

# ENCYCLOPAEDIC QUESTION ANSWERING

Iustin Dornescu

A thesis submitted in partial fulfilment of the  
requirements of the University of Wolverhampton  
for the degree of Doctor of Philosophy  
February 2012

This work or any part thereof has not previously been presented in any form to the University or to any other body whether for the purposes of assessment, publication or for any other purpose (unless otherwise indicated). Save for any express acknowledgments, references and/or bibliographies cited in the work, I confirm that the intellectual content of the work is the result of my own efforts and of no other person.

The right of Iustin Dornescu to be identified as author of this work is asserted in accordance with ss.77 and 78 of the Copyright, Designs and Patents Act 1988. At this date copyright is owned by the author.

Signature: .....

Date: .....



# Abstract

Open-domain question answering (QA) is an established NLP task which enables users to search for specific pieces of information in large collections of texts. Instead of using keyword-based queries and a standard information retrieval engine, QA systems allow the use of natural language questions and return the exact answer (or a list of plausible answers) with supporting snippets of text. In the past decade, open-domain QA research has been dominated by evaluation fora such as TREC and CLEF, where shallow techniques relying on information redundancy have achieved very good performance. However, this performance is generally limited to simple factoid and definition questions because the answer is usually explicitly present in the document collection. Current approaches are much less successful in finding implicit answers and are difficult to adapt to more complex question types which are likely to be posed by users.

In order to advance the field of QA, this thesis proposes a shift in focus from simple factoid questions to encyclopaedic questions: list questions composed of several constraints. These questions have more than one correct answer which usually cannot be extracted from one small snippet of text. To correctly interpret the question, systems need to combine classic knowledge-based approaches with advanced NLP techniques. To find and extract answers, systems need to aggregate atomic facts from heterogeneous sources as opposed to simply relying on keyword-based similarity. Encyclopaedic questions promote QA systems which use basic reasoning, making them more robust and easier to extend with new types of constraints and new types of questions. A novel semantic architecture is proposed which represents a paradigm shift in open-domain QA system design, using semantic concepts and knowledge representation instead of words and information retrieval. The architecture consists of two phases, analysis – responsible for interpreting questions and finding answers, and feedback – responsible for interacting with the user.

This architecture provides the basis for EQUAL, a semantic QA system developed as part of the thesis, which uses Wikipedia as a source of world knowledge and

employs simple forms of open-domain inference to answer encyclopaedic questions. EQUAL combines the output of a syntactic parser with semantic information from Wikipedia to analyse questions. To address natural language ambiguity, the system builds several formal interpretations containing the constraints specified by the user and addresses each interpretation in parallel. To find answers, the system then tests these constraints individually for each candidate answer, considering information from different documents and/or sources. The correctness of an answer is not proved using a logical formalism, instead a confidence-based measure is employed. This measure reflects the validation of constraints from raw natural language, automatically extracted entities, relations and available structured and semi-structured knowledge from Wikipedia and the Semantic Web. When searching for and validating answers, EQUAL uses the Wikipedia link graph to find relevant information. This method achieves good precision and allows only pages of a certain type to be considered, but is affected by the incompleteness of the existing markup targeted towards human readers. In order to address this, a semantic analysis module which disambiguates entities is developed to enrich Wikipedia articles with additional links to other pages. The module increases recall, enabling the system to rely more on the link structure of Wikipedia than on word-based similarity between pages. It also allows authoritative information from different sources to be linked to the encyclopaedia, further enhancing the coverage of the system.

The viability of the proposed approach was evaluated in an independent setting by participating in two competitions at CLEF 2008 and 2009. In both competitions, EQUAL outperformed standard textual QA systems as well as semi-automatic approaches. Having established a feasible way forward for the design of open-domain QA systems, future work will attempt to further improve performance to take advantage of recent advances in information extraction and knowledge representation, as well as by experimenting with formal reasoning and inferencing capabilities.

# Acknowledgements

The completion of this thesis would not have been possible without the support and encouragement of my supervisors, colleagues, family and friends. Many people have contributed, in one way or another, by discussing ideas, reviewing my research, providing feedback, participating in experiments, giving friendship and support, and reminding me to take time off once in a while. To all of them, I owe a special thank you.

In particular I want to thank my supervisors, Dr. Constantin Orăsan, Prof. Ruslan Mitkov and Prof. Roberto Navigli. I have benefited enormously from their insight, enthusiasm and critical engagement with my research over the past few years. I am sincerely grateful to my director of studies, Dr. Constantin Orăsan, who patiently steered and supported me during the course of my studies, and had enough faith in my skills to grant me this opportunity and freedom to carry out my PhD research. I also want to thank my examiners, Prof. Enrico Motta and Dr. Georgios Paltoglou, for their insightful comments and helpful suggestions to improve my thesis.

I am grateful to all my colleagues and fellow students from the Research Group in Computational Linguistics for their friendship as well as for their discussions and feedback on my work. They made a big difference and kept me happy – I cannot imagine my PhD years without them. Laura Hasler, Georgiana Marşic, Alison Carminke and Erin Stokes especially helped me with my thesis by proofreading. Wilker Aziz, Laura Hasler, Iustina Ilisei, Miguel Rios and Irina Temnikova were invaluable in participating in my experiments. Out of the office, Wilker and Miguel also made sure I relaxed every so often with good food and good company.

Special thanks go to Prof. Dan Cristea, who first introduced me to Natural Language Processing back in Iaşi and kindled my research interest in the field. I am also grateful for the collaboration and friendship with the members of his team from UAIC and the researchers from IIT, with whom I took my first NLP steps.

I would like to thank my family for their constant support, and their optimism and belief that I would complete my thesis. I am forever indebted to them for their encouragement which played a tremendous role in my decision to pursue a PhD. A big thank you to them and to my friends at home for welcoming me back each time as though nothing has changed, despite my ever-longer periods of absence. Thanks also to my English family for my Christmases away from home and the way they have included me.

Last and most of all, thank you to Laura, for everything – PhD-related and not.



# Contents

|  |             |
|--|-------------|
| <b>Abstract</b>  | <b>iii</b>  |
| <b>Acknowledgements</b>  | <b>v</b>    |
| <b>Table of Contents</b>                                       | <b>vii</b>  |
| <b>List of Tables</b>  | <b>xi</b>   |
| <b>List of Figures</b>   | <b>xiii</b> |
| <b>List of Abbreviations</b>                                   | <b>xv</b>   |
| <b>1 Introduction</b>  | <b>1</b>    |
| 1.1 Overview . . . . .   | 1           |
| 1.2 Aims and contributions . . . . .                           | 4           |
| 1.3 Structure . . . . .  | 5           |
| <b>2 Question Answering and Encyclopaedic Knowledge</b>        | <b>9</b>    |
| 2.1 Overview . . . . .   | 9           |
| 2.2 Question answering . . . . .                               | 9           |
| 2.2.1 Types of QA systems . . . . .                            | 11          |
| 2.2.2 General architecture of open-domain QA systems . . . . . | 20          |
| 2.2.3 QA and Web search . . . . .                              | 25          |
| 2.3 Wikipedia: a collaborative encyclopaedia . . . . .         | 26          |
| 2.4 Wikipedia: a resource for NLP research . . . . .           | 33          |
| 2.4.1 Wikipedia and semantic relatedness . . . . .             | 33          |
| 2.4.2 Wikipedia and information retrieval . . . . .            | 34          |
| 2.4.3 Wikipedia and information extraction . . . . .           | 36          |
| 2.4.4 Wikipedia and knowledge representation . . . . .         | 43          |
| 2.4.5 Wikipedia and question answering . . . . .               | 48          |

|          |   |           |
|----------|---|-----------|
| 2.4.6    | Conclusions . . . . .   | 50        |
| <b>3</b> | <b>Novel Approach for Encyclopaedic Question Answering</b>                | <b>53</b> |
| 3.1      | Overview . . . . .  | 53        |
| 3.2      | Textual QA . . . . .  | 53        |
| 3.3      | Criticism of textual QA . . . . .   | 55        |
| 3.3.1    | Natural language ambiguity . . . . .                                      | 56        |
| 3.3.2    | Design issues . . . . .   | 58        |
| 3.3.3    | Perceived performance . . . . .   | 60        |
| 3.4      | Proposed approach: paradigm shift . . . . .                               | 61        |
| 3.5      | Semantic architecture for open-domain QA . . . . .                        | 65        |
| 3.5.1    | Analysis phase . . . . .  | 68        |
| 3.5.2    | Feedback phase . . . . .  | 70        |
| 3.5.3    | Challenges . . . . .  | 72        |
| 3.6      | Conclusions . . . . .   | 75        |
| <b>4</b> | <b>EQUAL: Encyclopaedic QQuestion Answering system for List questions</b> | <b>77</b> |
| 4.1      | Overview . . . . .  | 77        |
| 4.2      | Geographic question answering . . . . .                                   | 78        |
| 4.3      | EQUAL in GikiP . . . . .  | 80        |
| 4.4      | EQUAL in GikiCLEF . . . . .   | 83        |
| 4.4.1    | Question parsing . . . . .  | 85        |
| 4.4.2    | Semantic interpretation and constraints . . . . .                         | 85        |
| 4.4.3    | Answer extraction . . . . .   | 88        |
| 4.4.4    | Constraint verifiers . . . . .  | 89        |
| 4.5      | Results . . . . .   | 93        |
| 4.5.1    | Evaluation metrics . . . . .  | 93        |
| 4.5.2    | GikiP results . . . . .   | 94        |
| 4.5.3    | GikiCLEF results . . . . .  | 97        |
| 4.6      | Discussion . . . . .  | 104       |

|          |  |            |
|----------|--|------------|
| 4.7      | Conclusions . . . . .                        | 108        |
| <b>5</b> | <b>Semantic Document Analysis</b>            | <b>109</b> |
| 5.1      | Overview . . . . .                           | 109        |
| 5.2      | Related work . . . . .                       | 111        |
| 5.3      | Densification: task definition . . . . .     | 115        |
| 5.4      | Densification: approach . . . . .            | 117        |
| 5.4.1    | Candidate pruning . . . . .                  | 117        |
| 5.4.2    | Candidate selection . . . . .                | 122        |
| 5.5      | Evaluation . . . . .                         | 123        |
| 5.5.1    | Human evaluation . . . . .                   | 124        |
| 5.5.2    | Wikipedia markup evaluation . . . . .        | 134        |
| 5.5.3    | Densification impact on EQUAL . . . . .      | 138        |
| 5.6      | Conclusions . . . . .                        | 141        |
| <b>6</b> | <b>Concluding Remarks</b>                    | <b>145</b> |
| 6.1      | Research goals revisited . . . . .           | 145        |
| 6.2      | Original contributions . . . . .             | 147        |
| 6.3      | Directions for future work . . . . .         | 149        |
|          | <b>References</b>                            | <b>153</b> |
|          | <b>Appendix A: Previously Published Work</b> | <b>168</b> |
|          | <b>Appendix B: GikiP Topics</b>              | <b>171</b> |
|          | <b>Appendix C: GikiCLEF Topics</b>           | <b>173</b> |



# List of Tables

|      |   |     |
|------|---|-----|
| 2.1  | Expected answer type classification scheme typically used in TREC . . . | 22  |
| 4.1  | Decomposition of topics into chunks by the question analyser . . . . .  | 86  |
| 4.2  | GikiP official results for all participants . . . . .                   | 94  |
| 4.3  | GikiP normalised results . . . . .                                      | 95  |
| 4.4  | GikiP results for EQUAL . . . . .                                       | 97  |
| 4.5  | GikiCLEF results for all languages . . . . .                            | 98  |
| 4.6  | GikiCLEF results for English . . . . .                                  | 103 |
| 4.7  | GikiCLEF normalised results for English . . . . .                       | 104 |
| 5.1  | $\chi^2$ ranking of candidate pruning features . . . . .                | 120 |
| 5.2  | Candidate pruning performance . . . . .                                 | 122 |
| 5.3  | Trade-off between recall and precision for candidate pruning . . . . .  | 122 |
| 5.4  | Criteria assessed by human raters . . . . .                             | 126 |
| 5.5  | Agreement for criterion Correctness . . . . .                           | 129 |
| 5.6  | Agreement for criterion Relatedness . . . . .                           | 131 |
| 5.7  | Number of targets per code for criterion Relatedness . . . . .          | 132 |
| 5.8  | Agreement for criterion Type . . . . .                                  | 133 |
| 5.9  | Wikification tools . . . . .  | 135 |
| 5.10 | Performance of wikification tools . . . . .                             | 136 |
| 5.11 | Densification: results for Wikipedia markup evaluation . . . . .        | 138 |
| 5.12 | Densification: performance impact on EQUAL . . . . .                    | 141 |
| 1    | GikiP 2008 topics (English) . . . . .                                   | 171 |
| 2    | GikiCLEF 2009 topics (English) . . . . .                                | 173 |



# List of Figures

|     |   |     |
|-----|---|-----|
| 2.1 | Partial view of Wikipedia’s Category Graph . . . . .      | 30  |
| 2.2 | Templates in Wikipedia . . . . .                          | 32  |
| 2.3 | Linked Open Data cloud diagram . . . . .                  | 47  |
| 3.1 | Generic architecture for open-domain QA . . . . .         | 67  |
| 4.1 | Sample of GikiP and GikiCLEF topics . . . . .             | 79  |
| 4.2 | Decomposition of question into constraints . . . . .      | 88  |
| 4.3 | Semantic interpretation of question constraints . . . . . | 90  |
| 5.1 | User interface employed in the experiment . . . . .       | 127 |



# List of Abbreviations

AI – Artificial Intelligence

AR – Anaphora Resolution

CL-ESA – Cross-Lingual Explicit Semantic Analysis

CL-IR – Cross-Lingual Information Retrieval

CLEF – Cross-Language Evaluation Forum

CFR – Conditional Random Fields

DUC – Document Understanding Conference

EAT – Expected Answer Type

EL – Entity Linking

ESA – Explicit Semantic Analysis

GA – Genetic Algorithm

KB – Knowledge Base

KR – Knowledge Representation

IDF – Inverse Document Frequency

IE – Information Extraction

INEX – Initiative for the Evaluation of XML retrieval

IR – Information Retrieval

Ka – Krippendorff's Alpha

LD – Linked Data

LOD – Linked Open Data

MI – Mutual Information

ML – Machine Learning

MUC – Message Understanding Conference

NER – Named Entity Resolution

NL – Natural Language

NLP – Natural Language Processing

NP – Noun Phrase

OWL – Web Ontology Language

RDF – Resource Description Framework

SVM – Support Vectors Machine

TAC – Text Analysis Conference

TF – Term Frequency

TREC – Text REtrieval Conference

URI – Uniform Resource Identifier

URL – Uniform Resource Locator

QA – Question Answering

QALD – Question Answering over Linked Data

WM – Wikipedia Miner

# Chapter 1

## Introduction

### 1.1 Overview

The World Wide Web has grown dramatically since its inception in 1992 as a global interconnected system for document sharing amongst researchers (Berners-Lee et al., 1992). With over 130 million<sup>1</sup> domains and a billion unique URLs (Alpert and Hajaj, 2008) and with more than two billion estimated users (Lynn, 2010), it has fundamentally transformed the way information is shared, distributed and accessed. As the amount of information available online started to grow exponentially, the need for increasingly sophisticated search tools led to the creation of Web search engines that allowed users to retrieve documents based on keyword queries. While search engines are very well suited for retrieving relevant documents, they are much less effective when users need to find very specific pieces of information.

To reduce time and effort in formulating effective queries, question answering (QA) systems were proposed as an alternative to Web search engines to help users who need to find small pieces of factual information rather than whole documents. Question answering is a specialised type of information access in which systems must return an exact answer to a natural language question (Maybury, 2004a). It uses natural language processing (NLP) techniques to process a question, then searches for the required information to identify the answer and presents the answer to the user.

