type (entity or property) and the frequency of mention of the relation in DBpedia datasets. For each relation, only those wikispaces that contain, in their corresponding DBpedia resource pages, at least one of the target relations mentioned were chosen. The DBpedia resource pages having been built by extracting data from the infoboxes in their corresponding wikispaces, reflect the contents of the wikispaces. There are 10 datasets, each for a single relation. For some relations, the number of Wikipedia pages extracted from the XML dump was less than the expected number. A total of 927 pages were annotated at the end.

3.3.4 Annotation scheme

The gold standard is built by annotating the sentences, expressing each of the target relations, with domain, range and relation tags. The annotation scheme used has been partially adopted from SemEval ⁴ Annotations. Each target sentence was annotated with the relation name, domain and range. The tags used are elaborated below:

⁴<http://semeval2.fbk.eu/semeval2.php?location=>

Relation	Domain	Ontology level of domain	Range	Range type (property or entity)	Frequency (No. of pages)
birthDate	Person	1	date	property	262741
writer	Work	1	Person	entity	112099
language	Thing	0	Language	entity	74089
numberOfStudents	University	3	non Negative Integer	property	20231
color	Thing	0	string	property	3575
gameEngine	Video game	3	thing	entity	1796
wineRegion	Grape	4	Wine region	entity	48
collection	Museum	4	String	property	27
isPartOf	Brain	2	Anatomical Structure	entity	16
launchDate	Space mission	2	date	property	15

Table 3.1 – Target relations showing aspects considered during selection

- `<e1>[DOMAIN]</e1>` - Used to tag the entity representing the domain of the relation, i.e. the main entity being talked about in the wikipedia. It represents the actual mention of the entity or its variants, or a pronoun(he, she, it e.t.c.) used in place of the entity.
- `<e2>[RANGE]</e2>` - Used to tag the range value of the relation. This is an entity or property that is in relation with the domain and is mentioned in the infobox or DBpedia resource page.
- `<rel>[RELATION]</rel>` - Used to tag the unit of words expressing the target relation.

The relation tag is optional because for some sentences, the expressed relation is implied and not explicitly expressed in surface form. The sentences below are some examples showing the annotation scheme:

- `<s id="200-1" relation ="director"> <e1>Pulp Fiction</e1> is a 1994 American crime film <rel>directed by</rel> <e2>Quentin Tarantino</e2>, who co-wrote its screenplay with Roger Avary.</s>`
- `<s id="200-2" relation ="writer"> Pulp Fiction is a 1994 American crime film directed by Quentin Tarantino, who <rel>co-wrote</rel> its <e1>screenplay</e1> with <e2>Roger Avary</e2>.</s>`
- `<s id="200-3" relation ="producer"> Tarantino and his <rel>producer</rel>, <e2>Lawrence Bender</e2>, brought the script to Jersey Films, the production company run by Danny DeVito, Michael Shamberg, and Stacey Sher.</s>`

The annotation of entities in relation has been done before, as highlighted in the related work section, but most of the works deal with finding the sentences with a mention of both domain and range, and only annotating the two in the found sentence. This thesis takes a step further

and also annotates the key word units that actually express the relation using an additional tag (`<rel></rel>`). This was the challenging part, i.e. defining what exactly is the unit of words that express the relation. The additional relation tag helps to give the precise main word(s) that actually expresses the relation, and is useful for optimizing and testing the automatic extraction algorithms.

3.3.5 Annotation guidelines

In every annotation task, there are a set of rules stating what is acceptable as per the phenomenon under study. These rules are defined in the annotation guidelines. The guidelines may evolve as the annotation process continues, capturing new rules or updating the existing rules, to be most representative of the phenomenon. The general criteria for building the annotation guidelines is presented next, while the detailed guidelines themselves are given in Appendix A.

General criteria

The methodology for creating the annotation guidelines used in building the gold-standard was an exploitative approach. There were no guidelines to start with and so they had to be written incrementally. First, there was the training phase, where the annotators were familiarized with relation extraction task, the Wikipedia dataset and the annotation scheme. Then, they annotated different documents for the target relations and met for comparison and discussion i.e. harmonizing initial differences and incrementally building the annotation guidelines based on interaction and negotiation of common solutions to controversial annotations. Once the guidelines were completed, the annotators annotated similar files using the set guidelines. The inter-annotator agreement was calculated and the differences observed used to streamline the annotation process.

For the general guidelines, this thesis adopts the definitions of an entity as given by the SemEval annotation task-8 (i.e. semantic heads of nominals) and named entity recognition⁵ (i.e. a named entity belongs to a specific class e.g. proper names). These were chosen because they are standards that have already been adopted by a wide community of researches, and are also closely related to the task at hand. Also, an entity is whatever word, compound word or phrase that is the domain or range of a relation as indicated in the infobox or DBpedia resource page.

3.4 Inter-Annotator Agreement

Every annotation task is subject to unintended errors which might be introduced in the annotation process. Thus, a measure needs to be used to assess the reliability of the annotated corpus. In their survey paper, [Artstein and Poesio, 2008] argue that reliability can be viewed in different ways. Reliability or intra-annotator agreement can be the extent to which the annotation process yields the same result over time, typically by measuring a single coder's agreement with previous work. On the other hand, inter-annotator agreement, is a measure of reproducibility, where annotators working independently can produce the same result using the same set of

⁵http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/ne_task.html

guidelines. The evaluation carried out in this thesis is based on inter-annotator agreement. If the results are similar, one can infer that the annotators have understood the annotation guidelines, and that the guidelines are adequate for the task and captures well the phenomena being studied.

Accuracy is given by the annotation guidelines where the notion of “correctness” is defined i.e. what the guidelines allow or disallow. It is expected that the agreement between the annotators would be high if their accuracy, in relation to the guidelines, is high. Inter-annotator agreement also covers consistency among annotators. Consistency denoting the fact that if the annotators work independently and are able to produce comparable results, then the annotation task has been well understood. The two common inter-annotator agreement measurements that have been adopted in computational linguistic community are the Kappa and Dice coefficient.

Kappa coefficient

In order to measure whether or not a gold-standard annotated corpus is consistent enough to be useful, it is essential to measure the reliability of the corpus using empirically derived standard methodologies. Kappa, proposed in computation linguistic literature by [Carletta, 1996], is a common agreement measure, especially in the dialog community for measuring inter-annotator agreement. The Kappa statistic is a measure of agreement between two or more annotators who each classify a number of items N into a specified set of categories C . The equation for the Kappa statistic, K is given by $\frac{Pr(A)-Pr(E)}{1-Pr(E)}$, where $Pr(A)$ is the observed measure of how much the annotators agree, while $Pr(E)$ is a measure of the expected chance agreement between the annotators calculated using the observed data. According [Carletta, 1996, Eugenio, 2000], the amount of agreement that is expected by chance mainly depends on the number and relative frequencies of the labels under consideration, and thus the reliability measurements for the label classification should be measured using the Kappa Coefficient.

There exists several proposals in literature for computing Kappa coefficient; the K statistics for Cohen (1960) and Fleis (1971). The difference is on how $Pr(E)$ is computed. [Komagata, 2002] argues that the Cohen version of calculating Kappa is more desirable than the Fleis version for a two rater-task.

This thesis annotation task is close to named entity recognition tagging problem. An annotator tags a pair of entities which are in relation, and also the key words expressing the target relation. So far in the named entity tasks, Kappa has not been used to measure inter-annotator agreement [Artstein and Poesio, 2008], due to the nature of the task and issues with Kappa when it comes to skewed data [Carletta *et al.*, 1997, Rosenberg and Binkowski, 2004] i.e. in a given corpus, only a small percentage of the words are named entities. According to [Hasler, 2008, Carletta *et al.*, 1997], first one needs to measure how well the annotators agree in the boundaries (units) to be tagged. Only after this can one measure the Kappa agreement on the tags assigned to the agreed upon boundaries.

In this thesis, exact unit, either domain, range or relation, can only be tagged with one label. For Kappa to be used to calculate the tagging agreement for each label, there should be more than one competing label assigned to each unit. [Hendrickx *et al.*, 2010] used percentage agreement as the measure of inter-annotator agreement and stated that ‘chance agreement on preselected candidates is difficult to estimate’, and thus did not present any agreements

based on Kappa. For the annotation task, the Dice coefficient was adopted instead, motivated by [Magnini *et al.*, 2006], who also used Dice for inter-annotator agreement on Italian named entities and [Hasler, 2008] who used Dice to measure agreement between spans of text before applying the Kappa statistics on the labels.

Dice coefficient

The dice measure, [Dice, 1945], is a set agreement measure defined as twice the intersection of the sets, over the union of the sets. If X and Y are two sets, then Dice D is given by:

$$D = \frac{2|X \cap Y|}{|X| + |Y|}$$

The function ranges between 0 and 1, with 1 signifying perfect agreement between the annotators. First, Dice has been used to calculate the string similarity between tagged units (extent of domain, range or relation) for the two annotators, by using the bigrams of the different strings. It has also been used to measure the sentence and tag agreement for each of the labels. This has been calculated as:

1. C - the number of common annotations i.e. both annotators have identified the same unit of the entities being tagged or the same sentence.
2. A - Number of entities (or sentences) annotated by the first annotator.
3. B - Number of entities annotated (or sentences) by the second annotator.

Therefore Dice agreement for each label is calculated as $\frac{2C}{(A+B)}$. For the string comparison, C is the number of common bigrams between the two strings, while A and B are the number of bigrams for first and second string respectively.

The procedure, results on inter-annotator agreement, and the interpretation of the achieved results are presented next. The calculation of the inter-annotator agreement was done incrementally and the final average presented.

3.4.1 Procedure

The annotation was carried out in three phases and a total of 996 files were annotated independently by the two annotators only relying on the provided annotation guidelines.

The evaluation was as follows:

1. For each annotator, there were 10 files, one file for each target relation. Each file contains the sentences that have been annotated as expressing the relation.
2. Check sentences that match
 - (a) If match
 - i. Increment sentence agreement counter
 - (b) For each sentence
 - i. For each tag (domain/range/relation), check existence
 - A. If tag exists in both annotator files, increment tag agreement counter

- B. Calculate n-gram Dice similarity measure of the two substrings enclosed by the tags.
- C. If 100% Dice similarity,
- D. Increment the unit boundary similarity counter

3. Calculate Dice for sentence agreement and for each of the tag agreements.

The same procedure was run iteratively. Table 3.2 shows the average dice value for the sentences, domain, range and relation tags.