---

<sup>1</sup>In February 2012, <http://www.domaintools.com/internet-statistics/> reported 137 million

When searching for very specific information, phrasing the request as a question and receiving a short answer snippet was seen as a far more natural and efficient process than repeatedly creating complex keyword-based queries and reading through the list of relevant documents retrieved by a standard search engine. This process is embodied in open-domain QA systems, which have recently seen a surge in research interest, motivated by the opportunity to provide an alternative to popular search engines such as Google, Yahoo and Bing. A typical open-domain QA system uses the words in a question to automatically create queries, processes documents retrieved by an IR engine, selects relevant text snippets and identifies the exact answer of the question (Harabagiu and Moldovan, 2003). Usually, a system creates a ranked list of candidate answers, each accompanied by a link to its source document and a relevant text snippet.

Research in QA was catalysed by a series of competitive evaluations such as those conducted in the Text REtrieval Conference (TREC) (Voorhees, 2001) and Cross-lingual Evaluation Forum<sup>2</sup> (CLEF). These have been instrumental in providing a shared evaluation platform and defining a roadmap for the community. Open-domain QA research has addressed several types of questions such as:

- factoid – *In what year was Warren Moon born?*
- list – *What celebrities have appeared on The Daily Show?*
- definition – *What does LPGA stand for?*
- how – *How can I eliminate stress?*
- why – *Why did David Koresh ask the FBI for a word processor?*
- time-dependent – *What cruise line attempted to take over NCL in December 1999?*
- semantically constrained – *How have thefts impacted on the safety of Russia's nuclear navy, and has the theft problem been increased or reduced over time?*

Despite significant research carried out in the past decade, open-domain QA systems have only been successful for a handful of question types, mainly those seeking simple factual information such as definition or factoid questions, most likely because these are

---

<sup>2</sup><http://clef.isti.cnr.it/> from 2000 to 2009, <http://www.clef-initiative.eu/> since 2008

easy to answer using simple question patterns which exploit *information redundancy* (Hovy et al., 2000). Information redundancy assumes that in large corpora the correct answer is present in many documents and is phrased in a variety of ways, some of which are likely to match a small set of simple surface patterns. Standard QA systems are designed to identify paragraphs that are likely to contain the answer. This strategy is less suitable for more complex questions, whose answers are not phrased explicitly in one snippet of text and instead require the system to deduce them by combining information from more than one snippet.

This thesis addresses a type of complex question which is challenging for existing approaches, for example *Which Brazilian football players play in clubs in the Iberian Peninsula?* or *Which Portuguese rivers flow through cities with more than 150,000 inhabitants?* These are **encyclopaedic questions**: open list questions that explicitly ask for a specific type of named entity and usually contain additional conditions or constraints. They are challenging because instead of one correct answer they have a whole set of named entities which usually need to be identified and validated independently. To address this type of question QA systems need to go beyond searching for the paragraphs containing explicit answers and instead take steps towards employing world knowledge and using simple forms of common-sense reasoning.

Wikipedia<sup>3</sup> is a resource which can help bootstrap a new generation of open-domain QA systems capable of dealing with such questions. Intensively studied by the research community, it is of interest not only because of its wide coverage of encyclopaedic content but also because of the semi-structured information embedded in its structure. This thesis advocates the use of Wikipedia as a backbone in the next generation of open-domain QA, by employing it as the main source of generic world knowledge. It brings together open-domain QA and Wikipedia to address encyclopaedic questions, whose answers are not typically located in one small textual snippet but instead have to be found by combining pieces of information from multiple sources.

---

<sup>3</sup><http://wikipedia.org>

## 1.2 Aims and contributions

The **aim** of this thesis is to advance open-domain QA research by enabling more complex questions to be addressed than possible using current approaches. A shift of focus is proposed from textual, factoid questions to more complex types, such as encyclopaedic questions. To achieve this aim, several goals need to be met:

**Goal 1** is to advance a paradigm shift in the design of open-domain QA systems by proposing a new semantic QA approach that does not use words at its core, but concepts, enabling a QA system to combine simple facts from various sources, both unstructured (textual) and semi-structured (knowledge bases).

**Goal 2** is to test the viability of this novel semantic approach by implementing it in a new QA system and evaluating its performance in comparison with other QA systems.

**Goal 3** is to develop and evaluate a semantic analysis method which enables a QA system to link textual documents to its core knowledge base, enabling it to extend its coverage by linking mentions of terms and named entities from arbitrary texts to relevant Wikipedia articles.

By achieving these goals, this research makes several contributions to the fields of open-domain question answering and semantic analysis.

The **first original contribution** of this work is the proposal of a paradigm shift in open-domain QA research in order to broaden the types of questions covered by existing QA systems. This paradigm shift consists of moving away from the textual approach which uses words and information retrieval at the core of the QA architecture, towards a semantic approach which relies on concepts, atomic facts and knowledge representation instead. More concretely, this thesis proposes a novel architecture for semantic QA systems which can address encyclopaedic questions, a generalisation of factoid questions. The architecture contains two main processing phases: *analysis*,

responsible for interpreting questions, identifying ambiguities and finding answers, and *feedback*, responsible for creating an informative response and facilitating effective interaction with the user.

The **second original contribution** of this work is the development of EQUAL, a QA system which implements the analysis phase of the architecture summarised above. The system transforms Wikipedia into an entity-relationship graph which is explored when searching for answers. It makes use of both textual information and semi-structured data when validating answers. To demonstrate that the approach is suitable for dealing with encyclopaedic questions the system is independently tested in two competitions alongside both standard textual QA systems and semi-automatic submissions.

The **third original contribution** of the thesis is the development of a new semantic analysis task which transforms unstructured raw text into semi-structured information by linking mentions of entities or terms to relevant Wikipedia articles. This task, called *densification*, is related to the popular wikification method of Mihalcea and Csomai (2007). However, while the latter is focused on identifying a few high-confidence entity disambiguations, densification is not limited to prominent mentions and aims instead for a more complete semantic analysis in which most entities and terms are linked to relevant Wikipedia articles. This method allows any document collection to be connected to a semi-structured knowledge base extracted from Wikipedia and enables semantic QA systems to detect and (in)validate facts exploiting additional document collections. The impact of this method when adopted by EQUAL is also evaluated.

## 1.3 Structure

This thesis comprises three parts. The **first part**, represented by Chapter 2, provides the background for the thesis, offering an introduction to question answering and an overview of Wikipedia-based NLP tools and resources which are relevant to QA research. Chapters 3 and 4 constitute the **second part** of the thesis which argues for a

different approach in open-domain QA to address limitations of the standard approach. Encyclopaedic questions are advocated as key challenge for open-domain QA research. A new architecture is then proposed which represents a paradigm shift from the textual approach, built around words and based on information retrieval, to a semantic approach, built around concepts and based on knowledge representation. This novel approach is implemented by EQUAL, a QA system developed as part of the thesis, which was tested in two QA competitions, GikiP and GikiCLEF. Chapter 5 represents the **third part** of the thesis and investigates the problem of transforming unstructured text into semi-structured information that can be used by encyclopaedic QA systems. A new semantic analysis task to link ambiguous mentions of entities and terms from a text to relevant semantic concepts represented by Wikipedia articles is proposed and then evaluated by measuring its impact on EQUAL.

**Chapter 2** provides a review of previous work relevant to question answering, focusing mainly on NLP resources which have been developed based on Wikipedia. The chapter starts with a brief overview of the three main types of QA systems, then discusses the standard approach employed by current open-domain systems, and presents the architecture typically employed by those systems which address primarily factoid questions. This thesis proposes the use of Wikipedia as source of semantic information for QA systems. The structure of Wikipedia is described and several NLP-related studies are reviewed. These studies cover a variety of fields such as information retrieval, information extraction, knowledge representation and semantic relatedness, and reveal the potential of the encyclopaedia as a source of both textual and semi-structured information.

**Chapter 3** introduces the notion of textual QA to refer to the typical open-domain QA approach which focuses on factoid questions and relies on an information retrieval engine to retrieve text snippets that are very similar to the questions posed, then extracts and ranks the most likely answer candidates using some form of aggregated confidence score. This thesis argues that the applicability of textual QA is limited to simple questions whose answers are explicitly present in small textual snippets, and that as a result, it

fails to address realistic user needs. A shift of focus is suggested to overcome these limitations, from factoid questions towards more generic question types. One such type, encyclopaedic list questions, is advocated as the next key challenge in open-domain QA research. A generic, high-level architecture for semantic QA systems is proposed to enable systems to address not only encyclopaedic questions but also to pave the way for more intelligent systems which use automatic inference to answer other types of complex questions.

**Chapter 4** describes EQUAL, a QA system which implements the approach proposed in Chapter 3. Built around concepts rather than words, it transforms Wikipedia into an entity graph which it then explores to find and validate answers. The system detects different types of ambiguity and creates multiple interpretations corresponding to different understandings of the question. A question interpretation consists of a decomposition of the question into constituents which are then assigned to coarse-grained semantic constraints involving entities, types, relations and properties. Instead of retrieving paragraphs, EQUAL explores Wikipedia as an entity graph to determine which entities are correct answers for a particular interpretation and enforces constraints using structured, semi-structured and textual resources. The chapter also reports the evaluation of the system in two competitions, GikiP (Santos et al., 2008) and GikiCLEF (Santos and Cabral, 2009a). The results of the system in these contests are presented, and the challenges facing this approach are then discussed.

**Chapter 5** addresses the problem of converting unstructured text into semi-structured information needed by the QA approach proposed in Chapter 3 and implemented in Chapter 4. As the architecture centres around concepts rather than words, in order to exploit new document collections a system first needs to perform a semantic analysis of those documents to link their contents to the reference knowledge base used by the system. The chapter describes existing tools which link entity mentions in arbitrary text to the Wikipedia entities to which they refer. As these tools usually focus on a few prominent entities, the task of densification is proposed, which aims to link all mentions

of entities or terms to relevant Wikipedia articles. An experiment involving human raters is carried out to determine the feasibility of creating a large-scale gold standard for this task and a densification system is developed and evaluated in the context of the EQUAL system.

**Chapter 6** summarises the contributions of this research, discusses the extent to which the goals of the thesis have been achieved and provides a review of the thesis. It then draws conclusions and indicates directions for future research.

## Chapter 2

# Question Answering and Encyclopaedic Knowledge

### 2.1 Overview

This chapter provides background information to contextualise the research presented in the thesis, introducing the area of question answering and Wikipedia as a relevant resource. Section 2.2 starts with a brief overview of the question answering domain and a classification of QA systems. The focus then moves to standard open-domain QA systems, the most relevant type of system for this research, with a description of their typical architecture. In this thesis, it is argued that Wikipedia, the world's largest encyclopaedia, is a pivotal resource which can enable QA systems to address more complex question types than those currently addressed by existing approaches. Section 2.3 describes its structure, and Section 2.4 reviews a range of natural language processing studies which reveal its potential as a source of both textual and semi-structured information with direct application to question answering.

### 2.2 Question answering

Question answering is a specialised type of information retrieval (IR) in which systems must return an exact answer to a natural language question. It uses complex natural language processing techniques to process a question, search for the required information

in a collection of documents and then extract the exact answer and present it to the user, usually accompanied by the textual snippet where it was found. According to Maybury (2004a, page 3),

*“Question answering is an interactive human computer process that encompasses understanding a user’s information need, typically expressed in a natural language query; retrieving relevant documents, data, or knowledge from selected sources; extracting, qualifying and prioritizing available answers from these sources; and presenting and explaining responses in an effective manner.”*

The amount of information available on the Internet is ever-increasing: it grew from 26 million Web pages in 1998 to over 1 trillion in 2008, when several billion new pages were being created every day (Alpert and Hajaj, 2008). This exponential growth means that increasingly advanced search tools are necessary to find relevant information. To this end, QA was proposed as a solution for users who search for specific pieces of factual information rather than whole documents. Developing an automatic QA system is not a new task, with the earliest efforts dating from around 50 years ago; for example BASEBALL (Green et al., 1961) answered natural language questions about baseball league results. The task is still very challenging: building such a system requires many resources and complex natural language processing tools, such as named-entity recognisers, coreference resolvers, word sense disambiguators and temporal and spatial reasoners. As these tools have evolved and more resources have become available, increasingly sophisticated systems can be developed. The next sections of the chapter present the main types of QA systems (Section 2.2.1), and then examine the standard approach adopted in the development of the QA systems most relevant for this research (Section 2.2.2).

### 2.2.1 Types of QA systems

Depending on the target domain and the way questions are answered, there are three main types of QA systems: canned, closed-domain and open-domain (Harabagiu and Moldovan, 2003).

**Canned QA Systems** are the simplest type of QA systems because they do not answer questions automatically, but instead rely on a very large repository of questions for which the answer is known. Extending coverage with new questions and their answers relies on human effort. To answer unseen questions, systems usually retrieve the answer of the most similar existing question. They can be useful in restricted domains, where users' information needs are predetermined, e.g., a help desk.

This type of system originated from lists of Frequently Asked Questions and bulletin board systems, such as FidoNet and UseNet. Web 2.0 technologies enabled the development of social Q&A Forums, such as Yahoo!Answers<sup>1</sup>, WikiAnswers<sup>2</sup>, Stack Exchange<sup>3</sup> and Quora<sup>4</sup>. In these collaborative systems, users post new questions and contribute answers to existing ones, thus broadening the coverage of the systems. To increase recall and eliminate inconsistencies caused by duplicates, each question may have an associated set of alternative phrasings. Users also rate the correctness and utility of answers. These repositories are usually available on the Web and users can simply employ their favourite Web search engine to find out if their question already has an answer. Their advantage is that most questions and their answers cover complex information needs which require human-like cognitive skills.

The main contribution such systems make to QA research is the amount of data they provide, which could be used in developing and testing automatic QA systems. They can also be exploited to investigate question similarity metrics to establish when different

---

<sup>1</sup><http://answers.yahoo.com/>

<sup>2</sup><http://wiki.answers.com/>

<sup>3</sup><http://stackexchange.com/>

<sup>4</sup><http://www.quora.com/>

questions ask the same thing. A good similarity metric can distinguish whether two questions are simply alternates, for example when the word order differs or when synonyms or rephrasing are used, or whether the questions ask for different information.

**Closed-domain QA Systems** are built for very specific domains and exploit expert knowledge which is embedded in them, usually in the form of hand-written rules or purpose-built ontologies. The first QA systems of this type were developed in the 1960s and answered questions concerning information included in a database. Any data not present in the database is usually considered out-of-domain.

The advantage of this type of system is that it deals with very specific data which usually does not contain ambiguous terms and as a result can be processed more easily. For example, in a naval domain the word *bow* refers to *the front of a ship*, whereas generally, *bow* can also be *a weapon*, *a tied ribbon* or *an action*. When modelling a specific domain, the relevant concepts and the relations between these concepts are predetermined and explicitly stored, thereby reducing the number of possible interpretations that a question might have. Due to lexical and syntactic constraints within specific domains, questions are actually phrased in a sub-language. Users of such systems are usually trained to phrase natural questions in the format accepted by the system.

Natural language interfaces to databases are a good example of closed-domain QA systems. Instead of learning a complex query language such as SQL<sup>5</sup>, users are able to interrogate the data using natural language and the system automatically converts the user's question into a complex database query. Despite their usefulness, the development of such systems is expensive, they are schema-specific and difficult to extend to a broader domain (Mollá and Vicedo, 2007).