Key :

Dice S - Dice value for sentence agreement

Dice D - Dice value for domain tag agreement

Dice R - Dice value for range tag agreement

Dice Rel - Dice value for relation tag agreement

Relation	Dice : S	Dice : D	Dice : R	Dice : Rel
birthDate	0.96	0.93	0.98	0.98
gameEngine	0.77	0.78	0.97	0.83
language	0.77	0.98	0.93	1
writer	0.84	0.93	0.90	0.89
color	0.39	0.5	0.5	0.22
numberOfStudents	0.89	0.92	0.96	0.64
wineRegion	0.63	0.90	0.83	0.61
launchDate	0.97	0.93	0.93	0.86
collection	0.6	0.93	0.36	0.64
isPartOf	0.78	0.67	0.87	0.33
Average	0.76	0.85	0.82	0.70

Table 3.2 – Average Dice similarity results

In Figure 3.2, the variations in the dice values across the relations are presented. Each column is calculated by taking the cumulative sum of the average dice values per relation (for sentence, domain, range and relation) from Table 3.2. The sum of the 4 values: the average of sentence, domain, range and relation values, gives the cumulative dice value for the each of the relation. For each relation, the least value among the four values shows the most challenging aspect of that particular relation. For instance, for “collection”, the least value is the range dice value, while for “color”, relation dice value is the least correctly depicting the challenges that were faced during the annotation process.

Figure 3.3 compares the number of extracted sentences versus the total number of pages looked at from the selected Wikipedia articles. For “numberOfStudents”, “language” and “gameEngine”, the number of extracted sentences is almost half the total number of Wikipedia articles that were looked at, while for “wineRegion”, there are many occurrences of the relation in the Wikipedia text. “color” relation has the least number of mentions of the relation in the Wikipedia text.

3.4.2 Challenges during annotation

In developing the annotation guidelines and completing the annotation task, the following were some issues encountered:

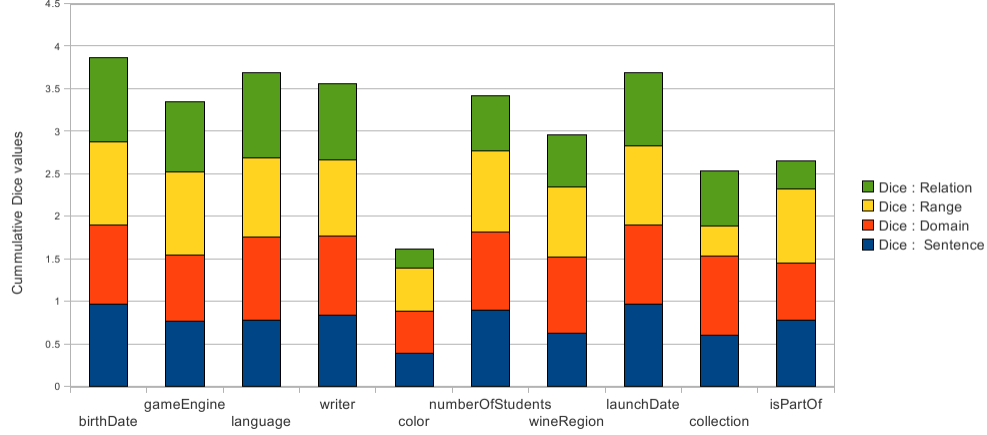


Figure 3.2 – Variations in Dice values across relations

1. Selecting a representative set of relations - first, there was the need to select a representative set of relations from the DBpedia ontology. The criteria used to select the 10 target relations were based on the relation frequencies, if the range value is an entity or property and level in the ontology.
2. No reference works - The closest work so far done in large scale is the SemEval annotation task-8 [Hendrickx *et al.*, 2010] on multi-way classification of nominals that only annotates the domain and the range in the sentence. They also do not include sentences that express more than one relation, to simplify the task. In this thesis, the unit of words best describing the relation is annotated. If a sentence expresses more than one relation, only the unit of words expressing the target relation is annotated while if a sentences expresses the same relation in various ways the sentence is replicated each relation annotated independently in separate sentences (see Appendix for the details of the annotation).
3. Sentences expressing multiple relations - As seen from number 2 above, sentences expressing more than one relation are annotated, but only the target relation is annotated. This increased the difficulty of the task.
4. Diversity of the relations - From the 10 relations were chosen initially, it was noted that each relation possessed its own unique rules that sometimes could not apply across all relations. To capture this, the guidelines included sections with relation specific guidelines.
5. Domain and range as lists - Some relations had the multiply values for either the domain or range. To avoid having confusing tags, only one instance of the domain or range was annotated in a sentence at a time, and the sentence replicated the same number of times as the list.
6. Implicit relations - For some sentences, the target relation was only implied in the sentence, and was hard to detect the actual unit of words expressing that relation. For instance, in the sentence `<s relation="writer">The <e1>film</e1> was based on <e2>his</e2> novel of the same name, and was released in the US in 2005 as Pulse.</s>`. Where *his*

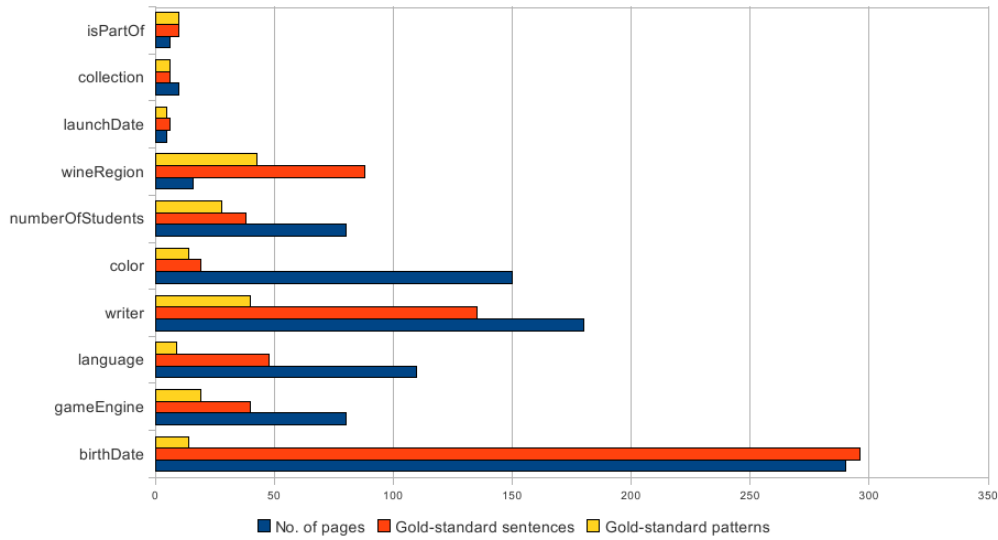


Figure 3.3 – Number of annotated pages vs number of gold-standard sentences and patterns

refers to the name of the author which was mentioned in the previous sentences. The “writer” relation is implied, in that because the film was based on a book written by someone, it can be said that the film writer is the author of the book.

7. Annotating the relation - From the inter-annotator agreement, it was noted that the agreement value was lower when it comes to the boundary of the unit of words that express the relation. A trade-off was made between very strict forms, i.e. including adverbs or auxiliaries, and generic forms, including only the main verb or preposition expressing the relation. The annotation leaned towards tagging only the main verb and/or preposition as the unit of words expressing the relation.
8. Need for world knowledge - For some relations, world knowledge was needed in order to correctly annotate the sentences. For instance, “wineRegion” relation also included sub or super regions with respect to the actual wine region mentioned in the text. For instance, if the infobox value is *France*, while in the sentence *Loire valley* is indicated as the wine region, one has to know that *Loire valley* is in *France* and thus the sentence is correct and indeed has a mention of the range. The same was for “isPartOf” relation that knowledge on anatomical structures would be useful in order to correctly say that some part of the brain is part of another anatomical structure.

3.4.3 Interpreting inter-annotator results

As mentioned earlier, a dice value of 1 shows perfect agreement between the annotators. From Table 3.2, for sentences that the annotators agreed that a particular relation existed, the annotators had higher agreement in the range tag and lower in the relation tag. More often than not, the range value is clear because it is indicated in the infobox or DBpedia resource page while the relation sometimes is not obvious, leading to differences in unit boundaries between the annotators. For instance in the sentence “The <e1>number of students</e1> varies year

to year from <e2>750-1000</e2>, currently home to grades 7 to 12.</s>”, the exact unit of words that expresses the relation is not obvious.

It can also be noted that for some relations, like “color”, there are very few sentences extracted that represent the relation, while for “wineRegion”, there are many sentences that talk about the relation. The next section summarizes the issues and challenges that pertain to each of the target relations.

Conclusions drawn from the annotation process

It was noted that each relation is unique and possesses different text styles and sentence structures that make the tagging process challenging. The annotation guidelines are meant to be generic and conclusive for all the relations but in real sense, one would need relation specific guidelines to handle the special cases. It was challenging to stick to the guidelines since some particular generic rule suitable for one relation would not apply across all relations. For this reason, a section was added in the guidelines, for relation specific guidelines. This improved the average agreement results, especially for relations with fewer sentences like “collection” and “launchDate”.

From the result, we see that the inter-annotator agreement is generally higher, for some relations such as “birthDate”, “writer”, “launchDate” and “language”, than the others. This can be attributed to the simplicity in which the relations are expressed in text. Very few sentences, if any, are ambiguous when it comes to these relations.

For “color” relation, the number of sentences expressing the relations were few, and therefore the inter-annotator agreement would not be conclusively calculated, as shown in the third iteration with all having zero dice value. Also, the relation is ambiguous, in that, what the color is referring to is not clear, i.e, the domain of the relation. For instance, for the wikipedia *39th Infantry Regiment (United States)*, the value of color relation in the infobox are blue and white, but what exactly is the color referring to? Is it the shield, uniform, badge, flag or insignia? It is also the case that most pages do not have the color relation mentioned in the infobox but only in the DBpedia resource page. There are also some errors in DBpedia where the color value is not correct e.g. containing the Motto, or the pixel (ex.175px) value of some image rather than a color value.

For “numberOfStudents”, the major challenge is that the number of students value in the infobox is in most cases given as an approximation in the Wikipedia text. For instance using statements like *...just over <e2>one thousand</e2> pupils.</s>* while in the infobox having the actual value of *1,040*. Also, the institution is sometimes implied and not mentioned explicitly in the sentence. Statements like *There are...*, are taken as expressing the relation, while *The number of students...* are taken as expressing the domain. The annotators mainly differ in the unit of text to be annotated as expressing the relation.

The most challenging relation was “wineRegion” due to the various ways that the range and domain value could be expressed in the text. For instance, the specific region mentioned in the infobox is *Loire Valley*, while in the sentence, the wine region mentioned is *France* or vice Visa. The world knowledge that *Loire Valley* is in *France* is therefore needed. The main dissimilarity between the annotators was the unit of text to mark as expressing the relation. The same problem of identifying the unit expressing the relation was with “gameEngine” relation. These

relations are not as obvious as for relations like “birthDate” or “writer”.

“Collection” and “isPartOf” relations were introduced after the initial discussion phase of building the annotation guidelines. The guidelines were not conclusive enough to be used for the annotation and relation specific guidelines for were added in the annotation guidelines. For “collection”, the range data type is String, but the actual value is ambiguous. Sometimes its a numerical value like *135,000 artifacts* , or a description of objects like *World War I aircraft and antique automobiles* or just a generic term like *children’s books*. Often the range value in sentence and that in the infobox are not exactly the same. This means that the range value depends on what has been assigned in the infobox and difficult to generalize across Wikipedia pages. The ambiguity in the range values leads to low similarity between the annotators. “isPartOf” relation is complex and needs domain knowledge in order to understand the domain jargon to conclusively say that a section of the brain is part of some anatomical structure. For this reason, only the sentences mentioning the exact value as in the infobox or DBpedia resource page were annotated.

The average inter-annotator agreement results indicate relatively high levels of agreement between the annotators, and one could say that the annotation guidelines and the built gold-standard (see Section 3.5 on how the final gold-standard sentences were chosen) are acceptable. The gold-standard sentences were used to evaluate the automatic sentence and pattern extraction and the evaluation results are presented in the next chapter.

3.5 Building the gold-standard sentences

As indicated earlier, both annotators independently annotated all the Wikipedia pages used in the annotation phase. Based on the inter-annotator agreement results (see section 3.3), it was noted that the annotators achieved acceptable agreement results most of the time, and thus the created resource is reliable.

To build the final gold-standard, the sentences that were agreed upon (i.e. both annotators agreed on the domain, range and relation tags) by both annotators were added to the gold-standard. For the sentences that were annotated by both annotators but in which they differed in the span of the labels used, the annotators reached a consensus and added the agreed sentences to the gold-standard. For the rest of the sentences that were annotated by either of them, they still had to reach a consensus on whether or not to include those sentences on the gold-standard. Table 4.1, column 3, shows the number of sentences that were finally included in the gold-standard.

3.6 Chapter Summary

In this chapter, the problem statement was introduced and how the first module of the proposed work-flow was conducted. The main points are summarized below:

1. The problem statement was on evaluating relational patterns extracted from Wikipedia and on carrying out a feasibility study on the use of these relational patterns, rather than those extracted from a corpus of questions, in entailment based QA to query over linked data.

2. The annotation scheme was defined and annotation guidelines were developed to guide the annotation process of building gold-standard sentences from Wikipedia expressing different target relations, with the domain and range of the relations annotated. 10 target relations were selected and 927 Wikipedia pages (that had a mention of the target relation in the infobox) were annotated.
3. The inter-annotator agreement for the annotation process using Dice coefficient measurement. The average results were 0.76 for sentence agreement, 0.85 for domain tag agreement, 0.82 for range tag agreement and 0.70 for relation tag agreement (see Table 3.2).

Chapter 4

Relation extraction: Module 2

From Chapter 3, the gold-standard sentences have been built. These sentences were used to evaluate the automatic sentence and pattern extraction from Wikipedia. From them, gold-standard relational patterns were also extracted. The details are presented next.

4.1 Extracting patterns from gold-standard sentences

In accordance to the aim of the study of testing the applicability of the patterns in using entailment based QA systems over linked data, relational patterns were extracted from the annotated sentences for each target relation. A pattern is an ordered set of domain, relation and range tags; where the relationship $relation(domain, range)$ holds. Each pattern represents only one relation. A relational pattern should be lexically unique within the set of patterns for a particular target relation. Table 4.1, column 4 gives the number of patterns that were extracted for each relation.

Relation	No. of annotated pages	No. of gold-standard sentences	No. of gold-standard relational patterns
birthDate	290	296	14
gameEngine	80	40	18
language	110	48	9
writer	180	135	37
color	150	19	14
numberOfStudents	80	38	28
wineRegion	16	88	43
launchDate	5	6	5
collection	10	6	6
isPartOf	6	10	10

Table 4.1 – Number of pages, sentences and extracted relational patterns

Table 4.2 shows some examples of patterns extracted from the annotated sentences.

As noted in the work-flow diagram (see Figure 3.1), the automatic sentence and pattern extraction are part of the overall goal of pattern acquisition process though not in the scope of

“birthDate”	“collection”	“wineRegion”
<ul style="list-style-type: none"> ➤ [PERSON] born [DATE] ➤ [PERSON] born on [DATE] ➤ [PERSON] born in [DATE] ➤ born [PERSON] on [DATE] 	<ul style="list-style-type: none"> ➤ [MUSEUM] centre for [STRING] ➤ [MUSEUM] collection of [STRING] ➤ [MUSEUM] consists of [STRING] ➤ [MUSEUM] dedicated to [STRING] ➤ [MUSEUM] features [STRING] 	<ul style="list-style-type: none"> ➤ [GRAPE] planted throughout [REGION] ➤ [GRAPE] popular in [REGION] ➤ [GRAPE] presence in [REGION] ➤ [GRAPE] prevalent in [REGION] ➤ [GRAPE] produced in [REGION]

Table 4.2 – Examples of patterns from gold-standard

this thesis. A brief description of the main ideas are given in the next section.

4.2 Automatic extraction

The work on automatic sentence and pattern extraction was carried out by [Mahedra, 2011]. The gold-standard sentences were used to evaluate the automatic sentence and relational pattern extraction (see Section 4.3 for results). The main ideas of how the extraction was carried out are highlighted below.

4.2.1 Sentence extraction

The sentence extraction module is aimed at acquiring sentences that could be used to extract potential relational patterns. A sentence is considered a candidate sentence for pattern extraction if it has a mention of both the domain and range of the relation. String matching algorithm is used for the sentence extraction. String matching aims at finding all the occurrences of a searched string in a given text data. For instance in the example triple below, *Forrest Gump* is the domain and *Winston Groom* is the range of the “writer” relation.

```
<http://dbpedia.org/resource/Forrest_Gump>
<http://dbpedia.org/ontology/writer>
<http://dbpedia.org/resource/Winston_Groom>
```

The sentence “<s id="41528-1"><e1>*Forrest Gump*</e1> is a 1994 American comedy-drama film based on the 1986 novel of the same name by <e2>*Winston Groom*</e2>.</s> ” matches both the domain and the range values.

The challenges in sentence extraction are brought about by the presence of linguistic phenomena such as pronouns e.g. he, she, it instead of the proper names; name and date variants e.g. name variants like Gates, Bill Gates, Bill e.t.c. and date variants like 15-06-2011, 15th June 2011 e.t.c; hypernyms like the C.E.O , the school, the museum e.t.c. instead of the actual name of the person, school or museum respectively.

For automatic extraction, different configurations were needed to optimize the identification of mentions in sentences automatically. Different configurations worked best for different rela-

tions depending on the types of the domain and range of that relation. Three string matching algorithms were used to optimize the sentence extraction:

1. Exact string matching: This involved searching the precise occurrence of domain or range in the sentence. Using the previous example, exact matches of *Forrest Gump* and *Winston Groom*, are checked, and the sentence with both matches retrieved.
2. Name matching: For some triples found in DBpedia, the domain or range name included additional disambiguation information about the entity. For example, entity FILM:*Gone with the Wind (Film)* and entity PERSON:*John Kay (Musician)*. The name matching string search technique omitted the disambiguation information (*Film*) and (*Musician*) while searching for the entities.
3. Substring matching: This is a kind of approximate string matching technique. Rather than matching the whole entity string, substrings of the entity are searched instead. For instance, the entity PERSON:*George Washington*, the substrings *George* and *Washington* are separately searched in the text.

Table 4.4, column 2 shows the number of sentences that were extracted to form part of the training set for automatic relational pattern extraction.

4.2.2 Pattern extraction

Once the sentences for each relation are extracted, the pattern sets are generated using suffix trees¹. A suffix tree is a data structure that shows the suffixes of a given string. It allows for quick manipulation of various string operations, like in this case, pattern extraction. For a phrase to be considered a pattern, it should include the place holders for the domain and range included in the phrase, as per the examples below we have [PERSON] and [DATE] respectively. Two features, DEPTH and SIZE, were used to acquire the patterns.

DEPTH refers to the level of the node in the tree while SIZE refers to the number of leaves a particular node has. A leaf is a node that has no children.

For example, take the following sentences:

1. <e1>Alex</e1> was born on <e2>July, 17, 1984</e2> in Ottawa, Canada.
2. <e1>Jimmy</e1> (<e2>25-10-1949</e2> - 13 12 1999) was a talented painter in Albania.
3. <e1>Marina</e1> (<e2>Dec, 18, 1889</e2> - Nov, 19 1940) is the second child of Brian Smith.
4. <e1>Josephine</e1> was born in Pennsylvania on 3 May 1990.
5. <e1>Sabrina</e1>, the daughter of Sir Philip, was born on <e2>August, 14, 1989</e2> in small town near Birmingham.

The below suffix tree would be used to represent these sentences. Each tabbed new line represents the child or children of the previous line.

¹Wikipedia/Suffix_tree

\succ [PERSON]
 – was born
 * on [DATE] in Ottawa, Canada.
 · in Pennsylvania on 3 May 1990.
 * ([DATE] -
 · 13 12 1999) was a talented painter in Albania.
 · Nov, 19 1940) is the second child of Brian Smith.
 * , the daughter of Sir Philip, was born on [DATE] in small town near Birmingham.

Based on the 5 sentences above, 3 pattern candidates could be extracted:

Example patterns:

Pattern1: [PERSON] was born [DATE] in Ottawa, Canada.

Pattern2: [PERSON] (DATE -

Pattern3: [PERSON] , the daughter of Sir Philip, was born on [DATE] in small town near Birmingham.

Pattern1 has DEPTH 2 because the pattern is completed in a node that is the grand child of root node and SIZE 1 because the pattern is completed at a leaf node, and thus has no children. Pattern2 has DEPTH 1 because the pattern is completed on a node that is the child of the root node and SIZE 2 because the node has two leaf nodes. Likewise, Pattern 3 has DEPTH 1 and SIZE 1.

Patterns whose SIZE is 1 are not good pattern candidates because they represent a complete sentence with no abstraction. Patterns with SIZE greater or equal to 2 are better candidates. Table 4.3 shows some examples of pattern candidates automatically extracted from the suffix trees.