Two of the most cited closed-domain QA systems are BASEBALL (Green et al., 1961, 1986) and LUNAR (Woods, 1977).<sup>6</sup> Developed in the 1960s, BASEBALL answered questions

---

<sup>5</sup>Structured Query Language, commonly used for retrieval and management of data in relational database systems

<sup>6</sup>For a comprehensive overview of QA over restricted domains see Mollá and Vicedo (2007)

about one season of the American baseball league, such as: *Who did the Red Sox lose to on March 5?* or *How many games did the Yankees play in May?* or *On how many days in August did eight teams play?* (Hirschman and Gaizauskas, 2001). According to Mollá and Vicedo (2007), although the textual analysis of the question was complex, it exploited the fact that all the data was stored in one single database table. This enabled a mapping from the semantic interpretation of the question to a database query. The user had to be aware of the kind of data that was available in the database when formulating the question, otherwise the question could not be answered. Given a question like *Which players scored a home run on Independence Day?*, the system would not be able to answer it because the fact that the *Independence Day* of the United States is celebrated on the 4<sup>th</sup> of July is not part of the domain. Information about the meaning of *Independence Day* can be added, but this question demonstrates that even closed domain systems need various types of world knowledge to correctly understand and interpret some questions.

LUNAR answered questions about the geological analysis of rocks returned by the Apollo moon missions, such as *What is the average concentration of aluminium in high alkali rocks?* or *How many Brescias contain Olivine?* The system was demonstrated at a lunar science convention and was able to correctly answer 90% of the in-domain questions posed by geologists (Hirschman and Gaizauskas, 2001). This high performance can be explained by the specificity of the domain which allowed the system to correctly answer questions posed by domain experts even though they did not receive prior training.

Today, domain knowledge is represented not only in structured databases, but also in semantic networks and ontologies (see Section 2.4.4), using standards for knowledge representation such as RDF<sup>7</sup> and OWL<sup>8</sup>. An example of a closed-domain QA system is provided by QALL-ME<sup>9</sup>, a project which developed a multilingual and multimodal QA system in the tourism domain, able to answer questions such as: *Where can I eat*

---

<sup>7</sup><http://www.w3.org/RDF/>

<sup>8</sup><http://www.w3.org/TR/owl-features/>

<sup>9</sup>Question Answering Learning technologies in a multiLingual and Multimodal Environment <http://qallme.fbk.eu/>

*paella tonight?, In which cinema in Birmingham is Avatar on this weekend? and I want to know a hotel in Wolverhampton that has wireless internet in the rooms and is close to the train station.* The domain is modelled using an OWL ontology<sup>10</sup> and data is stored in RDF format. Given a question, the system uses an advanced semantic textual entailment engine to select the most appropriate *question patterns* (Săcăleanu et al., 2008). Each pattern corresponds to a database query procedure that retrieves the relevant data. As well as the answer itself, the system provides the user with additional multimodal information such as city maps, routes, timetables, film trailers and images.

Ontologies and technologies developed for the Semantic Web, make it possible to develop portable closed-domain systems, such as AquaLog (Lopez et al., 2007), an ontology-driven question answering system which can be quickly adapted to new, albeit limited, domains by changing the underlying ontology. The system is noteworthy in that it uses natural language processing to extract triples from a question which are then matched to the terminology used by the domain ontology. Systems which automatically generate a domain lexicon by processing ontology lexicalisations are also known as portable QA systems. Unlike earlier systems where the mapping had to be performed at design time, this system uses PowerMap, an algorithm which performs the mapping at run time (Lopez, Sabou and Motta, 2006). The interactive interface of AquaLog allows users to select the best match when more than one alternative is possible, and these mappings are automatically added to the lexicon. This allows the system to quickly learn how to transform natural language questions into structured queries. AquaLog makes a priori assumptions regarding the relevance of the ontology to all queries, essentially using a controlled natural language for the questions.

Other applications are able to use multiple ontologies by dynamically exploiting the Semantic Web as a large scale source of distributed knowledge (d'Aquin et al., 2008). Multiple, heterogeneous online ontologies can be automatically analysed to harvest

---

<sup>10</sup><http://qallme.fbk.eu/index.php?location=ontology>

knowledge to derive ontology mappings (Sabou et al., 2008). This enables ontology-driven QA systems to break the confines of a single ontology and instead find answers in the entire Semantic Web. PowerAqua (Lopez, Motta and Uren, 2006) is a QA system designed to exploit semantic markup on the Web to provide answers to questions posed in natural language. The system transforms the questions into a set of logical queries which are then answered separately by consulting and aggregating information derived from multiple heterogeneous semantic sources. A more recent version of the system (Lopez et al., 2009) uses a Triple Similarity Service to match queries to triples found in different ontologies of the Semantic Web.

Creating QA systems for the Semantic Web means tackling several non-trivial technical issues (Lopez et al., 2011). This provides an interesting opportunity to develop ontology-based QA systems for Linked Open Data (LOD). The challenge is to automatically convert a natural question into a form which can be expressed using Semantic Web technologies for query processing and inferencing. QALD1<sup>11</sup> is a recent evaluation campaign aimed at scaling ontology-based QA to Linked Data scale. The main challenges involve dealing with a heterogeneous, distributed and huge set of inter-linked data.

FREyA is an interactive natural language interface for querying ontologies (Damljanovic et al., 2010, 2011). Unlike PowerAqua which automatically maps and includes new ontologies at query time, FREyA tries to limit complexity by only using a limited number of ontologies, because the more specific to the domain a system is, the better the performance it achieves. Although it qualifies as a closed-domain system, it is of special interest here as it uses an approach which shares similarities with the open-domain architecture (Dornescu, 2009) presented in the following chapter. A linguistic analysis of the question employing a syntactic parser yields a set of phrases (POCs) which are possible mentions of concepts from the underlying ontology, such as classes, properties, instances, property values, etc., using a rule-based heuristic method. An ontology look-up algorithm matches the POCs with actual ontology concepts and a consolidation algorithm

---

<sup>11</sup><http://www.sc.cit-ec.uni-bielefeld.de/qald-1>

creates a SPARQL query. When the system fails due to ambiguity or an incomplete lexicon, it prompts the user for clarifications. A learning module enables the system to learn from these interactions in order to produce confidence scores. The system achieved the best results in the QALD1 competition, with 27 of 50 questions interpreted correctly.

**Open-domain QA Systems** can be asked about virtually any topic and can theoretically extract the answer from any textual collection, in contrast to the previous types of systems. They evolved as an alternative to search engines, as more and more information became available on the Web.

Existing search engines excel at retrieving a ranked list of documents which are relevant to a keyword query, but they are unable to answer questions.<sup>12</sup> When searching for a specific piece of information rather than relevant documents, the user needs to construct complex queries, trying to guess a probable context of the answer and repeatedly refining the query until the most relevant text snippets returned contain the information sought. In contrast, in open-domain QA, instead of providing a list of relevant keywords – ideally a *perfect* query – the user just asks a question. Radev et al. (2005) claims that it is more natural for a user to type a question like: *Who wrote King Lear?* than to create a query such as: (wrote OR written OR author) AND (“King Lear”). As shown in Section 2.2.2, a QA system will automatically build queries and retrieve relevant documents, extract answers from the retrieved text snippets and present them as a confidence-ranked list to the user.

The majority of open-domain QA systems search for answers in large textual repositories rather than in databases or knowledge bases as is the case with closed-domain QA. Usually systems are developed and tested on a particular document collection. The documents may come from heterogeneous information sources ranging from newswire

---

<sup>12</sup>Some search engines (such as Google) have started to add QA capabilities for answering simple questions, e.g., *Who is the prime minister of Romania?*. However these capabilities are not clearly advertised or are presented as experimental features and their coverage is limited, resembling canned QA rather than open-domain QA.

corpora to encyclopaedias and the World Wide Web, as long as the system can convert them to plain text.

Open-domain QA is viewed as the “holy grail” of information retrieval as it requires complex text comprehension which is a difficult problem. Current open-domain QA systems can be asked about any topic, as long as the complexity of the question is moderate, i.e., understanding both the question and the text necessary to extract the answer can be simulated or approximated. Questions that require the systems to “reason”, e.g., extract and combine information from several documents or deduce information using common-sense inference, are considered difficult. Despite substantial research carried out in the last decade, open-domain QA systems have not become an alternative to Web search engines. This can be explained by the standard approach typically employed by open-domain QA systems, which is presented in Section 2.2.2.

The first Web-based QA system was START<sup>13</sup> (SynTactic Analysis using Reversible Transformations) by Katz (1997). It uses complex NLP modules that transform English sentences into *ternary expressions* (T-expressions):  $\langle \text{subject}, \text{relation}, \text{object} \rangle$ . These expressions are added to the knowledge base in the form of annotations of text documents. By applying similar processing to the question, the system extracts template expressions which it then uses to find answers in the annotated knowledge base. Hand crafted rules are added to the system in order to better deal with lexical variations. The system does not perform any form of reasoning: answering is limited to matching expressions from the question to T-expressions extracted from a single sentence. For example, START knows all the capitals in Africa, and for each one it also knows the population size. However it cannot deal with input such as: *Capitals in Africa and their population* or *Which capitals in Africa have population over 100,000*, because it does not have this exact answer in the knowledge base, and this question requires more processing than simply matching an S-expression.

---

<sup>13</sup><http://start.csail.mit.edu/>

**Other approaches** in QA are evidenced by a number of new systems which have become known very recently, for example, systems which offer natural language based search interfaces to structured knowledge, typically based on data that is manually coded and homogeneous (e.g. Wolfram Alpha<sup>14</sup> and Powerset<sup>15</sup>) or proprietary (e.g. TrueKnowledge<sup>16</sup>). Another type of QA system, made popular by the smartphone industry, is the virtual personal assistant: a tool which helps you achieve certain things using your mobile just by asking. It allows you to perform actions such as sending messages, scheduling meetings, making phone calls, and more, by issuing spoken instructions (Sadun and Sande, 2012). Probably the most widely publicised success story in the domain of open-domain QA is IBM's Watson (Ferrucci, 2012), the system built for the American *Jeopardy!* quiz, which defeated the two highest ranked human champions in a nationally televised two-game match.

True Knowledge (Tunstall-Pedoe, 2010) is a natural language search engine and question answering site, which maintains a large knowledge base of facts about the world and allows users to contribute facts and knowledge as well as to provide answers to questions. The knowledge is stored in a form allowing automatic reasoning which enables the system to combine knowledge by inferring new facts and cross-referencing existent information to produce a reasoned answer together with a detailed justification which displays the rules that were applied in finding the answer. For each piece of discrete information the system can determine where it comes from, when it was created, the confidence that it is true and even in what spans of time it is correct. Facts come in two types, source facts stored in the knowledge base and inferred facts, created by the knowledge generator via applying inference rules to existing facts. A key feature of the approach employed by TrueKnowledge is that users can not only add and curate facts, they can also extend the questions that can be addressed and even provide new inference rules for the knowledge generator (including code for steps that involve calculations).

---

<sup>14</sup><http://www.wolframalpha.com>

<sup>15</sup><http://powerset.com>

<sup>16</sup><http://www.evi.com>

Whilst free datasets such as Freebase and DBpedia (see Section 2.4.4) are incorporated, the system's knowledge base is not publicly available. However, API access is provided, enabling third parties to build applications using the True Knowledge technology.

Watson (Ferrucci, 2012) is the result of the challenge taken on in 2007 by IBM Research: that of building a computer system that could compete with human champions at the game of *Jeopardy!*. The quiz show requires participants to quickly understand and answer natural language questions (*clues*), which cover a very broad range of topics. The competition penalises inaccurate answers, therefore the system needs to reliably compute a confidence score and refrain from answering questions when this score is not high enough. To be competitive, the aim was to achieve *85% Precision@70* which means that Watson is expected to be confident enough to buzz in for 70% of the questions, and it should answer correctly at least 85% of these questions, spending mere seconds to answer each question.

The QA system PIQUANT (Chu-Carroll et al., 2005), which IBM had previously built for TREC competitions, only achieved roughly *16% Precision@70* and needed more than two hours to answer each question, when using a single machine. The DeepQA architecture (Ferrucci et al., 2010) was therefore proposed as a highly parallel approach which would allow the processing to be accelerated by orders of magnitude, but would also enable the researchers to combine multiple technologies to answer questions much more reliably. 25 full-time researchers participated in the endeavour and after almost 5 years and more than 8000 individual experiments involving tens of thousands of questions, Watson made history defeating the highest ranked players. Watson is specialised for the type of clues used in *Jeopardy!* and it would probably need to be re-configured in order to achieve similar performance on other question types. However, it is, without a doubt, an extremely impressive QA system.

The DeepQA architecture used by Watson defines four main stages of analysis. Each stage has multiple implementations which can produce alternative results. The alternatives are processed independently in a massively parallel computation. Evidence is gathered and

analysed for each answer and each alternative. Statistical machine learning algorithms are used to aggregated all evidence into a single confidence score. In the first stage, the system analyses the question type and its topic (Lally et al., 2012), and then uses an English Slot Grammar parser and predicate-argument structure generator (McCord et al., 2012) to prepare deeper linguistic information for subsequent steps.

The next stage is hypothesis generation. An analysis of tens of thousands of Jeopardy! questions used in the past revealed that although more than 95% of the answers have a page in Wikipedia, only 2% of answers could be found using DBpedia alone, which means that the system must use all available data sources simultaneously. Instead of searching for the correct answer directly, the system identifies a large set of candidate answers (Chu-Carroll et al., 2012). Each candidate answer is then combined with the question to create an independent hypothesis.

The system then searches for evidence supporting each of the hypotheses, independently examining all of the information sources available (encyclopedias, dictionaries, thesauri, newswire articles, and literary works, databases, taxonomies and ontologies, specifically, DBpedia, WordNet, and Yago). The most important types of evidence sought are the expected answer type (Murdock et al., 2012) and relation extraction and validation (Wang et al., 2012). In the final stage, a statistical machine learning approach is used to weigh and combine scores, using various evidence scoring algorithms, and a single confidence score is computed for each candidate answer (Gondek et al., 2012). Of course, at the time of the competition Watson ranked in the top 100 supercomputers in the world, which meant that all the stages were usually completed in under 3 seconds.

### 2.2.2 General architecture of open-domain QA systems

Harabagiu and Moldovan (2003) present the standard architecture of an open-domain QA system, governed by the stages it must perform, namely **question analysis**: understanding what the question is asking, **document retrieval**: finding relevant textual paragraphs, and **answer extraction**: identifying the exact answer in one of the retrieved

paragraphs. For a more comprehensive description of the architecture with further details regarding individual components see Hirschman and Gaizauskas (2001). The exact architecture of specific systems can be quite complex, accommodating features such as cross-linguality, e.g., when a question is asked in French, but the answer is searched for in an English corpus, answer fusion, when an answer is extracted from several documents, or Web ranking, when candidate answers are first searched for and identified on the Web and then they are located and ranked in the target collection.

**Question analysis** is concerned with processing input questions and determining the question type, the expected answer type, the question focus and what other entities and/or keywords are present. The **question type** is selected from a question taxonomy that the system uses for example, **definition** questions: *What is the Top Quark?*, **factoid** questions: *When was X born?*, **yes-no** questions: *Is Barack Obama the US president?*, **list** questions: *Give me capitals of Africa*. Hovy et al. (2000) proposed a semantic taxonomy of 94 types based on the user's intention. Each question type has a specific set of patterns which are used to extract answers.

Another feature determined at this stage is the **expected answer type** (EAT) such as PERSON, LOCATION, TIME-DATE, which is used when processing the text snippets retrieved. Only the snippets containing an entity of the expected type will be processed further. Li and Roth (2002) use a two-stage classifier which assigns each question one of 6 coarse-types and one of 50 sub-types of EAT (see Table 2.1). They built a dataset comprising 5500 questions for training and 500 for testing, achieving 91% classification accuracy for coarse types and 84.2% accuracy for subtypes. Techniques usually employed by QA systems for classifying questions range from regular expressions and hand-crafted rules (Hermjakob, 2001), to language models (Pinto et al., 2002), support vector machines (Zhang and Lee, 2003) or maximum entropy (Le Nguyen et al., 2007). The models use features such as words, n-grams, syntactic information from parse trees, and semantic types from Word-Net.