“numberOfStudents”	“wineRegion”	“gameEngine”
\succ [UNIVERSITY] has around [NONNEGATIVEINTEGER]	\succ In [WINEREGION], [GRAPE]	\succ [VIDEOGAME] is based on the [THING]
\succ [UNIVERSITY] has approximately [NONNEGATIVEINTEGER] pupils	\succ [WINEREGION], [GRAPE] tends to produce	\succ [VIDEOGAME] uses the [THING] engine
\succ [UNIVERSITY] currently has around [NONNEGATIVEINTEGER]	\succ [WINEREGION], [GRAPE] is	\succ [VIDEOGAME] is programmed in [THING] and runs in most modern browsers (such as
\succ [UNIVERSITY] serves about [NONNEGATIVEINTEGER] students	\succ [GRAPE] in [WINEREGION]	\succ [VIDEOGAME] and its [THING]

Table 4.3 – Examples of automatically extracted pattern candidates

Table 4.4, column 3 shows the number of relational patterns that were automatically extracted from the sentences used as the training set. It can be noted that the lesser the number

of training sentences, the fewer the number of extracted relational patterns.

Relation	No. of training sentences	No. of Patterns
birthDate	50963	14302
language	10745	767
writer	10261	682
color	1114	35
number OfStudents	662	99
gameEngine	626	32
wineRegion	162	7
collection	32	1
isPartOf	20	3
launchDate	4	0

Table 4.4 – Number of training sentences and extracted patterns per relation

Once the sentences and relational patterns were extracted, the gold-standard sentences (see Section 3.5) were used to evaluate them. The results are presented next.

4.3 Automatic extraction evaluation

The results of the automatic sentence extraction are presented next, followed by those for the automatic relational patterns extraction.

4.3.1 Sentence extraction evaluation

Each sentence had a unique ID, making the sentence evaluation a matter of comparing the sentence IDs in the automatic sentence extraction corpus with those in the gold-standard. The standard Precision, Recall and F-measure were used to evaluate the automatic sentence extraction algorithm.

Precision is the measure of the portion of the true positive sentences (sentences extracted by the system that are also in the gold-standard) against all positive sentences extracted by the system. Formally, for sentences S , α is the number of sentences in the gold-standard, β'' is the total number of sentences extracted by the system and $\beta' = \{S | S \in \beta'' \wedge S \in \alpha\}$ i.e. the total number of sentences present both in the automatically extracted corpus and in the gold-standard. Therefore Precision P is calculated as $\frac{\beta'}{\beta''}$.

Recall is the measure of the percentage of the true positive sentences that are extracted by the system against all sentences in the gold-standard. Therefore Recall R is calculated as $\frac{\beta'}{\alpha}$.

F-measure is the measure of accuracy that considers both Precision and Recall. It is the weighted average of the two. For the experiment, the F_1 score was used and is calculated as $2 \cdot \frac{P \cdot R}{P + R}$.

Different configurations (based on the string matching technique described in section 4.2) were used to optimize the sentence extraction for each relation. Table 4.5 shows the results of the best configuration with the highest F-measure for each relation.

The relations with low F-measure score e.g. “color”, “collection” represent those relations that are most difficult to extract. As was noted during the building of the gold-standard, it was challenging to detect the domain value that “color” is referring to, and the range value

	birth Date	game En- gine	language	writer	color	number Of Students	wine Re- gion	launch Date	collection	is PartOf
P	0.99	0.28	0.4	0.59	0.04	0.67	0.4	1	0.14	0.55
R	0.84	0.33	0.4	0.44	0.21	0.21	0.32	0.43	0.3	0.81
F₁	0.91	0.3	0.4	0.5	0.06	0.32	0.35	0.6	0.19	0.65

Table 4.5 – Evaluation results for automatic sentence extraction

that “collection” relation is referring to. This has contributed to the difficulty in automatically extracting these sentences.

4.3.2 Pattern extraction evaluation

For evaluation, the ability of a set of patterns belonging to some relation R to match sentences for the same relation R among all sentences in the gold-standard Γ was measured. This was to highlight on the capability of the patterns to answer binary queries for the target relations.

Let $\beta|\beta \in R$, where β has a fixed part, and a variable part representing the place holders for the domain and range of the relation. For instance in the pattern $[WORK]$ written by $[PERSON]$ for “writer” relation, *written by* is the fixed part and $[WORK]$ and $[PERSON]$ are the place holders for the domain and range respectively. Let $\alpha|\alpha \in \Gamma$, where α is the set of gold-standard sentences belonging to some relation R . Each β is transformed into a regular expression and is matched against all sentences in gold-standard Γ . Let α' be the set of sentences that have matched by at least one pattern $\beta \in R$.

Precision is calculated as the portion of those sentences matched by the patterns of relation R that are also in the gold-standard sentences for relation R over all sentences extracted by the system i.e. $\frac{\alpha \cap \alpha'}{\alpha}$.

Recall is calculated as the portion of those sentences that have been matched by the patterns of relation R that are also in the gold-standard sentences for relation R over all sentences belonging to the gold-standard sentences α of that relation i.e. $\frac{\alpha \cap \alpha'}{\alpha}$.

F-measure is the measure of accuracy that considers both Precision and Recall. It is the weighted average of the two. For the experiment, the F_1 score was used and is calculated as $2 \cdot \frac{P \cdot R}{P + R}$.

The results of the automatic pattern extraction are given in Table 4.6. The highest F1-measure is achieved by patterns from “birthDate” relation, and none of the patterns in “collection” or “isPartOf” matched a sentence in the gold-standard.

The patterns automatically extracted from Wikipedia training sentences, and those extracted from the gold-standard sentences were separately used for testing the question-pattern mapping. The details of the question-pattern mapping are presented in Chapter 5.

4.4 Chapter Summary

The main points are summarized below:

1. Relational patterns were extracted from the gold-standard sentences and there is a set of relational patterns for each target relation.

Relation	No. of Pat-terns	No. of Patterns that have matched at least 1 correct sentence for target relation	Precision	Recall	F1-Measure
birthDate	14302	146	0.73	0.25	0.37
language	767	24	0.23	0.46	0.31
writer	682	36	0.49	0.19	0.28
color	35	1	0.2	0.16	0.18
gameEngine	32	4	0.17	0.12	0.14
wineRegion	7	5	0.23	0.1	0.14
number OfStudents	99	5	0.15	0.11	0.12
collection	1	0	0	0	0
isPartOf	3	0	0	0	0
launchDate	0	0	0	0	0

Table 4.6 – Evaluation results for automatic pattern extraction

2. The gold-standard sentences were used to evaluate the automatic sentence and relational pattern extraction from Wikipedia.
3. Evaluation of the automatic sentence and pattern extraction using the gold-standard sentences. For both automatic sentence and pattern evaluations, “birthDate” relation had the highest f1-measure of 0.91 and 0.37 respectively (see Tables 4.5 and 4.6).

Chapter 5

Question Answering: Module 3

From module 1 (see Chapter 3), the gold-standard was developed and from module 2 (see Chapter 4), the gold-standard was used to evaluate the automatic sentence and relational pattern extraction. With the relational patterns in place, they were used to evaluate the feasibility of applying them in entailment based QA over Linked data. First, how sample user questions were acquired from the web is presented, followed by the simulation of entailment based QA using similarity metrics.

5.1 Acquiring user questions

In order to test the applicability of the relational patterns, sample questions asking about the range of the target relations were collected. For the experiment, only questions asking about the range were considered. This does not limit the use of the patterns, because in principle they could be used for questions asking about either the domain or range of a relation.

The sample questions were extracted from Answers.com¹. In total there were 127 questions collected for all the target relations. The number of test questions are comparable to those used in Qall-me evaluation (see Table 2.5). The major issue is the complexity of the questions (particularly on the number of conjunctive queries which are extracted from the question). If a user question expresses more than one relation, one could derive conjunctive queries which when joined together, express the same relations in the original user question. This is a relevant parameter because different parts of the question would map onto different relational patterns and therefore, it would be necessary to aggregate the queries for answer retrieval.

For instance, the question *On what date was [PERSON:"Darwin Oliva"] born and when did he die?* expresses both "birthDate" and "deathDate" relations and two conjunctive queries could be derived *On what date was [PERSON:"Darwin Oliva"] born?* and *when did [PERSON:"Darwin Oliva"] die?* each of which would map to different relational patterns. For the experiments, only questions expressing one relation at a time were considered.

For this study, only questions containing an entity mentioned in Wikipedia were used in the user query. In each question, the entity in focus was highlighted and its type indicated. The

¹<http://www.answers.com/>

types of the entities correspond to those found in DBpedia ontology. The examples below show some sample questions and how the types were indicated.

Example questions:

“birthDate”

On what date was [PERSON:"Darwin Oliva"] born?

When was [PERSON:"Darwin Oliva"] born?

“collection”

What features in [MUSEUM:"British Museum"] museum?

What is housed in the [MUSEUM:"British Museum"]?

“gameEngine”

Which game engine is the [VIDEOGAME:"Dota 2"] built on?

What is the engine for [VIDEOGAME:"Dota 2"]?

5.2 Question-Pattern mapping

Measuring string similarity between two strings is the task of finding a pattern string that is closest in similarity (whether lexical or semantic) to the string at hand. This means finding the pattern, from the set of patterns, whose meaning is closest (based on similarity scores) to the user question i.e. the meaning of the pattern is textually entailed by the meaning of the question or part of the question. Textual entailment works on lexical-syntactic level to assuage the requirement of deep NLP tools for question processing and understanding. In the entailment based QA paradigm, (see Figure 2.6), pattern mapping is checked at runtime between the user question and the set of relational patterns, and the pattern(s) with the highest similarity score with respect to the user query is selected. The corresponding Sparql is then instantiated for answer retrieval. Before the mapping is done, the entity in the user question is replaced by its type, and the typed question is what is mapped to the relational patterns.

For instance, the question: *On what date was "Darwin Oliva" born?*

is transformed to: *On what date was [PERSON] born?*

The pattern *[PERSON] born [DATE]* is selected as the pattern that is most similar (i.e. has the highest similarity score) to the user question.

For the experiment, Smith-Waterman [Smith and Waterman, 1981] metric was used to simulate textual entailment by finding the pattern that is most similar to the user query. Generally, the user question is longer than the relational patterns. The similarity metric is optimized for handling strings with unequal lengths and maximizing the similarity score based on common sub-sequences. The SimMetrics² library that implements the string comparison metric is used for the experiment. Smith-Waterman algorithm is expound below.

Smith-Waterman

Smith-Waterman [Smith and Waterman, 1981] algorithm works well in identifying similar regions between two strings. It was first applied to protein strings in the field of biology. According to the algorithm, the longest common sub-sequence maximizes the similarity measure. The

²<http://staffwww.dcs.shef.ac.uk/people/sam.chapman@k-now.co.uk/simmetrics.html>

algorithm compares segments of all possible lengths and optimizes the similarity measure. It is guaranteed to find the optimal local alignment with respect to the scoring system being used (which includes the substitution matrix and the gap-scoring scheme). The string edit operations include insertion, deletion and substitution. A scoring matrix is computed for all possible matches and edit operations. For each character match, a positive contribution is assigned to the score, while for each edit operation, a penalty is assigned. At the end, the high metric value represents high similarity between the strings being compared.