The **question focus** usually indicates which entity the question is about, e.g., *Bush* in *When was Bush born?*. Keywords and entities are also extracted in order to construct

Table 2.1: Expected answer type classification scheme proposed by Li and Roth (2002), commonly used by TREC participants.

| Coarse EAT   | Fine EAT subtype  |
|--------------|---|
| ABBREVIATION | abbreviation, expression abbreviated  |
| ENTITY       | animal, body, color, creative, currency, diseases and medical, event, food, instrument, lang, letter, other, plant, product, religion, sport, substance, symbol, technique, term, vehicle, word |
| DESCRIPTION  | definition, description, manner, reason   |
| HUMAN        | group, individual, title, description   |
| LOCATION     | city, country, mountain, other, state   |
| NUMERIC      | code, count, date, distance, money, order, other, period, percent, speed, temperature, size, weight   |

queries that will be sent to an information retrieval engine. NLP tools, such as part of speech taggers, syntactical parsers, shallow parsers and semantic role labellers, are employed in order to create a complex representation that encodes the meaning of the question, as interpreted by the system.

**Document retrieval** has as input the question representation built in the previous stage. It constructs queries corresponding to the question and uses a search engine to identify relevant text snippets in the target collection. The IR engine can make use of different types of linguistic annotations. When a small document collection is used, their text can be annotated during indexing with lexical information (words, stems, lemmas), syntactic information (chunks, multi-word expressions, dependency trees), named entities annotations, semantic role labels or discourse structures. These can be used at the retrieval stage for improving the relevance of the retrieved data. When dealing with large or dynamic collections (such as the Web) it is impractical to perform these annotations on all documents, thus the more complex analysis is performed only on the sub-set of documents retrieved. Usually retrieval is a multi-stage process: at first, full documents are retrieved using simple techniques, and then, increasingly more complex techniques are applied to smaller text snippets, usually paragraphs or even sentences.

**Alternations** can be used to create more advanced queries (Harabagiu and Moldovan,

2003). **Morpho-lexical alternations** are employed to increase recall: using resources such as WordNet (Fellbaum, 1998) the query is expanded with synonyms and morphological variations of the keywords, e.g., singular and plural forms. Another form of alternation is rephrasing the question, e.g., from the question *When was X born?* the phrase query “*X was born in*” is generated. These *semantic alternations* allow more precise retrieval due to information redundancy in large collections such as the Web, where it is probable to find such rephrased answers using a small set of high-precision, low-recall patterns.

**Answer extraction** is the third step performed by this type of QA system. At this stage, the retrieved snippets are processed and candidate answers are extracted using all the information gathered in the previous steps, e.g., keywords, entities and relations. Documents that do not have a candidate of the expected answer type are discarded. The remaining candidate answers are then filtered and ranked according to different measures of similarity to the question (Harabagiu et al., 2003; Moldovan et al., 2007).

Pattern-based extraction is one of the most common ways to extract and rank answers for certain types of questions. For example, given the question *What is the population of London?* the system can use templates such as “*the population of <?city> is <?number> inhabitants*”, “*<?city> has a population of <?number>*”, “*<?city> has <?number> inhabitants*”. These template patterns can be automatically acquired by training the system on known *<question, answer>* pairs. For example, if the system knows that the correct answer to the question *What is the population of London?* is 7,855,600, then it can automatically extract patterns that would help find the answer to the similar question *What is the population of Birmingham?* Usually these patterns are learned when building the system, by using training data (Schlaefter et al., 2006; Schlaefter, 2007).

**Answer filtering** uses the EAT and pre-compiled lists of entities derived from ontologies, gazetteers and encyclopaedias to filter out candidates with incompatible types. As more of these resources are employed and their coverage is expanded, the number of eponymous entities also grows and the semantic ambiguity of entities is also increased,

which can negatively impact overall performance (Schlobach et al., 2004; Prager et al., 2003). The most relevant answers are selected using different measures such as word similarity, e.g., the *tfidf* score of the exact matched words, the score of the unmatched words, the number of adjacent words in both the question and the snippet, the average distance between the candidate answer and the question words, and syntactic similarity between the parse tree of the question and that of the phrase containing the candidate answer.

**Answer re-ranking** is the stage when the ranked list of answers is computed. Nyberg et al. (2003) uses a clustering algorithm to merge evidence from several documents which supports a similar answer. For each cluster a representative answer is selected and an aggregated confidence score is computed. Ranking is more complex when multiple strategies are employed (Jijkoun et al., 2006), as each strategy has different score distributions. To combine and re-rank the answers found by multiple QA agents, a simple confidence-based voting is used by Chu-Carroll et al. (2003), a maximum-entropy model was proposed by Echiabi et al. (2004) and a successful probabilistic framework for answer selection is employed by Ko et al. (2010).

To improve accuracy, **answer validation** methods can be used to determine the correctness of the extracted answer. Both the question and the answer are generally transformed to a deeper representation (logical form) and then formal methods such as theorem proving algorithms try to assess the connection between the two, by using information and facts contained within the documents, world knowledge (which is often extracted from WordNet), and domain specific ontologies. For example, textual entailment and logic formalisms are used by Moldovan et al. (2007), while a simpler alternative is to use the edit distance between the question and the answer as a measure of similarity.

Systems typically use the Web as an additional resource for both finding and validating answers, because vast amounts of data means greater coverage, greater redundancy and greater chances that simple patterns previously observed during system development

will also occur on the Web: *Where is the Eiffel Tower located?* → “*The Eiffel Tower is located in*”. These simple lexical patterns exploit the redundancy of information on the Web: presumably the answer to most factoid questions is present in multiple documents and formulations (Clarke, Cormack and Lynam, 2001). Answers that are found on the Web are then searched for and validated in the target collection.

To illustrate the whole process with an example of a textual QA system, FALCON was one of the top-performing systems in the TREC 9 and TREC 10 campaigns (Harabagiu et al., 2000, 2001). In the first stage, question analysis, the question is processed to determine the expected answer type and to extract a list of keywords used for retrieval. In the second stage, document retrieval, an IR engine which supports boolean operators is used to retrieve paragraphs from the TREC document collection. A feedback loop modifies the query if the search space becomes either too restrictive or too broad, in order to select roughly 500 paragraphs. These are passed on to the last stage, answer extraction, which is concerned with extracting the answer. Paragraphs that do not contain a candidate answer of the expected type are discarded. The remaining paragraphs are parsed and head-modifier relations are extracted from their parse trees. Lexico-semantic unifications are tried between the features of the answer and those of the question, allowing linguistic variations, such as different morphological forms, synonyms, or rephrasing. A confidence score is assigned for each candidate answer snippet based on features such as the number of words present in the question and the compactness of the answer. A ranked list of answers is returned by the system.

### 2.2.3 QA and Web search

Open-domain QA research has primarily focused on factoid questions: those questions whose answers are named entities, concepts, measures, temporal expressions, etc., and on description questions: definition, why, relation. We call these types of systems **textual QA systems** because the core technology is not based on understanding meaning, i.e., using automatic inference or formal reasoning, but on searching for the most relevant

paragraphs in a large textual collection by relying primarily on lexical similarity. Such systems have been intensely studied in the past decade, but despite efforts they have not fulfilled the promise of replacing Web search engines. These engines have themselves evolved significantly and users can usually search and find answers to factoid questions quite easily, which has limited the impact of textual QA systems on the way users access information. In addition, some description questions, such as definitions and how questions, can be addressed by Web search engines which index existing Q&A sites. They also offer their users a familiar interface as a ‘universal’ search tool. Current ranking methods employed by search engines rely less on keywords and more on users’ behaviour: clicking a search result is equivalent to assigning a tag to that particular URL. In effect, a Web search engine allows a semantic search based on the tags assigned by users during searching rather than a lexical search based on the words in the document. Because of the capabilities of Web search engines, QA systems need to robustly address more realistic questions if they are to provide a real alternative.

To address more complex information needs and to deliver more accurate answers, semantic knowledge should be embedded at every stage of the QA pipeline. In the following section, Wikipedia<sup>17</sup> is presented as a core multi-faceted resource which is already fostering significant advances throughout the NLP research landscape and which can provide semantic world knowledge for QA systems.

## 2.3 Wikipedia: a collaborative encyclopaedia

Wikipedia is a free online encyclopaedia supported by the non-profit Wikimedia Foundation.<sup>18</sup> It contains more than 250 language editions totalling over 20 million articles that are collaboratively written and maintained by volunteers. About 100,000 active users make regular contributions, but anyone with Internet access can edit (almost) all articles. Since its launch in 2001, it has become the largest and most popular general

---

<sup>17</sup><http://www.wikipedia.org/>

<sup>18</sup><http://wikimediafoundation.org/wiki/Home>

reference work on the Internet. For QA, Wikipedia is interesting not only because of the large amount of encyclopaedic articles that are continuously updated, but also because of structural properties that make it easier to process by computers than generic Web content.

As an **encyclopaedia**, Wikipedia has attracted criticism regarding the accuracy of its articles and the information bias in terms of coverage<sup>19</sup>. Many consider that the free-collaboration edit policy led to inaccurate, unreliable content. However, in a controversial article, Giles (2005) showed that the errors are comparable to the ones in Encyclopaedia Britannica<sup>20</sup>, and concluded that, due to the coordination and organisation efforts, the quality of Wikipedia articles is on a par with standard printed encyclopaedias. A review of articles in the field of pathology found that Wikipedia's coverage were of a high standard (Wood and Struthers, 2010). While the accuracy is acceptable (due to the many edits a certain level of consensus is achieved), the coverage is biased by the interests of the contributors. General concepts and recent events tend to be better represented with more detailed information and more frequent updates. Wikipedia is one of the most visited domains on the Web, currently ranked sixth globally among all websites on Alexa<sup>21</sup> and has an estimated 365 million readers worldwide.

As a **corpus**, Wikipedia is of great interest. While large amounts of text can be crawled on the Web, Wikipedia articles are much better suited for creating corpora due to their structure and uniform style, their contents being peer reviewed and constantly updated in order to conform to the guidelines. Different language versions are aligned, thus Wikipedia is a well suited collection for multilingual applications. The way information is structured and stored is influenced by its purpose as an encyclopaedia, and by the wiki software<sup>22</sup> it uses. Wikipedia is a large collection of interlinked documents called **pages**. According to their content, each page is part of a **namespace**.<sup>23</sup> The most important

---

<sup>19</sup>[http://en.wikipedia.org/wiki/Reliability\\_of\\_Wikipedia](http://en.wikipedia.org/wiki/Reliability_of_Wikipedia)

<sup>20</sup><http://www.britannica.com/>

<sup>21</sup><http://www.alexa.com/siteinfo/wikipedia.org?range=5y&size=large&y=t>

<sup>22</sup><http://www.mediawiki.org/>

<sup>23</sup><http://en.wikipedia.org/wiki/Wikipedia:Namespace>

namespaces for QA research are *Main*, *Categories* and *Templates*, while the less relevant ones include *Talk*, *User*, *User talk*, *Portal*, *Wikipedia* and *Help*. The namespace is usually prefixed to the page title creating a **unique identifier** for each page, e.g., *Liverpool F.C.*, *Category:English football clubs* or *Template:Football club infobox*. In the remainder of this section the structure of Wikipedia is described to contextualise terminology used throughout the thesis and to present ways in which the encyclopaedia can be exploited in NLP applications relevant to QA research.

## MAIN NAMESPACE namespace<sup>24</sup>

The pages from the *Main namespace* are those containing encyclopaedic content. Depending on what they contain, there are five page types:

- **articles** represent the majority of pages in the Main namespace. According to the Wikipedia guidelines, each article must describe only one concept, the one denoted by the title. The first sentence or paragraph of the article must clearly define what concept the article is about – the definition. Generally, in addition to the definition, each article must contain at least one link to another page, and there must be at least one incoming link to the article.
- **stub articles** only contain a brief description of the concept (its definition) and need to be expanded. They are usually very short (less than 100 words), but contain valuable information such as the classification of the concept. Stubs are clearly marked to facilitate user contributions.
- **redirect pages** are pages whose title represents a common synonym of a concept that is already described in another article. In order to avoid duplicating information, these articles consist only of an automatic redirect directive to the article that contains the actual information, e.g., *Elisabeth 2* and over 90 other articles redirect to *Elizabeth II of the United Kingdom*. Redirect pages usually represent partial names, common misspellings and abbreviations, and therefore

---

<sup>24</sup>[http://en.wikipedia.org/wiki/Wikipedia:Main\\_namespace](http://en.wikipedia.org/wiki/Wikipedia:Main_namespace)

are an excellent source of synonyms for named entities that can be exploited when analysing questions or when extracting answers, where identifying the mention of a known entity is more useful than just recognising that a span of text denotes some unknown entity.

- **disambiguation pages** have a list of links to all the known entities in cases where the same title can denote several distinct entities, for example, *Paris (disambiguation)* has links such as *Paris (mythology)*, *Paris (2008 film)*, *Paris (The Cure album)*, *Paris, New York* or *Paris Simmons*. Disambiguation pages are a source of homonyms and eponymous entities, and they are important for creating a thesaurus.
- **list pages** usually contain lists or tables of links to articles that have a certain characteristic or are part of a category. They are used to organise articles. Their main advantage consists of the fact that they may contain links to articles that do not exist yet, called red links. It is thus possible to identify candidate answers or infer some of their properties even if the corresponding articles have not yet been created.

#### CATEGORY NAMESPACE namespace<sup>25</sup>

These pages, called **categories**, provide a mechanism for structuring and organising the contents of Wikipedia, enabling the automatic creation of tables of contents and indexing functionalities. Each category page contains an *introduction*, which can be edited like any other page, and a *body* containing an automatically generated list of links to its articles and sub-categories. All pages, including categories themselves, must be part of at least one category, e.g., the article *Douro* is part of *Category:Rivers of Portugal*, and *Category:Rivers* itself is part of *Category:Water streams*. Categories form a hierarchical structure but not a taxonomy, since each article can appear in more than one category and each category can have more than one parent category. This allows multiple categorisation schemes to be used simultaneously, e.g., the article *Douro* is also

---

<sup>25</sup>[http://en.wikipedia.org/wiki/Wikipedia:Category\\_namespace](http://en.wikipedia.org/wiki/Wikipedia:Category_namespace)

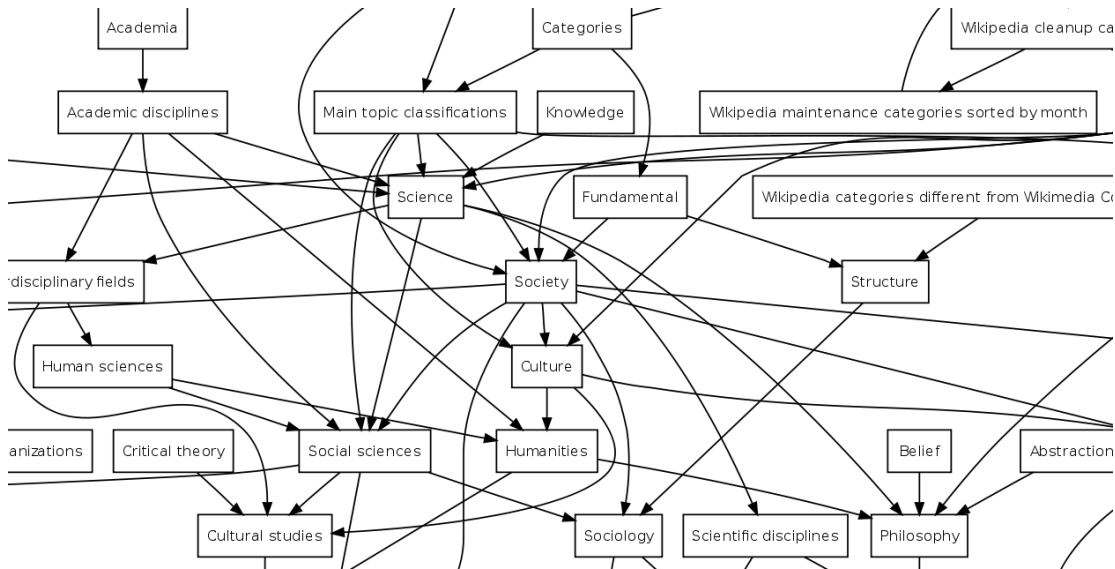



Figure 2.1: Partial view of Wikipedia's Category Graph

part of *Category:Wine regions of Portugal*. The category system in Wikipedia is a result of a collaborative effort to organise articles, i.e., a folksonomy (Vander Wal, 2007; Peters, 2009), which can be automatically processed and linked to ontologies on the Semantic Web (Specia and Motta, 2007).

There are two main types of categories: **topic categories**, which contain articles related to a particular topic, such as *Category:Education* or *Category:London*, and **list categories**, which contain articles of a certain type, such as *Category:Capitals in Europe* or *Category:British actors*. List categories act as a specific gazetteer, useful when searching for answers of a specified type. For example, given the question *Which rivers from Portugal are longer than 500 Km?* the system can automatically identify the *Category:Rivers of Portugal* and only examine the articles that actually describe Portuguese rivers. The information embedded in the Wikipedia category system can lead to much better precision when searching for information than a standard keyword-based IR engine.

## TEMPLATE NAMESPACE `namespace`<sup>26</sup>

Template pages provide information to help readers, such as navigation aids, links to disambiguation pages and editorial notes to help manage the collaborative editing process. They can be used for various purposes including formatting, displaying information consistently, creating navigational boxes and sidebars which link series of related articles together, and also for coordinating and organising the editorial effort.

The wiki software<sup>27</sup> provides a mechanism called *transclusion*<sup>28</sup> to include in a page the content of any **template page**. This helps when updating contents, since only the display of information is redundant (the same content is displayed in several pages) while the storage is efficient because the update only requires the modification of the template page. An important feature of this mechanism is that templates can have parameters. For example `{{flag icon|Romania}}` inserts a small in-line flag of Romania . If the Romanian flag is changed, none of the thousands of pages that display it needs any modification. Another role of the templates is to automatically assign the articles that use them to categories, for example, the template call `{{Birth date|df=yes|1879|3|14}}` results in the output: “March 14, 1879”, and at the same time makes the article *Albert Einstein* part of *Category:1879 Births*.

An **infobox** is a particular type of template that is used to display factual information in a consistent format, usually as a floating table in the upper right region of the article. Infoboxes make extensive use of parameters (see Figure 2.2b) which is why they can be associated with the templates used in Information Extraction. By extracting the values of the parameters used, a system can obtain more reliable data than can be obtained by analysing natural language text. As shown in Figure 2.2b, infobox template calls have a semistructured content that can be exploited to reliably extract information in an automatic fashion. For example the nickname, the stadium and the manager of *Liverpool*

---

<sup>26</sup>[http://en.wikipedia.org/wiki/Wikipedia:Template\\_namespace](http://en.wikipedia.org/wiki/Wikipedia:Template_namespace)

<sup>27</sup>MediaWiki: <http://www.mediawiki.org/>

<sup>28</sup><http://en.wikipedia.org/wiki/Wikipedia:Transclusion>

|   |  |  |
|---|--|--|
| <div> <div> <div>Liverpool F.C.</div> <div>  </div> </div> <div> <div> <div>Full name</div> <div>Liverpool Football Club</div> </div> <div> <div>Nickname(s)</div> <div>The Reds</div> </div> <div> <div>Founded</div> <div>1892</div> <div>(by John Houlding)</div> </div> <div> <div>Ground</div> <div>Anfield</div> <div>Liverpool, England</div> <div>(Capacity: 45,362)</div> </div> <div> <div>Chairman</div> <div> <div> Tom Hicks (co-chairman)</div> <div> George Gillett (co-chairman)</div> </div> </div> <div> <div>Manager</div> <div> Rafael Benítez</div> </div> <div> <div>League</div> <div>Premier League</div> </div> <div> <div>2007-08</div> <div>Premier League, 4th</div> </div> </div> </div> |  | <pre> {{Infobox Football club   current      = Liverpool F.C. season 2008-09   image        = [[Image:LFC.svg 150px Liverpool emblem]]   fullname     = Liverpool Football Club   nickname     = The Reds   founded      = 1892&lt;br&gt;(by [[John Houlding]])   ground       = [[Anfield]]&lt;br /&gt;[[Liverpool]], England   capacity     = 45,362   chairman     = {{flagicon USA}} [[Tom Hicks]] (co-chairman)&lt;br /&gt; {{flagicon USA}} [[George N. Gillett Jr. George Gillett]] (co-chairman)   manager      = {{flagicon ESP}} [[Rafael Benítez]]   league       = [[Premier League]]   season       = [[Premier League 2007-08 2007-08]]   position     = Premier League, 4th }} </pre> |
| (a) rendered infobox  |  |  |

Figure 2.2: Templates in Wikipedia

*F.C.* can be directly extracted from the wiki source text. Infoboxes are a source of  $\langle attribute, value \rangle$  pairs that can be exploited by QA systems and other NLP applications. By employing information extraction techniques, large amounts of structured data can be extracted from Wikipedia in the form of factual databases or ontologies (see Section 2.4.4).

## Hyperlinks

All the pages in Wikipedia are connected using hyperlinks. There are several types of links: wiki links are internal links between articles, inter-wiki links connect corresponding pages from different language versions of Wikipedia, external links are hyperlinks to other websites or Web resources, and category links connect each page to the categories it belongs to. The most important sub-graphs of this large scale directed multi-graph are the Wikipedia article graph (WAG) and the category graph (WCG) a part of which is shown in Figure 2.1.

Wiki links can be exploited in several ways in a QA system. By extracting the anchor text of each hyperlink, a thesaurus of aliases can be created. The anchor text is also a disambiguated reference to a concept, thus the context of a snippet is not only the

surrounding text, but also the pages it has links to. By accessing the linked articles, additional information can be used in the different phases of the QA pipeline. Wiki links are intended for human users, and contributors try to avoid clutter and redundant or irrelevant links. For this reason not all relevant links may be present in an article.

## 2.4 Wikipedia: a resource for NLP research

Since its inception in 2001, Wikipedia has become one of the most accessed sites on the Web. In the past few years Wikipedia has become the target of an increasing number of research projects in domains relevant to question answering, such as information extraction, knowledge representation, information retrieval and Semantic Web technologies (Medelyan et al., 2009). In some domains (including QA) the structure of Wikipedia has just started to be exploited, and so significant improvements can be expected in the near future. Wikipedia is attractive to QA research not only because of the huge amount of continually updated data, but also because of its structure and the many tools and resources that have already been built on top of it.

### 2.4.1 Wikipedia and semantic relatedness

Formal analysis of the Wikipedia graph (Zesch and Gurevych, 2007) revealed that it has the characteristics of **semantic networks** such as WordNet (Fellbaum, 1998). Ponzetto and Strube (2006, 2007b) show that Wikipedia is much more suitable as a semantic network than WordNet due to its broader coverage and rich link structure. Therefore Wikipedia can be used to measure the similarity of words or documents by creating an appropriate similarity measure. Given two words, WikiRelate! (Strube and Ponzetto, 2006) finds two corresponding Wikipedia articles that contain the respective words in their title and then computes the articles' relatedness from the bag-of-words similarity of their texts and the distance between their categories in the Wikipedia category graph.

Explicit Semantic Analysis (ESA), proposed by Gabrilovich and Markovitch (2007), also uses a vector similarity measure, but instead of representing the documents as word-vectors (bags-of-words), the documents are vectors in the Wikipedia article space. They

construct an inverted index from words to the Wikipedia articles they are used in so that given a document, the system constructs an article-vector by aggregating index information for each of its words. The similarity score of any two documents is the cosine similarity of their associated article-vectors, and the authors demonstrate that the system can recognise similar documents even when they have little lexical overlap.

### 2.4.2 Wikipedia and information retrieval

Given its broad information coverage, its structural properties, and its continual updates, Wikipedia appeals to the information retrieval (IR) community for both enhancing existing methods and developing new ones. One such method has been **query expansion**, namely adding synonyms, alternative spellings, abbreviations and aliases, in order to increase recall. Many participants in the TREC HARD Track<sup>29</sup> use Wikipedia to construct a thesaurus containing this kind of information (Allan, 2005). Milne, Witten and Nichols (2007) first select a set of Wikipedia articles that are relevant to the TREC document collection and build a corpus-specific thesaurus. They map the keywords of a query to this resource and use the extracted redirects, misspellings and synonyms in order to create an advanced expanded query. Using the TREC HARD data they report significantly better performance in terms of F-measure.

Li, Luk, Ho and Chung (2007) use Wikipedia as an external corpus: they first run the query and retrieve 100 relevant articles from the encyclopaedia, and extract their categories. The list of articles is re-ranked according to the most frequent categories. The 20 highest-ranked articles are analysed to extract 40 terms which are then added to the initial query. They report mixed results when evaluating on the TREC HARD data (Allan, 2005), with better results for weak queries, usually containing two or three words.

**Query segmentation** is another method which makes use of Wikipedia in order to improve retrieval accuracy. It aims to separate the keywords of a query into segments so

---

<sup>29</sup>High Accuracy Retrieval from Documents: <http://trec.nist.gov/data/hard.html>

that each segment maps to a semantic concept or entity. Correctly splitting the phrase into chunks is crucial for search engines, e.g., [*new york times subscription*] should be segmented as [*“new york times” subscription*], not as [*“new york” “times subscription”*]. Tan and Peng (2008) train a generative query model using n-gram statistics from the Web and the titles of Wikipedia articles. They use an expectation-maximization (EM) algorithm, to optimise the model parameters outperforming the traditional approach by 22%, from 0.530 to 0.646. Just by adding Wikipedia information they report an additional 24.3% increase over the standard approach which uses point-wise mutual information (MI) between pairs of query words: if the MI value between two adjacent query keywords is below a pre- defined threshold, a segment boundary is inserted at that position.

The multilingual nature of Wikipedia has been exploited for **cross-lingual information retrieval** (CL-IR). The inter-wiki links can be used to create multilingual entity dictionaries that can be used to translate queries from the source language to the target language. Additionally, the entries can be expanded using the redirect links in each language version of Wikipedia. This method was used in cross-lingual QA (Dornescu, Puşcaşu and Orăsan, 2008).

Potthast, Stein and Anderka (2008) extend the ESA similarity metric and propose a new retrieval model called Cross-Language Explicit Semantic Analysis (CL-ESA). The main idea is that most of the concepts, i.e., articles, used to represent documents in ESA have mappings to other language versions of Wikipedia, making these concepts largely language independent. They hypothesise that the relevance of a document (written in the target language) to a query (source language) can be calculated without translating neither the document nor the keywords of the query. To test this hypothesis, an English document was used as query in EN Wikipedia, and the German translation of the document as a query in the DE Wikipedia. The two results sets had an average correlation of 72%. The authors point out that the dimensionality of the concept space, i.e., the numbers of aligned articles in Wikipedia, is very important and that the method performs better on the large Wikipedias (EN,DE,FR,PT) because the concept space overlaps better.

### 2.4.3 Wikipedia and information extraction

The goal of **information extraction** (IE) is to automatically extract structured information from text documents. Typical subtasks of IE are:

- **named entity recognition** (NER) — recognition of spans of text that represent entity names (people, organisations, places), temporal expressions (times, dates), numerical expressions (physical measure, currency), and so on;
- **coreference and anaphora resolution** — identification of chains of noun phrases and pronouns that refer to the same entity;
- **relation extraction** — identification of relations between entities, such as *bornIn*(Person,Country) or *worksAt*(Person,Organisation).

#### Named Entity Recognition

Named Entity Recognition (NER) is concerned with identifying spans of text that denote entity names and classifying them using a type taxonomy. The most common classification is that employed in the MUC conferences<sup>30</sup>: LOCATION, PERSON, ORGANISATION, DATE, OTHER. According to the Wikipedia guidelines, in each page at least the first mention of an entity should be linked to its corresponding article (disambiguated references). Also, the category system can be exploited in order to have a fine-grained classification (entity types). These properties make Wikipedia a perfect corpus candidate for developing new models.

Toral and Muñoz (2006) note the possibility of using Wikipedia in NER and propose a method to automatically extract a gazetteer for the MUC categories, using just a part-of-speech tagger and WordNet. They suggest that the method is easily applicable to other languages by using EuroWordNet. They did not demonstrate the usefulness of the extracted gazetteers in actual NER systems and they used the general MUC classification scheme.

---

<sup>30</sup> *Message Understanding Conference* [http://www-nlpir.nist.gov/related\\_projects/muc/](http://www-nlpir.nist.gov/related_projects/muc/)

Kazama and Torisawa (2007) propose a simple NER method which exploits the encyclopaedia: given an input sentence, they extract all its n-grams and identify which ones are entries in the thesaurus extracted from Wikipedia (titles of articles). The class of the identified name is extracted from the first sentence of the article (the definition) using lexical rules. These type labels are used as an additional feature by a conditional random fields (CRF) tagger, along with standard features such as surface form and part of speech tag. They report an increase in NER performance using the additional information from Wikipedia.

Watanabe, Asahara and Matsumoto (2007) go beyond the definitions in Wikipedia articles: they extract a gazetteer of entities by mining lists and tables from the articles, based on the hypothesis that entities that are collocated in such semi-structured elements share similar semantic types. They train a graph CRF model based on context features extracted from the HTML markup, the definition and the categories of Wikipedia articles. A random set of 2300 articles from the Japanese version of Wikipedia was manually annotated, and the identified 14285 distinct named entities were tagged using a set of 13 categories selected from the 200+ types present in the full hierarchy of Sekine, Sudo and Nobata (2002). They report an F1 score of 78% for all the entities and their method obtained an F1 score of 54% for the 3898 entities that had no associated Wikipedia article.

## Entity Disambiguation

**Entity Disambiguation** is a refined version of NER in which names need to be linked to the entities they actually denote, a form of cross-document coreference resolution also known as entity linking (EL). It is particularly relevant in information retrieval and question answering since it enables systems to correlate information about one entity when examining documents from heterogeneous sources.

This task is particularly important since the same name can denote several distinct entities, e.g., *Paris* — capital of France, several cities in USA, a mythological figure, a celebrity and so on. Also, one entity can be classified simultaneously with different

categorisation schemes in Wikipedia, e.g., *Nicolas Sarkozy* is a person, a University of Paris alumni, a lawyer, a mayor and a president. Such granularity is more useful than the coarse-grained classes usually employed in NER.

Bunescu and Pasca (2006) created a thesaurus using article titles and known redirects. When several entries match, they use a cosine similarity metric between the query and the document or between the query and the document's categories in order to select the best match. From all the wiki links to entity articles they extract tuples  $\langle \text{anchor text}, \text{entity}, \text{context} \rangle$  (Bunescu, 2007). Entity disambiguation is formulated as a ranking problem: given an ambiguous name  $\langle \text{anchor text}_i, ?, \text{context}_i \rangle$ , rank the possible entities and select the one that is most similar to training observations. In order to measure the similarity,  $\langle \text{context-word}, \text{category} \rangle$  correlations are used in training a ranking Support Vector Machines (SVM) kernel on the dataset of disambiguated name occurrences. Their experiments show that the use of word-category correlations yields substantially better accuracy than the context-article cosine-similarity baseline.

Cucerzan (2007) created a more complete dictionary by exploiting not only the anchor text of the wiki links but also list pages and the category graph. He generates  $\langle \text{entity}, \text{tag} \rangle$  pairs from the links within list pages (e.g.,  $\langle \text{R.E.M. (band)}, \text{LIST band name etymologies} \rangle$ ) and from the category tags (e.g.,  $\langle \text{Texas (band)}, \text{CAT Scottish musical groups} \rangle$ ). Additionally, a much larger set of context pairs  $\langle \text{entity}, \text{context entities} \rangle$  is also extracted. Using this rich set of features, he performs entity disambiguation on names identified with a simple system based on capitalisation rules. He reports accuracy around 90%, and the method is independent of the language version of Wikipedia used.