The algorithm (as adapted from Wikipedia³) is explained below:

A matrix H is built as follows:

$$H(i, 0) = 0, 0 \leq i \leq m$$

$$H(j, 0) = 0, 0 \leq j \leq n$$

if $a_i = b_j w(a_i, b_j) = w(\text{match})$ or if $a_i \neq b_j w(a_i, b_j) = w(\text{mismatch})$

$$H(i, j) = \max \begin{cases} 0 \\ H(i-1, j-1) + w(a_i, b_j) & \text{Match/Mismatch} \\ H(i-1, j) + w(a_i, -) & \text{Deletion} \\ H(i, j-1) + w(-, b_j) & \text{Insertion} \end{cases}, 1 \leq i \leq m, 1 \leq j \leq n$$

Where:

> a, b = Strings over the Alphabet Σ

> m = length(a)

> n = length(b)

> $H(i, j)$ - is the maximum Similarity-Score between a suffix of $a[1...i]$ and a suffix of $b[1...j]$

> $w(c, d), c, d \in \Sigma \cup \{-'\}, '-'$ is the gap-scoring scheme

This metric is suitable for the comparison between the user query and the set of patterns for each relation because it is usually the case that segments of the user query, rather than the whole query, are similar to some pattern or segments of a pattern, and hence maximizing the overall score with similar segments seems a better similarity measure to use for this task.

5.3 Question-Pattern mapping evaluation

The Question-Pattern mapping evaluation is aimed at getting the number of sentences that were mapped correctly to a matching pattern, and the correct sparql query instantiated i.e. mapped to the patterns that actually represent the relation expressed in the user question. This can be explained by the fact that once a question is mapped onto the correct pattern, the corresponding sparql query would be instantiated with the entity in the question and used to retrieve the desired answer. The evaluation of pattern mapping module was done twice, once using the gold-standard relational patterns and a second time using auto-extracted relational patterns. In total there were 127 questions, 90% of which were correctly mapped when using

³http://en.Wikipedia.org/wiki/Smith%E2%80%93Waterman_algorithm

the gold-standard patterns while 79% were correctly mapped when using the auto-extracted patterns. Table 5.1 shows for each relation the number of user questions, and those correctly or incorrectly mapped.

Relation	No. of questions	Questions correctly mapped		Questions incorrectly mapped	
		Gold-standard patterns	Auto-extracted patterns	Gold-standard patterns	Auto-extracted patterns
gameEngine	13	13	13	0	0
launchDate	16	16	0	0	16
numberOfStudents	14	14	14	0	0
wineRegion	11	11	11	0	0
writer	22	22	22	0	0
color	14	13	12	1	2
language	10	9	10	1	0
collection	7	6	7	1	0
birthDate	12	7	9	5	3
isPartOf	8	3	2	5	6

Table 5.1 – Questions correctly vs those incorrectly mapped

For some relations, for instance “language” and “birthdate”, there are more questions correctly mapped while using the auto-extracted patterns. This could be due to the fact that there are more patterns automatically extracted, as compared to those extracted from the gold-standard, and thus more variabilities.

Figure 5.1 depicts the pattern mapping results showing the percentage of the questions that were correctly mapped using either the gold-standard or the auto-extracted patterns.

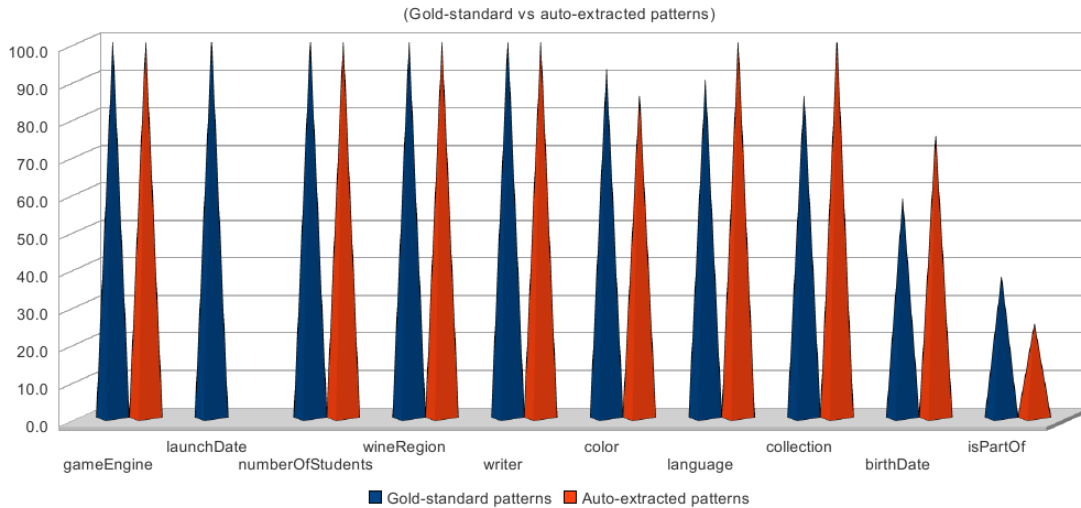


Figure 5.1 – Results showing percentage of correctly mapped questions per relation

It can be noted that by using the simple similarity metrics, Smith-Waterman algorithm, it is possible to still get a high percentage of questions being mapped to the correct relational

pattern. Some questions are still mapped to the wrong patterns. Examples of wrong mappings include:

- Question: *What colors are used in the [THING] logo?* is wrongly mapped to *[LANGUAGE] used in [THING]* which is a pattern for “language” relation, probably because of the presence of *[THING]* in both.
- Question: *What are the dialects of [THING] found in the United States?* is wrongly mapped to *[STRING] [THING]* which is a pattern for “language” relation, probably because the pattern is too general.

Better string mapping and entailment algorithms that can take additional features to guide the pattern mapping could be used to improve the results.

5.4 Query generation and Answer retrieval

For a set of relational patterns for a particular relation there is a Sparql query associated with that set. The sparql queries were manually written and associated to each set of patterns belonging to a particular relation. Given a relational pattern that has been mapped to a user question, the corresponding query for that pattern involves the instantiation and binding of the variable in the sparql query with the entity in the question.

From the previous example question, *On what date was "Darwin Oliva" born?* and the selected pattern *[PERSON] born [DATE]*, the below sparql query, associated to the pattern, is instantiated with *"Darwin Oliva"*.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?birthDate WHERE {?person rdf:type dbo:Person .
?person foaf:name ?name .
?person dbo:birthDate ?birthDate .
FILTER (?name="Darwin Oliva"@en) }
```

The instantiated sparql query is then sent to the DBpedia endpoint to retrieve the answer. The answer is embedded in an xml answer object that can be visualized back to the user in an appropriate form. The xml answer object returned is shown below.

```
<sparql xmlns="http://www.w3.org/2005/sparql-results#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3.org/2001/sw/DataAccess/rf1/result2.xsd">
<head> <variable name="birthDate"/> </head>
<results distinct="false" ordered="true">
<result> <binding name="birthDate">
<literal datatype="http://www.w3.org/2001/XMLSchema#date">1989-03-21
</literal> </binding> </result> </results> </sparql>
```

5.5 Chapter Summary

In this chapter, the feasibility of using relational patterns in entailment based QA was presented. The main ideas are summarized below:

1. Possible user questions were acquired from the web, 127 in total, and Smith-Waterman algorithm was used to map the user question to the relational pattern that was most similar to the question in order to find out the relations being expressed in the user question.
2. Evaluation of the question-pattern mapping process by using (separately) patterns extracted from the gold-standard sentences, then those automatically extracted from Wikipedia pages. 79% of the questions were correctly mapped to the matching pattern when using the auto-extracted patterns and 90% of the questions were correctly mapped when using the gold-standard patterns.
3. Each set of relational patterns had a corresponding sparql query, and once the relational pattern that best maps to a user question was found, the associated sparql query was instantiated with the entity in the question and sent to DBpedia sparql endpoint for answer retrieval.

Chapter 6

Conclusion and further research

6.1 Summary of work

Question Answering is a task of computers answering arbitrary questions posed in natural language and providing the correct answers back quickly and succinctly. The current QA trend is moving towards QA over linked data. The objective of this thesis has been to evaluate relational patterns extracted from Wikipedia and to carry out a feasibility study on the use of these relational patterns, rather than those extracted from a corpus of questions, in entailment based QA over linked data. Entailment based QA uses Recognizing Textual Entailment (RTE) paradigm, an example system being Qall-me, and relies on relational patterns that represent the different ways in which a given relation (or predicate –in Linked data model–) can be expressed in surface form.

To achieve this objective, Wikipedia was used as the source of free text from where the relational patterns were extracted and DBpedia was used as an example linked data resource. This is because DBpedia was built by extracting structured information from Wikipedia, especially from the infoboxes. Wikipedia has redundancy of information i.e., information represented structurally in the infobox and as free text (see Figure 2.1). Therefore, one is able to get range values of relations from the infoboxes or from the DBpedia resource pages. 10 relations, from DBpedia ontology, were used for the experiment. To acquire sentences expressing the chosen relations, the domain and range values were looked up (from the infoboxes) in their corresponding Wikipedia text and only those sentences with a mention of both the domain and range were extracted as the sentences expressing that particular relation. These sentences were annotated with tags for the unit of words expressing the target relation, tags for the domain values and tags for the range values. With this, a gold-standard of annotated sentences was created. The results of the inter-annotator agreement show a high level of agreement and consistency, with respect to the built annotation guidelines, between the annotators. The average results were 0.76 for sentence agreement, 0.85 for domain tag agreement, 0.82 for range tag agreement and 0.70 for relation tag agreement (see Table 3.2).

From the gold-standard sentences, gold-standard relational patterns were extracted i.e. each relational pattern is an ordered set of domain, range and relation tags found in an annotated sentence. An average of 18 patterns per relation were extracted, with the lowest being 5 patterns

for “launchDate” relation and the highest being 43 patterns for “wineRegion” relation (see Table 4.1). The gold-standard sentences were used to evaluate the automatically extracted sentences and relational patterns from Wikipedia and the results are quite promising. For both automatic sentence and pattern evaluations, “birthDate” relation had the highest f1-measure of 0.91 and 0.37 respectively (see Tables 4.5 and 4.6).

Sample uses questions expressing the target relations were also acquired from the Web. The gold-standard and auto-extracted relational patterns were used (separately) to evaluate the question-pattern mapping module using questions acquired from the web. Smith-Waterman string similarity metric was used in the question-pattern mapping module to measure the similarity between a user question and the set of relational patterns so as to get the relational pattern that was most similar to the user question i.e. the relational pattern with the highest similarity score with respect to the user question. The results of the pattern mapping were promising, with 79% of the questions being correctly mapped to the matching pattern when using the auto-extracted patterns and 90% of the questions being correctly mapped when using the gold-standard patterns (see Table 5.1). Once the matching relational pattern for a particular question was acquired, its associated Sparql query was instantiated with the entity in the user question and sent to DBpedia endpoint for answer retrieval.

To answer the research questions, entailment based QA paradigm is possibly a good method for QA over linked data. It has been shown that one could acquire relational patterns showing different ways in which a predicate can be expressed. The difficulty would be in finding many reliable pages for extracting relational patterns for as many relations are possible.

When it came to the patterns extracted from Wikipedia, some patterns e.g. for “birthDate” were not expressive enough to cover many linguistic variabilities. Most articles used the form *[PERSON] (born [DATE])* which is different from how users pose questions e.g. *When was [PERSON] born? On what date was [PERSON] born?* e.t.c. This could be due to the encyclopedic nature of wikipedia. For other relations like “launchDate”, “numberOfStudents”, “writer”, “gameEngine”, the extracted patterns represented a wider range of linguistic variabilities and were similar to how users pose questions. Some example questions and matching patterns shown in Table 6.1.

Question	Relational pattern
How many students enrolled in [UNIVERSITY]?	[NONNEGATIVEINTEGER] enrolled at [UNIVERSITY]
Who wrote the lyrics to [WORK]?	[PERSON] wrote [WORK]
When was the [SPACEMISSION] launch into space?	[SPACEMISSION] launched [DATE]
What game engine is [VIDEOGAME] developed on?	[VIDEOGAME] developed using [THING]

Table 6.1 – User questions with similar relational patterns

Relational patterns extracted from text are scalable to the many relations that exist provided one finds reliable sources for extracting the relational patterns.

6.2 Further research

The research area on Question Answering, and especially on scaling up QA to linked data, is a wide and emergent research area that still needs more study in order to benefit from the rich linked data resources available on the Web. Possible future work would be in ways of handling the heterogeneous nature of the resources and ontologies, and aggregating answers from different linked data resources.

Simple string matching algorithms were used for automatic sentence extraction. More sophisticated methods could be used to get more precise sentences expressing different relations. Also, the results for automatic pattern evaluation show that some optimization still needs to be done on the extracted relational patterns.

In the thesis, the questions used to test the pattern mapping module were only asking about the range of the relation. Research on the bi-directional use of the relational patterns to answer questions asking about either the range or the domain of the relation could be pursued further. Also, the questions used in the experiment expressed not more than one relation. For questions expressing more than one relation, different parts of the question could map to different relational patterns and therefore, it would be necessary to aggregate the queries for answer retrieval. A study on questions expressing many relations and the possible ways of aggregating the Sparql queries for answer retrieval is a possible further research.

For the feasibility experiments, the Sparql for each relational patterns were developed offline. A possible future work would be in the direction of automating the construction of Sparql queries from relational patterns.

Also, simple similarity measures have been used to get the similarity score between the user question and the set of relational patterns. It would be interesting to perform textual entailment using entailment engines to handle the various aspects of entailment, syntactic as well as semantic, and also to guide the patterns during pattern mapping process.

Bibliography

- [Androutsopoulos *et al.* 1995] I. Androutsopoulos, G.D Ritchie, P. Thanisch. (1995): Natural Language Interfaces to Databases - An Introduction. *Journal of Natural Language Engineering*.
- [Artstein and Poesio, 2008] Ron Artstein and Massimo Poesio. (2008): Inter-coder agreement for computational linguistics (survey article). *Computational Linguistics* 34(4): 555-596.
- [Eugenio, 2000] Barbara Di Eugenio. (2000): On the usage of Kappa to evaluate agreement on coding tasks. *LREC2000, the Second International Conference on Language Resources and Evaluation, Athens, Greece*.
- [Bizer *et al.*, 2009a] Bizer, C., Heath, T. and Berners-Lee, T. (2009): Linked Data - The Story So Far. *Int. J. Semantic Web Inf. Syst.*, 5 (3): pp. 1-22.
- [Bizer *et al.*, 2009b] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak and Sebastian Hellmann. (2009): PDF DocumentDBpedia – A Crystallization Point for the Web of Data. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web, Issue 7, Pages 154–165*.
- [Carletta *et al.*, 1997] Carletta, J. C., Isard, A., Isard, S., Kowtko, J., Doherty-Sneddon, G., and Anderson, A. (1997): The Reliability of a Dialogue Structure Coding Scheme. *Computational Linguistics*, 23(1): 13-31.
- [Carletta, 1996] Carletta, Jean. (1996): Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–254.
- [Damljanovic *et al.*, 2011] Danica Damljanovic, Milan Agatonovic and Hamish Cunningham: (2011): FREYA: an Interactive Way of Querying Linked Data using Natural Language. *Proceedings of the 1st Workshop on Question Answering Over Linked Data (QALD-1)*, Pages 10-23.
- [Damljanovic *et al.*, 2010] Danica Damljanovic, Milan Agatonovic and Hamish Cunningham. (2010): Natural Language Interfaces to Ontologies: Combining Syntactic Analysis and Ontology-Based Lookup through the User Interaction. *ESWC (1)*: 106-120.

-
- [D'Aquin *et al.*, 2007] D'Aquin, M., Baldassarre, C., Gridinoc, L., Angeletou, S., Sabou, M. and Motta, E. (2007): Characterizing knowledge on the semantic web with Watson. *In Proc. of 5th EON Workshop at International Semantic Web Conference.*
- [Dice, 1945] Lee R. Dice. (1945): Measure of the amount of ecological association between species. *Journal of Ecology*, 26(3):297-302.
- [Doddington *et al.*, 2000] Doddington, G., A. Mitchell, M. Przybicki, L. Ramshaw, S. Strassell, and R. Weischedel. (2000): The automatic content extraction (ACE) program—tasks, data, and evaluation. *In Proc. of LREC.*
- [Dagan and Glickman, 2004] Ido Dagan and Oren Glickman. (2004): Probabilistic textual entailment: generic applied modeling of language variability. *In PASCAL workshop on Learning Methods for Text Understanding and Mining, Grenoble.*
- [Ferrández *et al.*, 2011] Óscar Ferrández, Christian Spurk, Milen Kouylekov, Iustin Dornescu, Sergio Ferrández, Matteo Negri, Rubén Izquierdo, David Tomás, Constantin Orasan, Günter Neumann, Bernardo Magnini, Jose Luis Viced (2011): The Qallme Framework: A specifiable-domain multilingual Question Answering architecture. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web volume doi:10.1016/j.websem.2011.01.002, Pages 1-12, Elsevier, 2/2011.*