The Text Analysis Conference (TAC) is a series of evaluations and workshops organised to foster NLP research by providing a common evaluation platform enabling organisations to share their results. Since 2009, one of the tasks of the yearly challenge has been Entity Linking: given a name (of a Person, Organisation, or Geopolitical Entity) and a document containing that name, systems must determine automatically if the reference knowledge base contains a node corresponding to the named entity, and which node is the best

match. The task is focused on semi-structured data: a set of Wikipedia infoboxes are used as the reference knowledge base and systems are encouraged not to use additional text from the Wikipedia articles (Ji et al., 2010). The results show that usually systems perform best for persons and worst for geo-political entities. Using semantic features extracted from the structure of the encyclopaedia usually improves performance. Some systems submitted entity linking runs both with and without using Wikipedia text. Employing the textual content usually yields performance improvement that is statistically significant at 89% confidence level.

### Coreference Resolution

The purpose of this task is to determine whether two spans of text (called referential entities) denote the same entity from the real world, without disambiguating which entity. It is related to anaphora resolution which consists of determining when a word or phrase, the **anaphor**, refers to a previously mentioned entity, the **antecedent** (see Mitkov, 2002).

Ponzetto and Strube (2006) use features such as similarity metrics between the first sentences of the articles describing the two potential coreferent expressions, the number of wiki links between them and the semantic distance between the articles using the category graph. Compared to a baseline method, they report a decrease of precision, but an increase in recall and F-measure, on the MUC data. On the same data, Yang and Su (2007) also report an increase in performance, using a similar baseline. They do not exploit the Wikipedia structure, but rather extract co-occurrence patterns in Wikipedia in order to assess the relatedness.

Mihalcea and Csomai (2007) employ Wikipedia as a resource for automatic keyword extraction and word sense disambiguation to *wikify* (hyper)text by automatically extracting the most important words and phrases in the document, and identifying for each such keyword the appropriate link to a Wikipedia article. The best ranking method for keyword extraction was to sort candidates by the prior probability of the

n-gram to be a link. For disambiguation, they use a data-driven method that integrates both local and topical features into a machine learning classifier, which outperforms the most-frequent-sense baseline.

Milne and Witten (2008*b*, 2009) propose a three stage approach to detect and disambiguate Wikipedia topics in documents. In the first step, candidate anchors are selected based on their probability to be a link. In the second step, a classifier predicts for every anchor the probability of each possible target topic to be the correct ‘sense’. The classifier is trained on several hundred of random articles, using features such as the topic commonness (the most frequent ‘sense’ is preferred) and the relatedness to the other ambiguous topics, measured by a link-based relatedness measure (Milne and Witten, 2008*a*). In the third stage, for every disambiguated topic another classifier predicts how central it is to the entire document. Topics which are very related to each other are sought, and specific topics are preferred to more generic ones. The list of noteworthy topics is ordered based on the predicted probability.

Commercial services exist which offer a similar functionality, e.g., AlchemyAPI<sup>31</sup> or Zemanta<sup>32</sup> and OpenCalais<sup>33</sup>, which also return the topics most relevant to a text document. Wikipedia has become the de-facto reference repository of real-world entities.

## Relation Extraction

The encyclopaedic nature and the uniform style of the articles in Wikipedia is appealing for methods that extract relations between entities from plain text. Additionally, many methods exploit the semi-structured text (infoboxes, templates, tables) and the links (category assignment) in order to extract high accuracy relations and map them to ontologies (see Section 2.4.4).

Extracting relations from plain text usually means starting with a set of known relations (called *seeds*), such as *capitalOf(Lisbon,Portugal)*. By extracting snippets of texts where

---

<sup>31</sup><http://www.alchemyapi.com/>

<sup>32</sup><http://www.zemanta.com/>

<sup>33</sup><http://www.opencalais.com/>

the known entities occur, examples of positive and negative contexts are automatically identified and used to construct extraction patterns, e.g., *?x ( \* ) is the capital and largest city of ?y*. Using these patterns, new text fragments are identified, and new  $\langle \text{entity}, \text{relation}, \text{entity} \rangle$  triples are extracted (Wu and Weld, 2007).

Wikipedia is well suited for such a task thanks to the wiki links: there are over 60 million links in the article graph (WAG), and considerably more if we consider it a multi-graph. In addition, since each article describes only one concept, all the entities mentioned in its content are candidates for semantic relations, e.g., as the object in  $\langle \text{subject}, \text{relation}, \text{object} \rangle$  triples. For example, in the article describing the *Douro* river all the links to articles describing cities are candidates for the relation *flowsThrough(Douro,?x)*. Thus the link structure of Wikipedia and the category system can provide valuable additional information.

Ruiz-Casado, Alfonseca and Castells (2007) use nouns from WordNet to identify new relations in the Simple English version of Wikipedia<sup>34</sup>. They report accuracy of 61%-69%, depending on the relation type. While this method is aimed at enriching WordNet (discovering new relations between WordNet concepts), Ruiz-Casado, Alfonseca and Castells (2006) apply a similar method to more general entity types. They extract pages relevant to five topics and then apply the method to extract a set of eight relations: *birth-year*, *birth-place*, *death-year*, *actor-film*, *writer-book*, *country-capital*, *country-chiefOfState*, *player-club*. They obtain high accuracy on the subset that corresponds to the topic of each relation, e.g., 93% for *player-club* in the football subset, but the accuracy is greatly decreased when the patterns are applied to the rest of the articles, e.g., 8% for *player-club*. This suggests that the same surface pattern can express different relations depending on the domain, and that more semantic constraints are needed.

Wang, Zhang, Wang and Yu (2007) extract seed examples from the infoboxes. However, they do not extract instances of the relations whenever the extraction patterns match.

---

<sup>34</sup><http://simple.wikipedia.org/>

Instead, they use *selectional constraints* in order to increase the precision of regular expressions without reducing coverage. For example, for the *directedBy(film,director)* relation they obtain patterns such as: *?x (is/was) (a/an) \* (film/movie) directed by ?y*. The selectional constraint in this case means that *?x* has to be a film and *?y* a person, in order to extract a valid instance of the relation when the pattern matches. The authors report precision and accuracy values above 90%.

Wang, Yu and Zhu (2007) propose a different approach. They no longer use lexical patterns, but rather extract features from the articles corresponding to the candidate entities. These features include the head nouns from their definition and categories titles, infobox attributes and terms that appear in sentences in which the two entities co-occur. They use a special machine learning algorithm designed for situations when only positive instances are available for training. Accuracy depends on the relation and the amount of training data, ranging from 80% for the *hasArtist* relation to 50% for the *isMemberOf* relation.

Wu and Weld (2007) propose a bootstrapping system that can automatically discover missing links and enhance the taxonomic data and the infoboxes' structure. They describe a prototype of a semi-supervised machine learning system which accomplishes these tasks on a subset of Wikipedia. Similar to previous approaches, they start with the data present in the infoboxes in order to create training data. Their system, Kylin, trains two classifiers. A document classifier selects the most appropriate infobox for a given article by using heuristics that try to find a category very similar to the infobox title. Then a sentence classifier is trained using the data extracted from infoboxes. This classifier tags each sentence with the relation it is likely to contain. For each relation a CRF model is trained in order to extract the triple from the sentence.

Wu and Weld (2007) go beyond simple relation extraction. They use a discriminant rule model to recover links that are not marked in the wikitext. While the evaluation performed on a subset of Wikipedia articles shows good results, it is possible that the heuristics they use do not perform as well on the entire Wikipedia. Future improvements

they propose include the enhancing of infoboxes by adding new fields from similar infoboxes, inheriting fields from parent infoboxes using WordNet hypernymy relations and replacing infoboxes that are either rarely used or just small variations of more popular ones. The system is also sensitive to the amount and bias of the training data. In order to be able to extract information for completing less popular infoboxes, Wu, Hoffmann and Weld (2008) extend the Kylin system by exploiting information from the Web. The new system, K2, yields a substantial improvement in recall without compromising precision.

Most IE systems are developed to identify specific relations based on the available training data. Their coverage is limited and performance drops if applied to a different corpus. TextRunner (Banko et al., 2007) pioneered an alternative approach: open information extraction, where an unbounded number of relations are extracted from text in a domain-independent fashion by automatically extracting the relation name as well as its two arguments. Wu and Weld (2010) propose an open-IE approach based on Wikipedia which also uses the infobox attributes to identify seed samples of relations in Wikipedia articles. However, instead of learning specialised, lexicalised patterns, it builds relation independent training data to learn an unlexicalised extractor. The best performance was achieved using the path between the two arguments in the dependency tree.

These advancements open new avenues for semi-structured information access: retrieving data using triples is no longer limited to a small set of relations with a constrained expressive power, but open to a potentially unlimited set of relations, while still allowing more control than keyword-based IR. Furthermore, using wikification these triples can be linked to Wikipedia topics, enabling a new generation of semantic search engines which have IE at their core rather than IR. The following section presents extraction mechanisms designed to exploit mainly the semi-structured information from Wikipedia and make it available in a machine readable format.

#### **2.4.4 Wikipedia and knowledge representation**

Ontologies are a means of formally representing knowledge. In an ontology, concepts are linked by relations and are described by attributes. Concepts usually denote **classes** of

objects (such as *person*, *country*, *band*) while entities are **instances** of these classes, e.g., *Nicolas Sarkozy*, *France*, *The Beatles*. An ontology has three components:

1. the *schema vocabulary*: the classes and the relations amongst them, also known as the terminology – T-BOX
2. the *data*: the instances of concepts linked using the schema vocabulary, also known as the assertions – A-BOX
3. the *ontology language*: specifying, for example, whether the *is-A* relation is transitive or not

The expressivity of the ontology language is what makes automatic inference possible. Its complexity ranges from thesaurus level to description logics (e.g., OWL Lite, OWL DL), full first-order logics (e.g., OWL Full, McGuinness and van Harmelen, 2004) or higher logics (e.g., the CycL language, Lenat and Guha, 1989, 1991).

**DBpedia** (Auer et al., 2008) is a project aimed at extracting information from Wikipedia and making the data available on the Internet as RDF triples. It uses the category hierarchy from Wikipedia in order to create classes of concepts. Attributes and relations are extracted from the infoboxes, while the articles are instances of concepts. The RDF data is exposed to the Semantic Web and is available for consumption by computers.<sup>35</sup>

The main source of information for the DBpedia dataset consists of the infobox templates used in the Wikipedia articles. As shown in Section 2.3, infoboxes are a particular case of templates which are used to display information that is stored as attribute-value pairs (see Figure 2.2b). For example, from the article describing *England*, the template *infobox country* can be used to extract the relation *hasCapital(England, London)* and the attribute *populationTotal(England, 49,138,831)*. Extracting structured data from articles is not a trivial task. Wikipedia, being a collaborative effort, is sometimes inconsistent and ambiguous, and contains contradictory data, errors and even spam. In many cases the

---

<sup>35</sup>see <http://dbpedia.org> for details

same meaning is represented inconsistently, e.g., three different relations with the same meaning: *dateOfBirth*, *birthDate* or *birth\_date*. These problems are generated by the fact that Wikipedia is intended to be read by humans, who can easily view and understand the generated HTML document using a Web browser.

Initially, DBpedia was only concerned with making the infobox content available as RDF data, and did not even attempt to create a taxonomy of the concepts. Apart from the infobox relations directly extracted from the infoboxes, only the category links were extracted and labelled with the relation *isRelatedTo*. The RDF data was only meant to be machine readable, and it was not intended for automatic inference (i.e., not a true ontology). Version 3.1 of DBpedia (released in August 2008) included a core OWL ontology comprising 170 classes and 900 class properties. This ontology is very flat: the path to *Andre Agassi* is *Resource*→*Person*→*Athlete*→*TennisPlayer*, while for *Albert Einstein* it is *Resource*→*Person*→*Scientist*. This enables many different names that exist for the same relation to be unified (mapped to the same ontology property, in this case *Person#birthdate*). The latest version of DBpedia has more than 1 billion of RDF triples, a quarter of which were extracted from the English language version of Wikipedia alone.

One of the shortcomings of the Wikipedia category graph is that the category types are not explicit. Links to **list categories** (as mentioned in Section 2.3) correspond to the hypernym ↔ hyponym relations of type *is-A*. This relation is very important in QA (and other NLP applications as well) because it encodes the types of each entity. **Topic categories** can introduce erroneous *is-A* relations, with a negative impact on the overall performance of the system.

The ambiguity of *is-A* category links is tackled by Yago — Yet Another Great Ontology (Suchanek, Kasneci and Weikum, 2007*a,b*). This is an ontology that maps Wikipedia leaf categories to WordNet synsets. For example *Category:People from Paris* is a *subClass* of *wordnet\_person\_100007846*. By assigning the WordNet synsets to Wikipedia articles based on their categories, a large concept taxonomy is produced. On top of it, several heuristics are employed for extracting further relations to add to the knowledge base.

For example, person names are analysed in order to differentiate between given and family names, adding relations like *familyNameOf*(*Albert Einstein*, “*Einstein*”), while other heuristics extract relations from the category names, for example *bornInYear*, *establishedIn* and *locatedIn*. Yago provides formal inference capabilities, using its own language. The data is available online as RDF triples.<sup>36</sup> A manual verification showed that average accuracy for all relations was above 90% (Suchanek et al., 2007a).

Both DBpedia and Yago are part of the Linking Open Data (LOD) project<sup>37</sup>. Thus, they are interlinked and entities have dereferentiable URI identifiers, i.e., the data for each resource is available online and can be accessed using a Semantic Web browser. The greatest achievement of the LOD project is the interlinking of DBpedia with other data sources: the LOD Cloud<sup>38</sup> (see Figure 2.3). The most relevant datasets for this research are<sup>39</sup>:

- **Geonames, Eurostat and World Factbook:** three important datasets providing up-to-date geographical information and statistics about places and countries
- **OpenCyc and WordNet:** two important semantic networks, which provide accurate *is-A* links
- **Linked MDB, Music-brainz and RDF Book Mashup:** datasets providing information about films, musical artists and books, respectively.