- [Frank *et al.*, 2007] Anette Frank, Hans-Ulrich Krieger, Feiyu Xu, Hans Uszkoreit, Berthold Crysmann, Brigitte Jörg and Ulrich Schäfer. (2007): Question answering from structured knowledge sources. *J. Applied Logic* 5(1): 20-48.
- [Granger *et al.*, 2002] Granger, S., Hung, J., and Petch-Tyson, S. (2002): eds. Computer learner corpora, second language acquisition, and foreign language teaching. *Amsterdam: John Benjamins.*
- [Harabagiu *et al.*, 2003] Sanda M. Harabagiu, Marius A. Păcă and Steven J. Maiorano. (2003): Open-domain textual question answering techniques. *Natural Language Engineering* 9:0303, 231-267.
- [Hasler, 2008] Hasler, L. (2008): Spoken Requests for Tourist Information: a Speech Acts: a Speech Acts Annotation. *Text, Speech and Dialogue - 11th International Conference, TSD 2008, Brno, Czech Republic.*
- [Hendrickx *et al.*, 2010] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Marco Pennacchiotti, Diarmuid O Seaghdha, Sebastian Padó, Lorenza Romano and Stan Szpakowicz. (2010): SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations Between Pairs of Nominals. *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2010).*
- [Komagata, 2002] Komagata, Nobo. (2002): Chance Agreement and Significance of the Kappa Statistic.
- [Leech, 2005] Leech, G. (2005): "Adding Linguistic Annotation" in Developing Linguistic Corpora: a Guide to Good Practice, ed. M. Wynne. *Oxford: Oxbow Books: 17-29.*

- [Lesmo and Robaldo, 2006] L. Lesmo and L. Robaldo. (2006): From Natural Language to Databases via Ontologies. *In Proc. of Lexical Resources and Evaluation Conference (LREC2006), Genova, Italy.*
- [Lopez *et al.*, 2011] Lopez, V., Uren, V., Sabou, M and Motta, E. (2006): Is question answering fit for the semantic web? (2011): a survey. *Accepted for publication in: Semantic Web Journal.*
- [Lopez *et al.*, 2010] Lopez, V., Andriy Nikolov, Marta Sabou, Victoria S. Uren, Enrico Motta and Mathieu d'Aquin. (2010): Scaling Up Question-Answering to Linked Data. *EKAW 2010: 193-210.*
- [Lopez *et al.*, 2009] Lopez, V., Victoria S. Uren, Marta Sabou and Enrico Motta. (2009): Cross ontology query answering on the semantic web: an initial evaluation. *K-CAP 2009: 17-24.*
- [Magnini *et al.*, 2006] B. Magnini, E. Pianta, C. Girardi, M. Negri, L. Romano, M. Speranza, and R. Sprugnoli. (2006): *I-CAB: the Italian Content Annotation Bank, in Proceedings of LREC.*
- [Mahedra, 2011] Rahmad Mahedra. (2011): Automatic Extraction of Relational Patterns from Wikipedia. Unpublished master's thesis (in progress), Free University of Bozen-Bolzano, Bolzano, Italy.
- [Negri and Kouylekov, 2009] Negri M. and Kouylekov M. (2009): Question Answering over Structured Data: an Entailment-Based Approach to Question Analysis. *In Proceedings of RANLP 2009*, September 14-16, Borovets, Bulgaria.
- [Nguyen *et al.*, 2007] D. P. T. Nguyen, Y. Matsuo and M. Ishizuka. (2007): Exploiting Syntactic and Semantic Information for Relation Extraction from Wikipedia. *IJCAI Workshop on Text-Mining and Link-Analysis (TextLink 2007).*
- [Popescu *et al.*, 2003] Ana-Maria Popescu, Oren Etzioni and Henry Kautz. (2003): Towards a Theory of Natural Language Interfaces to Databases. *In Proceedings of the 8th international conference on Intelligent user interfaces.*
- [QALD-1 proceedings, 2011] Proceedings of the 1st Workshop on Question Answering Over Linked Data (QALD-1). (May 30, 2011): Heraklion, Greece Co-located with the 8th Extended Semantic Web Conference.
- [QALL-ME Consortium, 2009a] Qall - Me Consortium (<http://qallme.fbk.eu/>). (2009): Qall-me Framework and Textual Entailment Based Question Answering, Tech. Rep., FBK, University of Wolverhampton, University of Alicante, DFKI, Comdata, Ubiest, Waycom.
- [QALL-ME Consortium, 2009b] Qall - Me Consortium (<http://qallme.fbk.eu/>). (2009): Qall-me - Pattern Acquisition for Question Answering, Tech. Rep., FBK, University of Wolverhampton, University of Alicante, DFKI, Comdata, Ubiest, Waycom.

- [QALL-ME Consortium, 2009c] Qall - Me Consortium (<http://qallme.fbk.eu/>). (2009): Qall-me - Semantic Web and Data Access, Tech. Rep., FBK, University of Wolverhampton, University of Alicante, DFKI, Comdata, Ubiest, Waycom.
- [Ravichandran and Hovy, 2002] Ravichandran, D. and E.H. Hovy. (2002): Learning Surface Text Patterns for a Question Answering System. *Proceedings of the 40th ACL conference, Philadelphia, PA*.
- [Roth *et al.*, 2002] D. Roth, C. Cumby, X. Li, P. Morie, R. Nagarajan, V. Punyakanok, N. Rizzolo, K. Small and W. Yih. (2002): Question-Answering via Enhanced Understanding of Questions. *TREC - 2002*
- [Rosenberg and Binkowski, 2004] Rosenberg, A. and Binkowski, E. 2004. (2004): Augmenting the Kappa Statistic to Determine Inter-annotator Reliability for Multiply Labeled Data Points. *HLT/NAACL*.
- [Sacaleanu *et al.*, 2008] Sacaleanu B., Spurk C., Orasan C., Ferrandez O., Kouylekov M., Negri M., and Ou S. (2008): Entailment based Question Answering over structured data. in: Allan Ramsay, Kalina Bontcheva (eds.): *Coling 2008: Companion Volume: Posters and Demonstration, pages 29-32, Manchester, UK, Coling 2008 Organizing Committee*.
- [Shen *et al.*, 2005] Shen D. , Kruijff G. J. and Klakow D. (2005): Exploring Syntactic Relation Patterns for Question Answering. *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP)*.
- [Tanaka *et al.*, 2010] Shohei Tanaka , Naokaki Okazaki and Mitsuru Ishizuka. (2010): Learning web query patterns for imitating Wikipedia articles. *Proceeding COLING '10 Proceedings of the 23rd International Conference on Computational Linguistics*.
- [Smith and Waterman, 1981] T.F. Smith and M.S. Waterman. (1981): Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195 - 197.
- [Soubotin and Soubotin, 2001] M.M. Soubotin and S.M. Soubotin. (2001): Patterns of Potential Answer Expressions as Clues to the Right Answer. *Proceedings of the TREC-10 Conference. NIST, Gaithersburg, MD, 134-143*.
- [Vila *et al.*, 2010] Marta Vila, Horacio Rodriguez and M. Antonia Marti. (2010): WRPA: A System for Relational Paraphrase Acquisition from Wikipedia. *Procesamiento del Lenguaje Natural, Vol. 45, pages 11-19*.
- [Webb and Webber, 2008] N Webb and B Webber. (2008): Special issue on the Interactive Question Answering: Introduction. *Natural language Engineering 1 (1):1-8*.
- [Zesch *et al.*, 2007] Zesch, T., I. Gurevych, and M. Mühlhäuser. (2007): Analyzing and Accessing Wikipedia as a Lexical Semantic Resource, in *Data Structures for Linguistic Resources and Applications*, pp. 197-205.

Appendix A

Annotation guidelines

The corpus is developed by two annotators and the following annotation guidelines were used in order to facilitate the annotation process.

1. Target relation expressed differently in the same sentence

If the target relation in a sentence is expressed in more than one way, the sentence is duplicated (triplicated –as necessary–), each variation annotated uniquely.

Example 1

- (a) `<s id="19554632-1" relation="writer">"<e1>Big Dipper</e1>" is a
<rel>song by</rel> <e2>Elton John</e2> with lyrics by Gary Os-
borne. </s>`
- (b) `<s id="19554632-1" relation="writer">"<e1>Big Dipper</e1>" is a
song by Elton John with <rel>lyrics by</rel> <e2>Gary Osborne</e2>.
</s>`

Example 2

- (a) `<s id="2075364-3" relation="writer">Featuring <e1>lyrics</e1> <rel>
written by</rel> vocalist <e2>Eddie Vedder</e2> and music written
by guitarist Stone Gossard, "Black" is a soliloquy by a broken-hearted
man, who is remembering his absent lover.</s>`
- (b) `<s id="2075364-3" relation="writer">Featuring lyrics written by vo-
calist Eddie Vedder and <e1>music</e1> <rel>written by</rel> gui-
tarist <e2>Stone Gossard</e2>, "Black" is a soliloquy by a broken-
hearted man, who is remembering his absent lover. </s>`

2. Relation mentions expressed in compound nouns

For relation mentions expressed in compound nouns, if it is expressed in one compound word, the whole compound relation is annotated, otherwise only the target relation is annotated.

Example

➤ <s id="2032-4" relation="writer"> In keeping with <rel>writer-director</rel>
<e2>Quentin Tarantino's</e2> trademark of nonlinear storytelling,
the narrative is presented out of sequence. </s>

3. Implicit mentions of relations

Implicit mentions of relations are annotated. Implicit means that there is no surface realization in words of the relation that can be annotated. The domain and the range must both be present in the sentence.

Examples

- (a) <s id="2303-5" relation="birthDate"> <e1>Albert Bernard Ackerman</e1> (<e2>November 22, 1936</e2> – December 5, 2008)
was an American physician.....</s>
- (b) <s id="2405-23" relation="language"><e1>Japanil Kalyanaraman</e1>
is a <e2>Tamil</e2> language film starring Kamal Haasan in the lead
role of the protagonist. </s>

4. Target relation annotation variants

There are 3 proposals of annotating the different ways in which a target relation can be expressed, thus relations are annotated according to either of the three.

Example 1: Verb + prep

The verb and preposition next to one another express the target relation.

- (a) <s id="2024-18" relation="numberOfFilms"><e1>She</e1> has
since <rel>appeared in</rel> over <e2>250</e2> adult films.</s>

Example 2: verb only

- (a) <s id="20552-90" relation="gameEngine"><e1>Prince of Persia</e1>
<rel>utilizes</rel> a heavily-modified version of the <e2>Scimitar</e2>
engine, which was also used in Assassin's Creed.</s>

Example 3: Verb [anything] prep*

The verb and preposition expressing the relation are not next to each other.

- (a) <s id="405-34" relation="launchDate">The <rel>launch</rel>, at
7:00 a.m. EST <rel>on</rel> <e2>November 9, 1967</e2> from
Launch Complex 39, was the first from the John F. Kennedy Space
Center on Merritt Island.</s>
- (b) <s id="20232-1" relation="wineRegion"> <e1>Albana</e1> is a
white Italian wine grape <rel>planted</rel> primarily <rel>in</rel>
the <e2>Emilia-Romagna</e2> region.</s>
- (c) <s id="20782-3" relation="writer"><e1>The song</e1> was <rel>written</rel>
solely <rel>by</rel> members, <e2>Maynard James Keenan</e2>,
Billy Howerdel and Josh Freese.</s>

Example 4 : prep only

- (a) <s id="2496111-3" relation="numberOfStudents">As of 2007, the <e1>Colorado school district</e1> consists of 30 elementary schools, 8 middle schools, 5 high schools, 4 charter schools, 1 alternative school, 1 technical education center, and 1 adult education center <rel>with</rel> nearly <e2>39,000</e2> students enrolled in the district.</s>
- (b) <s id="3598487-1" relation="numberOfStudents"><e1>Annie Wright School</e1> is a preschool-12th grade independent school <rel>of</rel> about <e2>450</e2> students.</s>
- (c) <s id="12210477-1" relation="wineRegion"><e1>Blatina</e1> is red wine grape variety <rel>of</rel> Bosnia and <e2>Herzegovina</e2>.</s>

Example 5: Including auxiliary verb

The auxiliary verb is included in the relation pattern only if the meaning of the pattern might change with the absence or presence of the auxiliary.

Example 6: Modifiers - adverbs, adjectives, prepositional phrases

Expressions that play the role of syntactic modifiers are not annotated as part of the relation pattern.

- <s id="1685671-9" relation="numberOfStudents">The <e1>School</e1> <rel>has</rel> approximately <e2>896</e2> day students and 80 boarding students; girls from Junior Kindergarten to Grade 12 (approximately ages 4–17).</s> –approximately is not annotated–

5. Domain and Range annotation regulations

Named entity annotation conventions¹ are observed while tagging the domain and range e.g. only capitalized generic names are annotated, only the head NP without the determiner is annotated.

Example 1: None capitalized generic names are not tagged

- (a) <s id="20982-90" relation="gameEngine"><e1>Prince of Persia</e1> <rel>utilizes</rel> a heavily-modified version of the <e2>Scimitar</e2> engine, which was also used in Assassin’s Creed.</s> – ‘engine’ left out –