Another project that extracts relations from Wikipedia is **WikiTaxonomy** (Zirn, Nastase and Strube, 2008; Ponzetto and Strube, 2007a). Explicit relations are extracted by splitting the category title. For example, given an article *?x* from *Category:Movies directed by Woody Allen* they extract triples like *directedBy*(*?x*, “*Woody Allen*”) and *isA*(*?x*, “*movie*”). More importantly they use heuristics to extract implicit relations from categories like *Category:Albums by The Beatles*. Articles directly linked to this category are marked

<sup>36</sup><http://www.mpi-inf.mpg.de/~suchanek/downloads/yago/home.htm>

<sup>37</sup><http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData>

<sup>38</sup><http://www4.wiwiwiss.fu-berlin.de/lodcloud/state/>

<sup>39</sup>a comprehensive list is found at <http://esw.w3.org/topic/TaskForces/CommunityProjects/LinkingOpenData/DataSets>

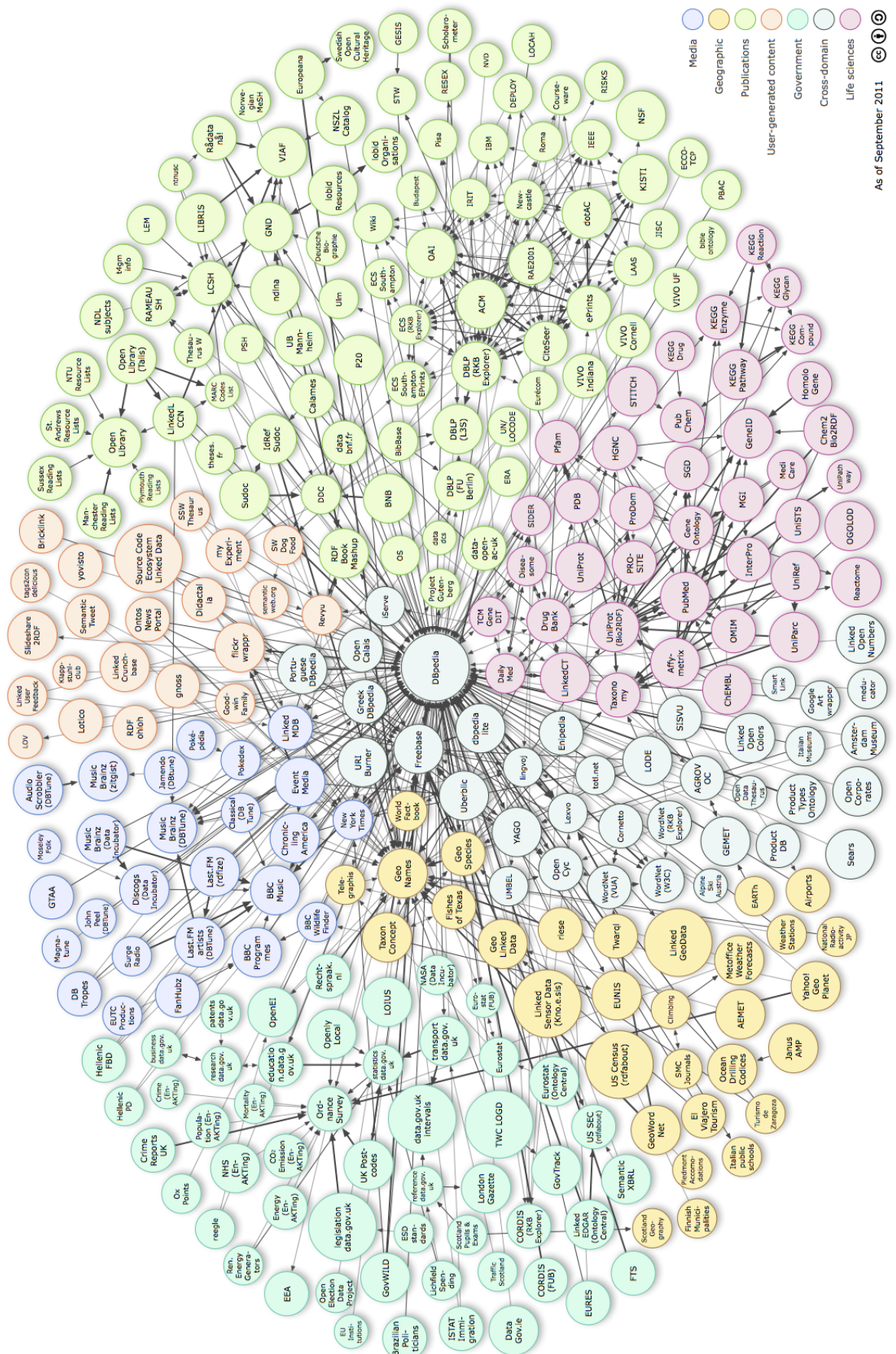


Figure 2.3: Linked Open Data cloud diagram, by Richard Cyganiak and Anja Jentzsch.  
<http://lod-cloud.net/>

with the relation *is-A*(?a, “album”) but also *artist*(?a, “The Beatles”), and for articles that represent songs on the album, the implicit relation *artist*(?song, “The Beatles”). They extract a total of 3.4 million *is-A* and 3.2 million spatial relations (e.g., *locatedIn*), along with 43,000 *memberOf* relations and 44,000 other relations such as *causedBy* and *writtenBy*. These relations are not part of a formal ontology (just assertions, no inference capabilities), but they can be merged with other datasets (e.g., the DBpedia dataset) or mapped to an ontology.

**Freebase**<sup>40</sup> is to structured knowledge data what Wikipedia is to encyclopaedic knowledge: a collaborative effort in which users contribute facts and create an ever-growing knowledge base. As is the case with other resources, the focus is not fostering automatic reasoning, but rather making data machine-accessible. Following Semantic Web vision, applications that automatically aggregate data from several sources can be easily developed, creating informative new views, i.e., mashups. Most of the initial facts were extracted from Wikipedia. The mapping links between Freebase and DBpedia resources are part of the LOD cloud.

### 2.4.5 Wikipedia and question answering

Wikipedia articles span numerous domains (coverage) and offer a great deal of detail (factual information), which makes them a promising source of answers. The fact that each article starts with a brief definition is exploited by Web search engines like Google (the *define:* operator) and Ask.com (queries starting with *What is ...* or *Who is ...*), which return the definition from the relevant Wikipedia article. The encyclopaedia has also been used by QA systems in academic competitions, at first as an additional resource, and more recently as the target document collection.

In the TREC QA Track<sup>41</sup>, Wikipedia was used for the first time in the 2004 competition. Lita, Hunt and Nyberg (2004) only used the entities from Wikipedia as an additional

---

<sup>40</sup><http://freebase.com>

<sup>41</sup><http://trec.nist.gov/data/qa.html>

gazetteer and found that it had a *possible* answer coverage higher than traditional resources such as WordNet, Geonames and CIA World Fact Book. Ahn et al. (2005) used Wikipedia, amongst other sources, to locate relevant articles and extract candidate answers.

The QA@CLEF Track<sup>42</sup> is another popular forum for evaluating QA systems. Monolingual and cross-lingual QA are addressed by providing corpora and tasks in many different languages. Along with traditional newswire collections, a Wikipedia HTML snapshot (from November 2006) was used as a target collection. Participants generally transformed it to plain text and ignored the structure of the encyclopaedia, therefore their systems are of little interest from the point of view of the method proposed in this research. However, Wikipedia articles received special attention mainly for definition questions like *What is a Top Quark?* or *Who is the president of France?*. Additionally, the inter-wiki links were examined in order to better address translation issues in the cross-lingual tasks (see Dornescu, Puşcaşu and Orăsan, 2008).

WiQA<sup>43</sup> was another CLEF Track specifically designed for Wikipedia. It was aimed at assisting Wikipedia contributors when editing an article, by suggesting snippets from related articles that contain relevant information which should be added to the article (Jijkoun and de Rijke, 2007). Participants mainly used a bag-of-words similarity measure to identify related articles and then employed a ranking method to select paragraphs that contained novel information.

In 2007, the Initiative for Evaluation of XML Retrieval (INEX)<sup>44</sup> organised an Entity Ranking Track in which systems had to return a list of relevant Wikipedia entities in response to queries (de Vries et al., 2008). Vercoustre et al. (2007) used an adapted PageRank algorithm and combined Wikipedia features with a standard search engine, doubling the performance of the search engine alone. The track was organized again in

---

<sup>42</sup><http://www.clef-initiative.eu/track/qaclef>

<sup>43</sup><http://ilps.science.uva.nl/WiQA/>

<sup>44</sup><http://www.inex.otago.ac.nz/>

2008 along with a new Question Answering Track which also used Wikipedia as target document collection. It is important to note the *entity–article* duality in these tasks.

**GikiP** (Santos et al., 2008), a pilot task that exploited Wikipedia structure, was organized at GeoCLEF in 2008 (see Section 4.2). Its purpose was to foster the development of systems able to return a list of entities from Wikipedia that correspond to a topic that has some sort of geographic constraint. The task was multilingual (English, German and Portuguese) and the topics were translated into all participating languages, to facilitate system development.

**GikiCLEF** (Santos and Cabral, 2009<sup>a,b</sup>) was the successor of GikiP organised at CLEF in 2009. The focus on geographically motivated questions remained, but the number of languages was increased to ten, and the total number of topics was also increased. It attracted more participations, covering open-domain QA systems, interactive information retrieval and novel semantic QA approaches.

This thesis proposes a novel QA approach which was tested in GikiP and GikiCLEF. The generic architecture is described in Chapter 3. Details of the system components and the system’s results are presented in Chapter 4.

## 2.4.6 Conclusions

This chapter gave an overview of previous works to contextualise the research pursued in this thesis along two aspects: first, introducing question answering and the main types of QA systems; and second, illustrating the potential of Wikipedia in various NLP fields relevant to QA research.

The standard architecture of open-domain QA systems was presented in Section 2.2.2, with an emphasis on textual QA – systems which rely on an information retrieval engine to identify relevant text snippets based on their words rather than on their meaning. The next chapter will further analyse various limitations which affect this type of system

and which are inherent in their design, arguing that to overcome these limitations new semantic approaches which exploit world knowledge are needed.

Wikipedia, a large repository of human knowledge, is a pivotal resource with tremendous potential for NLP and QA research. Section 2.3 described how Wikipedia is structured and defined the terminology used throughout the thesis. Section 2.4 reviewed several NLP studies in the fields of information retrieval, information extraction, knowledge representation and semantic relatedness, which exploit the semi-structured content of the encyclopaedia. The majority of these studies create tools or resources which can be directly employed by the different NLP components of a QA system.

Using Wikipedia as a resource opens up the possibility for QA systems to use world knowledge and to perform simple forms of common-sense reasoning which could allow more difficult questions to be answered. Chapter 3 proposes a novel semantic QA approach which builds upon these ideas. Chapter 4 describes a QA system that implements the approach and directly exploits Wikipedia at a semantic level.



## Chapter 3

# Novel Approach for Encyclopaedic Question Answering

### 3.1 Overview

This chapter proposes a paradigm shift in open-domain question answering from the *textual approach*, built around a keyword-based document retrieval engine, to a *semantic approach*, based on concepts and relations. Section 3.3 presents some of the problems affecting state-of-the-art textual QA technologies. Section 3.4 advocates that a paradigm shift is necessary to address these problems and to enable more complex question types. Section 3.5 proposes a novel architecture for semantic QA systems which exploit world knowledge extracted from encyclopaedias at a semantic level. The key issues regarding this innovative approach and its challenges are then outlined.

### 3.2 Textual QA

The Internet has started a “digital revolution”, changing the way people communicate, the way they publish and access documents and the way they interact with each-other. As the amounts of information available started to grow exponentially, the need to find documents on the Web gave birth to search engines such as Yahoo! and Google which allow users to search for web pages based on the words they contain.

One of the limitations of such Information Retrieval (IR) engines is that they are designed to retrieve documents. When a user is not searching for a document but for a very specific piece of information, he has to retrieve relevant documents using the IR engine and then examine them to manually locate the specific information sought. Sometimes the user has to repeat this tedious process, adjusting his query until he finds, in one of the retrieved documents, a paragraph containing the exact piece of information he was looking for. For some types of factual information needs, this process can be very repetitive and could lend itself to automation. Towards the end of the 20<sup>th</sup> century, open-domain QA was touted as the key to address this problem: users would not be limited to searching documents using complex keyword-based queries, but instead leverage NLP technologies and ask natural language questions: the computers would search for relevant documents and extract the answer on behalf of the user.

This proved to be an elusive vision. Answering questions, in its general form, is an AI-hard problem: computers need to possess extensive world knowledge as well as a number of human cognitive/intellectual skills. In general, natural language questions can cover any topic. Most information processing tasks can be phrased as natural language questions: summarisation, information extraction, reading comprehension, politics, maths, physics, philosophy and so on. Questions are also employed in tests meant to assess human abilities and skills, e.g., essay, multiple choice or short answer questions.

Because questions can be very complicated, QA research has focused on types of questions which deal with less complex information needs. As shown in section 2.2, *closed-domain QA* limits the necessary amounts of world knowledge to what is explicitly modelled when building the domain specific system, *open-domain QA* limits the types of questions addressable to those which can be answered by a single textual snippet, such as definition, factoid or list (enumeration) queries, while *canned-QA systems* are limited to huge collections of Q&A pairs, focusing on retrieving the ones that best match what is asked.

In open-domain QA, evaluation fora such as TREC<sup>1</sup> and QA@CLEF<sup>2</sup> have been instrumental in shaping the field: they helped create a community and provided common means to objectively compare systems' performance. Having an IR origin, they have focused on questions which can be answered by a single textual snippet, such as definitions or factoid questions, envisioning a gradual increase in question complexity (Burger et al., 2000). While there are strong arguments supporting the use of a controlled question taxonomy, the limitation to single-snippet answers imposed a significant constraint on the amount of information systems need to take into consideration: the underlying hypothesis is that the context of a snippet does not affect the answer. As shown in section 2.2.2, textual QA approaches are essentially 'smart' paragraph retrieval systems, enhanced with extraction and ranking. They focus not on understanding text, but on identifying the snippet most likely to answer the question. Aggregating information from multiple documents usually means re-ranking candidates based on their frequency.

Despite the interest generated and the amounts of research dedicated to this topic, the open-domain QA vision has not yet materialised and it has made little impact on the way people search information on the Web. The next section will outline some of the factors that contributed to the relatively limited success of QA.

### 3.3 Criticism of textual QA

There are many obstacles that prevented textual QA from becoming a viable alternative to Web search engines. This section examines some of the key factors which have a negative impact on the performance of state-of-the-art textual QA systems, as well as factors which hinder user adoption of QA technologies. Section 3.3.1 presents common sources of natural language ambiguity which affect NLP tools in general, and textual QA

---

<sup>1</sup>Question Answering Track at the Text REtrieval Conference <http://trec.nist.gov/data/qamain.html>

<sup>2</sup>Multi Lingual Question Answering Track <http://nlp.uned.es/clef-qa/>

in particular. Section 3.3.2 then argues that textual QA systems are difficult to extend to more complex types of questions, because they are too focused on factoid questions. Section 3.3.3 describes the role of QA evaluation campaigns, which focused almost exclusively on batch performance, thus favouring the development of non-interactive systems, rather than aiming to enhance the “perceived performance” experienced by human users.

### 3.3.1 Natural language ambiguity

What makes textual QA a difficult task is the complexity of processing highly ambiguous natural language. Ambiguity affects both humans and machines, but to different extents. Humans use their knowledge of the world to disambiguate amongst possible interpretations of natural language and are much less likely to get confused by apparently simple input. While misunderstandings among humans are not only possible, but frequent, computers lack similar knowledge representation abilities and get confused much easier.

The reason why machines find it hard to process text is because of the way meaning is expressed in natural language. On the one hand, the problem is that *the same form can encode different meanings* and selecting the correct one usually depends on context. Failure to disambiguate the correct meaning leads to a decrease in precision, as irrelevant data is retrieved. On the other hand, the converse problem is that *the same meaning can be encoded in different forms*. This causes a decrease in recall when the system is unable to find all the relevant information. Ambiguity can arise at different levels of representation.

**Lexical ambiguity** is the simplest form of ambiguity. It is caused for example by homophones in speech (words with the same pronunciation but different meaning, e.g., *to, too, two*) or homographs in writing (same spelling, different meaning, e.g., *port*: a type of wine or a city by the sea). Polysemy is a particular case when homographs have related meanings (e.g., *bank* the building or the financial institution). These phenomena cause

imprecise retrieval. The converse effect (poor recall) is caused by synonyms (different words having the same meaning, e.g., *association football* and *soccer*), especially when words are partial synonyms (depending on context, e.g., a *long* or *extended* period of time vs. *long arm* and *extended arm*). The problems are similar when it comes to named entities: one entity can have different aliases (Paris, the French capital, Leucetia), while one name can denote/refer to different real world entities (*Paris* – capital, film, legendary figure, etc.). Other lexical-semantic relations such as hypernymy and hyponymy (the relations between general and specific terms, e.g., *colour* and *red*) have similar effects.