- (b) <s id="5939-34" relation="language"><e1>Japanil Kalyanaraman</e1> is a <e2>Tamil</e2> language film starring Kamal Haasan in the lead role of the protagonist. – ‘language’ left out –

Example 2: Capitalized generic names are tagged

- (a) <s id="23796306-1" relation="gameEngine"><e1>Epic Mickey</e1> (sometimes marketed as Disney Epic Mickey) is a Mickey Mouse video game designed by Warren Spector, with 2D cinematics by Powerhouse Animation Studios, Inc. and <rel>developed</rel> by Junction Point

¹http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/ne_task.html

Studios for the Wii console, <rel>using</rel> Emergent Game Technologies’ <e2>Gamebryo Engine</e2>.</s>

Example 3: NP is tagged without the article or titles

- (a) <s id="1181487-29" relation = "gameEngine">All <e1>games</e1> are <rel>recreated</rel> in real-time on the Xbox 360 <rel>using</rel> the <e2>Halo 3</e2> engine.</s>
- (b) <s id="1685671-9" relation = "numberOfStudents">The <e1>School</e1> <rel>has</rel> approximately <e2>896</e2> day students and 80 boarding students; girls from Junior Kindergarten to Grade 12 (approximately ages 4–17).</s>
- (c) <s id="24751173-1" relation = "birthDate">Lieutenant <e1>Arthur Rahn</e1> (<re>born</re> <e2>18 July 1897</e2>, date of death unknown) was a World War I flying ace credited with six aerial victories.</s>

Example 5: Pronouns representing the domain entity are annotated.

- (a) <s id="1906406-3" relation = "gameEngine"><e1>It</e1> is a first person shooter and <rel>uses</rel> the <e2>Lithtech</e2> graphics engine, made popular by the more successful and critically acclaimed shooter No One Lives Forever.</s>

Example 6: If the range entity is a date or number, but expressed in a string, the sentence is annotated.

- (a) <s id="21621113-3">Established in 1969 <e1>Bristol Law School</e1> has since grown into one of the largest law school in England and Wales with nearly <e2>two thousand students</e2> enrolled (seventeen hundred full time students).</s>
- (b) <s id="1655252-1">The <e1>High School of Dundee</e1> is an independent, co-educational, day school in the city of Dundee, Scotland which provides both primary and secondary education to just over <e2>one thousand</e2> pupils.</s>

6. Multiple Range and/or Domain

If a sentence for some relation has multiple Range and/or Domain separated by conjunctions, “and, or” e.t.c, or in a list, the sentence is split *n* times, one for each occurrence of the domain/range.

Example 1: split thrice for each of the ranges

- (a) <s id="202-3" relation = "writer">The <e1>song</e1> was <rel>written</rel> solely <rel>by</rel> members, <e2>Maynard James Keenan</e2>, Billy Howerdel and Josh Freese.</s>
- (b) <s id="202-3" relation = "writer">The <e1>song</e1> was <rel>written</rel> solely <rel>by</rel> members, Maynard James Keenan, <e2>Billy Howerdel</e2> and Josh Freese.</s>

- (c) <s id="202-3" relation="writer">The <e1>song</e1> was <rel>written</rel> solely <rel>by</rel> members, Maynard James Keenan, Billy How-
erdel and <e2>Josh Freese</e2>.</s>

Example 2: split twice for each of the ranges

- (a) <s id="29802-7" relation="wineRegion"><e1>It</e1> is a commer-
cially viable grape vine which is <rel>grown in</rel> the <e2>North
East</e2> and Mid West of <e2>America</e2> and is vigorous when
grafted onto a phylloxera resistant root stock.</s>
- (b) <s id="29802-7" relation="wineRegion"><e1>It</e1> is a commer-
cially viable grape vine which is <rel>grown in</rel> the North East
and <e2>Mid West of America</e2> and is vigorous when grafted onto
a phylloxera resistant root stock.</s>

7. language

The region name represents the language of the region.

Examples

- (a) <s id="28175742-1" relation="language"><e1>After Four</e1> was
a <e2>Canadian</e2> youth television series which aired on CBC Tele-
vision from 1977 to 1978.</s>–Canadian for English–
- (b) <s id="22645001-1" relation="language"><e1>All Stars</e1> is a
1997 <e2>Dutch</e2> sports comedy film drama directed by Jean van
de Velde.</s>–Dutch for Dutch–
- (c) <s id="2399020-1" relation="language"><e1>Black Friday</e1> ()
is a 2004 <e2>Indian</e2> film by Anurag Kashyap about the 1993
Bombay bombings.</s>–Indian for Hindi–

8. birthDate

- (a) Tag both birth name and changed name in separate sentences
- i. <s id="22332190-1" relation="birthDate"><e1>Andrej Komatovic</e1>, widely
known as Andy Blueman (<rel>born</rel> <e2>September 4, 1982</e2>) is
a Slovenian Trance producer.</s>
 - ii. <s id="22332190-1" relation="birthDate">Andrej Komatovic, widely known as
<e1>Andy Blueman</e1> (<rel>born</rel> <e2>September 4, 1982</e2>)
is a Slovenian Trance producer.</s>
- (b) Include nicknames as part of the name
- <s id="25117753-1" relation="birthDate"><e1>Anthony "Romeo" Santos</e1>
(<rel>born</rel> <e2>July 21, 1981</e2>) is an American singer and fea-
tured composer of the Bachata group Aventura.</s>

9. writer

A piece of work can be written by more than one person, therefore, the Group or the individuals of the group are considered to have written the work.

➤ <s id="20332548-1" relation="writer"><e1>"And the Bands Played On"</e1> is a <rel>single by</rel> heavy metal band <e2>Saxon</e2> from their 1981 album Denim and Leather.</s> –Saxon is the name of the band that represents the individual members who are the writers of the song–

10. numberOfStudents

The value expressed in the text is an approximation of the exact value in the infobox. Such sentences are considered correct.

- (a) The infobox value for numberOfStudents is 174

➤ <s id="2005-10" relation="numberOfStudents">The <e1>school's</e1> image diminished during the last years, with pupil numbers dropping from 600 to just over <e2>200</e2>. </s>

- (b) The infobox value for numberOfStudents is 1400

➤ <s id="2006-11" relation="numberOfStudents">At the beginning of the 2005-2006 school year, <rel>enrollment</rel> was <e2>1,142</e2> (PK-grade 5: 581; grades 6-8: 266; and grades 9-12: 295) and </s>

- (c) “There are..” is considered to be expressing the relation.

➤ <s id="22705302-5" relation="numberOfStudents"><rel>There are</rel> over <e2>800</e2> students, and even though the school is an engineering school, it has quickly earned a reputation for an excellent sports selection.</s>

- (d) “The number of students/ student population ..” are considered to be expressing the Domain

- i. <s id="19283659-3" relation="numberOfStudents">The <e1>student population</e1> <rel>is</rel> approximately <e2>2500</e2> in grades 9 through 12.</s>
- ii. <s id="3145896-22" relation="numberOfStudents">The <e1>number of students</e1> <rel>varies</rel> year to year <rel>from</rel> <e2>750-1000</e2>, currently home to grades 7 to 12.</s>

11. wineRegion

If a sentence is mentioned as a wine region but the region name does not appear in the infobox, it is not annotated. But if the exact region name or its sub-regions or super-regions is mentioned in the text, the sentence is annotated.

- (a) North East and Mid West of USA-North East and Mid West of America

➤ <s id="29802-7" relation="wineRegion"><e1>It</e1> is a commercially viable grape vine which is <rel>grown in</rel> the <e2>North East</e2> and Mid West of <e2>America</e2> and is vigorous when grafted onto a phylloxera resistant root stock.</s>

- (b) Italy – the regions in Italy (Umbria’s Orvieto region, Torgiano and Colli Martani)

➤ <s id="2677-7" relation="wineRegion">The <e1>Grechetto</e1> grape is
<rel>found in</rel> DOCs of the central region-most notably <e2>Umbria's
Orvieto</e2> region as well as the DOCs of Torgiano and Colli Martani.</s>

12. collection

For this, for some pages, the string value for the range is the name of the collection, while for others it's the number of collections in the museum. The page annotation is based only on the value of the infobox.

(a) The infobox value is string representing the name of the collection:

- i. <s id="1522205-3" relation="collection"><e1>It</e1> is one of the oldest galleries in the world and <rel>houses</rel> one of the most famous collections of <e2>Old Master</e2> paintings.</s>
- ii. <s id="28255781-3" relation="collection">The <e1>museum</e1> <rel>features</rel> <e2>forty wax statues</e2> of notorious criminals, from mobsters to serial killers.</s>

(b) The infobox value is string representing the number of objects:

- i. <s id="4675-2" relation="collection">Its <e1>collections</e1>, which <rel>number</rel> more than <e2>seven million objects</e2>, are amongst the largest and most comprehensive in the world and originate from all continents, illustrating and documenting the story of human culture from its beginnings to the present.</s>
- ii. <s id="7214737-4" relation="collection">The <e1>collection</e1> <rel>of</rel> more than <e2>135,000 artifacts</e2> forms the basis for exhibitions in four distinct wings: Bartlett, Girard, Hispanic Heritage, and Neutrogena.</s>

13. isPartOf

Because of the complexity of the domain, only the exact range value as depicted in the infobox or DBpedia resource page is tagged.

Appendix B

Codes for the developed modules

The implementation of all modules and the datasets used in the experiments carried out in this thesis can be downloaded from <https://sites.google.com/site/lilywanzie/Codes-and-Files> (under Codes-and-Files).

B.1 Description of each module

B.1.1 Inter-Annotator agreement

- Description: Program to calculate the inter-annotator agreement, using Dice coefficient algorithm, between the two annotators involved in the annotation process.
- Programming language: Python (Inter_Annotator_Agreement.py).
- Input: It takes as input two files, each file is the annotation work carried out by an annotator for a single relation.
- Output: The Dice coefficient for the sentences, Domain, Range and Relation tags for each relation.

B.1.2 Pattern Extraction

- Description: This program extracts relational patterns from the gold-standard sentences i.e. each relational pattern is an ordered set of domain, range and relation tags found in an annotated sentence.
- Programming language: Java (ExtractPatterns.java).
- Input: It takes as input the gold-standard sentences for each relation.
- Output: Set of relational patterns for each relation.

B.1.3 Question-Pattern mapping module

- Description: This module maps a user question to a relational pattern from the set of relational patterns i.e. gets the relational pattern that has the highest similarity score with

respect to the user question. Using the selected relational pattern, it gets the associated sparql query, instantiates the variable in the query with the entity in the user question and sends the query to DBpedia sparql endpoint to retrieve the answer to the question.

- Input: It takes as input the set of relational patterns and a user question.
- Output: The instantiated sparql query of the question and an XML object, returned from DBpedia sparql endpoint, containing the answer to the question.

B.1.4 Evaluation

B.1.4.1 Sentence extraction

- Description: This program calculates the Precision, Recall and F1-measure of the automatically extracted sentences, for each relation, against the gold-standard sentences.
- Programming language: Java (Sentence_Evaluation.java).
- Input: It takes as input the sentences automatically extracted by the system, one file for each relation, and the gold-standard sentences.
- Output: Precision, Recall and F1-measure of the automatically extracted sentences for each relation.

B.1.4.2 Relational Pattern extraction

- Description: This program calculates the Precision, Recall and F1-measure of the automatically extracted relational patterns, for each relation, against the gold-standard.
- Programing language: Java (Pattern_Evaluation.java)
- Input: It takes as input the relational patterns automatically extracted by the system, one file for each relation, and the gold-standard sentences.
- Output: Precision, Recall and F1-measure of the automatically extracted relational patterns for each relation.

B.2 Datasets developed

The below datasets that have been developed can also downloaded from the site.

1. Gold-standard Sentences
2. Gold-standard Relational Patterns
3. Auto-Extracted Patterns
4. User Questions