**Syntactic ambiguity** is another type of ambiguity. The most common type of problem is caused by prepositional phrase attachment (e.g., the classic example “*I saw the man on the hill with the telescope*”, Newell 1993). This is a case of surface-structural ambiguity: different syntactic parses convey different semantic interpretations of the text. Finding the correct parse usually requires a certain level of common-sense reasoning, combining world knowledge and contextual information. In the given example, replacing *man* with *star* or *telescope* with *forest* preserves the same part-of-speech sequence, but changes the more likely reading of the sentence. A state-of-the-art syntactic parser cannot distinguish between these examples unless they were present in the training data. In contrast, a system that works with concepts could determine, for example, that it is more likely to use a *telescope* to look at *stars* than at *people*, and that it makes little sense to use a *forest* as a seeing instrument. This type of ambiguity is particularly important for questions because constraints are often expressed as prepositional phrases.

**Semantic and pragmatic ambiguities** are more difficult to tackle. In the case of *deep-structural ambiguities*, two or more readings of a sentence are possible despite having the same parse tree, as in, e.g., *Mary likes horses more than John*. It is even more challenging to tackle *coreferential and anaphoric ambiguities*, such as: “*John kissed his wife, and so did George.*” – whose wife did George kiss? Other ambiguity sources are phenomena such as *metonymy* (when a concept is replaced with the name of something intimately associated with it, e.g., *Downing Street* instead of the *British Prime Minister’s Office*)

and *synecdoche* (when a specific part of something is used to refer to the whole, e.g., *I have three mouths to feed*). Ambiguity can reside also at the **pragmatic** level, when the correct interpretation depends on the context: “*Do you know the time in Paris?*” – where both “Yes” and “GMT+1” are valid answers, but not necessarily informative because the person asking the question probably wants to know the actual time.

These are but a few sources of ambiguity that affect the performance of NLP tools. Realistically, ambiguity cannot be completely eliminated as utterances can be themselves deliberately ambiguous. There is a further obstacle: inferences such as presupposition and entailment help humans recover additional information, bridging the gap between what is explicitly stated and what is implied. This ability cannot be reproduced by current automatic techniques.

### 3.3.2 Design issues

Textual QA systems cannot deal properly with the issues described in the previous section because they were not designed to do so: systems simply exploit the similarity between certain types of questions and corresponding answer bearing snippets by primarily using lexical resources (usually dictionaries, thesauri, or knowledge from WordNet). Although answering questions is a complex task, standard QA systems have focused almost exclusively on the surface, textual side of the problem. This skewed the focus of QA research towards devising ever more complex retrieval mechanisms and confidence estimators for answer extraction. In this section, it is argued that the generic architecture employed in standard textual QA systems is inherently limiting both the achievable level of performance as well as the extensibility of the system.

The way QA systems usually deal with ambiguity is by employing existent NLP tools (from part of speech taggers and parsers that address lexical and syntactic ambiguities, to anaphora resolvers, named entity recognisers and word sense disambiguators) which try to “augment” plain text by adding annotation layers on top of each other, in order to construct a richer, deeper semantic representation. NLP tasks can be solved more or

less successfully and there is a plethora of tools that were developed to address them individually. However, the fact that in free text usually multiple ambiguity sources manifest simultaneously makes it even more challenging for a QA system to correctly interpret and understand the meaning of natural language.

Besides the fact that most tools assume perfect input, which is not the case in a NLP processing pipeline, they usually have exactly one output: the most probable given the input. By delegating the responsibility of solving ambiguity to general purpose tools, QA systems are effectively ignoring ambiguity. Using a sequential pipeline of tools tends to accumulate errors rather than eliminate them.

In order to achieve a good performance in the standard evaluation campaigns, most of the current textual QA systems are designed to pinpoint a small text snippet – usually a paragraph or a sentence – from the document collection that contains the answer. QA systems are usually unable to reliably extract the answer if such a snippet does not exist and the information necessary to compile the answer is spread across several documents, or requires analysing structured data such as tables. As shown in Section 2.2, most systems employ multiple extraction strategies and rank the candidate answers based on aggregated confidence. This ranking approach is limited to simple factoid answers, because textual QA systems lack the semantic representations necessary to reliably combine different pieces of information.

Another shortcoming of current textual QA systems is that they usually remove document meta-data. Because the document collections that were initially used in open-domain QA consisted mainly of newswire articles, systems usually transform all the documents to plain text in order to perform standard keyword-based document retrieval. Valuable information is ignored when, for example, the hyper-links from Wikipedia articles or links from newswire corpora are removed.

In this research, it is argued that QA systems should make full use of all the data and meta-data present in documents. Of particular interest are hyperlinks since they point

to other documents that are related in some way, providing a “**hyper-context**” of the document in the collection. Ignoring the hyperlinks means that the documents are treated independently (they do not have a context, they just have words in common). QA should focus on entities, e.g., documents that talk about the same individuals, rather than on words, e.g., documents having similar words. Link analysis could help reveal connectivity patterns that reflect a semantic similarity between documents (Thelwall, 2004).

### 3.3.3 Perceived performance

The open-domain QA roadmap (Burger et al., 2000) envisioned addressing increasingly complex questions. However, state-of-the-art performance is acceptable only for simple factoid and definition questions. This causes systems to be perceived as unreliable and feeble, as users are likely to ask questions which the system is unable to answer correctly. An important issue is the amount of time required to identify the answer: the CLEF campaign allowed systems up to 5 days to answer 200 questions. While in practice this was more than sufficient, by comparison, Web search engines return results almost instantly, even while the query is being typed.

The TREC evaluation metrics are in line with the IR tradition: the number of correct results in the top 10 or 20 answers. The user still has to check each answer and accompanying snippet, the way they do when using a standard search engine. The combination of limited coverage and low accuracy make textual QA systems a poor alternative to Web search engines which are already familiar to all Web users.

Perhaps one of the biggest factors limiting QA is the fact that the performance achieved does not justify the user interface paradigm adopted: that of a single interaction communication paradigm: one request – one response. QA systems are designed as ‘black boxes’: users are presented with the final results. Mistakes occurring at any intermediate stage of the complex processing pipeline propagate and result in erroneous answers. The complex processing steps are difficult to debug, even by the developer: each answer has to be manually traced in order to determine the error sources. This is a tedious task, and

most published research tends to evaluate modules individually, isolated from the other components. This makes it difficult for the system to learn from past experience, as the evaluation is performed end-to-end and modules interact in complex ways.

Open-domain QA research had a substantial surge in interest during the last decade, with a very active community. Despite many variations and enhancements which have been proposed over the years, QA systems that rely mainly on processing text using NLP tools are still confined to the laboratory, far from becoming a true natural language alternative to search engines. The main cause is probably the specificity/narrow coverage of QA systems compared to the generality of IR systems. Research drifted towards complex IR, instead of trying to incorporate world knowledge. The term “open-domain” is, in this respect, a bad choice as it implies the ability to answer any question, in the way a human can. Unfortunately, this is not yet the case and users are likely to feel disappointed with the performance of available systems and quickly go back to their favourite Web-search engine. To address this problem, a new paradigm is proposed in the next section.

### **3.4 Proposed approach: paradigm shift**

The textual approach has prevailed because computers do not have common-sense or reasoning abilities resembling those of humans. As existing technology does not enable machines to ‘understand’ the contents of a textual document, standard textual QA systems rely instead on the similarity between the question and the answer-bearing snippet. While this approach is suitable for some types of questions, it cannot address questions whose answers are not contained in one text snippet. As new resources have become available, the time is now right to add more semantic processing to the QA task. Wikipedia has already proved transformational in several fields of NLP, as shown in Section 2.4. It amasses large amounts of encyclopaedic information which could be readily exploited as world knowledge, especially since extracting structured data is currently a very active and successful research area. Wikipedia is already a knowledge hub for connecting various data sources. The emergence of a web-of-data, the availability

of linked-open-datasets, and the amounts of information represented in ontologies, which afford a certain level of automated inference, prompt us to propose a new approach to QA: an architecture which is built around concepts rather than words to allow linking the vast amounts of textual information available to the formal representations enabling simple inference. This new approach would help QA systems find answers when the supporting information is spread across several documents.

The massive amounts of RDF data available are an essential resource for reinvigorating QA research according to the proposed approach. This thesis advocates shifting the focus from the widely accepted textual QA to an encyclopaedic factual QA in which finding answers means combining various pieces of supporting information. One way to achieve this is by linking free text to a structured knowledge base. To do so, systems need to identify and disambiguate entities and their types, properties and relations. The aim is not to create a formal, rigid, unambiguous ontology that empowers formal reasoning and inference, as is the case in closed-domain ontology-driven QA, but instead to combine using structured data and semantic links, with aggregating heterogeneous knowledge sources and accessing free text. Advances in information extraction, ontology mapping and probabilistic reasoning make it possible to leverage semantic resources to employ simple inference methods.

Building QA systems around concepts using Semantic Web technologies means that systems can take advantage of structured data-sources. For example, a system does not need to extract the birth date of an actor from news-wire articles using the textual approach, because this information is likely to be already available in a structured format in IMDB, Freebase or DBpedia and can be queried directly. The QA system can evolve from the current IR approach towards an encyclopaedic approach where systems use an IE framework to extract semantic data from text. Using semi-structured data allows a QA system to focus on integrating existent resources and deciding which data sources need to be queried for relevant information, regardless of their textual or structured nature. Answering questions is seen more as a higher-level integration problem rather than as a

complex retrieval ‘script’. Of course, a system cannot rely exclusively on existing external databases: it should use IE tools to actively extract new data to continuously populate and update its knowledge base and it should also expand its coverage by incorporating new semi-structured datasets. This also enables efficient data access mechanisms by limiting the amounts of information that need to be extracted from raw data at query time.

To achieve this in an open-domain setting, it is essential to exploit Wikipedia. Current systems usually use the encyclopaedia as source for lists of named entities or as an additional textual collection. To fully exploit its contents, new QA systems should use Wikipedia as a backbone for representing world knowledge, albeit in an underspecified semantic format. For example, there are various links from the page describing *Fernando Alonso* to pages such as *Formula 1*, *Renault*, *Ferrari* and *Red Bull Racing* but they only imply a possible relation. The QA system can exploit these links simply as proof that a semantic association between entities exists. Automatically distinguishing different types of relations can provide the type of information necessary to answer questions, e.g., distinguishing the Formula 1 teams the champion drove for from the ones he did not.

Web of Data resources such as DBpedia and the LOD Cloud previously mentioned in Section 2.4.4 allow machines to access the contents of Wikipedia at a semantic level, rather than an annotated lexical level. Instead of performing classic keyword retrieval, systems can browse the concept graph and search for patterns that help uncover semantically relevant information. In addition to lexical similarity based on a bag-of-words document representation, QA systems should also employ semantic relatedness measures such as explicit semantic analysis (ESA) or Wikipedia Miner (WM) described in Section 2.4.1. Besides, semantic similarity measures can also be developed to distinguish documents which are similar because they discuss the same entities from documents which describe similarly typed entities, e.g., to distinguish documents describing the Eyjafjallajökull volcano from documents describing the same type of effusive eruptions as the one which

occurred in 2010.<sup>3</sup> Another advantage of the concept-based approach is that systems can deal with eponymous entities when their mentions are disambiguated and linked correctly. This also enables systems to find relevant information regardless of the actual alias present in a particular document. Evolving from analysing plain text towards processing semantically enriched text requires a large repository of entities and the LOD Cloud is a strong candidate for this role.

One way to foster the development of such systems is to employ more difficult questions in the shared evaluation campaigns. Questions are more difficult to answer if QA systems have to gather and integrate various pieces of support information from distinct sources in order to extract a correct answer. Open list questions are such an example because the system does not know how many correct answers exist and at the same time it is unlikely that all of them are simply enumerated in some paragraph of the document collection. This means that, instead of searching for the most likely candidate answer, a QA system has to verify all the possible answers. Another type of questions which are also difficult to answer are the so called “complex questions” (Diekema et al., 2004; Harabagiu et al., 2001, 2006). They usually contain several modifying prepositional phrases and can be regarded as a composition of constraints regarding events, entities, relations and attributes. Typically, relevant information for these kinds of questions is spread across various documents and/or data sources. These pieces of information need to be fused together to produce a final answer.

Complexity should be focused on integrating information rather than on performing human-like inference such as deduction, entailment or presupposition. To do so, systems need to manipulate concepts rather than just words, tokens and chunks of text. Newer campaigns, such as the GikiCLEF task<sup>4</sup> or the INEX Entity Ranking task<sup>5</sup>, focus on list questions, while the 1st Workshop on Question Answering over Linked Data<sup>6</sup> (QALD-1) Open Challenge addresses the problem of translating natural language questions

---

<sup>3</sup>[http://en.wikipedia.org/wiki/2010\\_eruptions\\_of\\_Eyjafjallajökull](http://en.wikipedia.org/wiki/2010_eruptions_of_Eyjafjallajökull)

<sup>4</sup><http://www.linguateca.pt/GikiCLEF/>

<sup>5</sup><http://www.inex.otago.ac.nz/tracks/entity-ranking/entity-ranking.asp>

<sup>6</sup><http://www.sc.cit-ec.uni-bielefeld.de/qald-1>

into a form that can be evaluated using standard Semantic Web query processing and inferencing techniques.

The distinction between response and answer needs to be emphasised. Besides results, a response should suggest ways to refine or clarify the question. In case the results are inaccurate, a human user can provide feedback to the system e.g., by selection from a list of suggestions or by modifying the question. A subsequent response can thus have more relevant answers and the system can collect data to improve its performance. End-to-end performance may be a good way to compare systems, but it might not reflect whether or not people find a certain system useful as an information access tool. One way to involve human users in evaluation campaigns is to consider limited time sessions. The number of correct answers that a user finds assisted by the system should correlate with real-world performance. This would foster research and development of QA systems which help actual users find information, complementing other search tools.

### 3.5 Semantic architecture for open-domain QA

The previous section argues that a paradigm shift is necessary to develop new QA systems capable of addressing more complex questions. This thesis proposes a novel approach that can be seen as a hybrid between textual QA, based on IR, and semantic QA, based on IE and Semantic Web technologies. The architecture described in this section combines both structured and unstructured data to analyse questions, identify answers and generate responses. The paradigmatic shift consists in that the architecture does not have words at its core, but concepts, i.e., classes, instances, relations and properties. This allows the aggregation of information from across data sources, combining textual information with structured and semi-structured data in a way that enables primitive forms of inference in an open domain.

In this new approach, answering questions is a process involving two main phases: a) the **analysis** phase, which is responsible for understanding the question and identifying

answers, and b) the **feedback** phase, responsible for interacting with the user. The purpose of the latter is two-fold. Firstly, it should provide a more efficient interface between the human and the machine, allowing, for example, disambiguation, justification and presentation of additional information using interactive interfaces, enabling QA as a useful tool for information search. Secondly, use data could be collected for user modelling and extracting feedback information from the logs. This will allow the system to ‘remember’ both errors and successes. Developing such interfaces pertains to the domain of human-computer interaction, and is not the focus of this thesis.

The data model adopted in the semantic QA architecture is closely related to conceptual modeling approaches such as entity-relationship models, entity-attribute-value models and RDF<sup>7</sup> models. Each entity corresponds to a node in a graph and has relations with other nodes, such as *wrote-novel* or *president-of* and attributes, such as *population-size* or *date-of-birth*. This model facilitates the aggregation of information from various sources as needed at query time. For example, instead of having *Paris* as an entry in the *LOCATION* gazetteer, this model allows a QA system to “know” that *Paris* is a Roman establishment, an European capital and a French city, to “know” its population size, its mayor, what universities it has, and so on.

By putting the concepts at the centre, a QA system can directly use non-textual resources from different NLP sub-fields (e.g., IE, KR as shown in Section 2.4), meaning that questions which have answers beyond the scope of a particular passage can be addressed. Systems should use structured data available from authoritative sources, and only extract information from text if necessary. In this unified approach, text is no longer a bag of words and entity names, but a context which refers to classes and instances, embedding their properties and relations. During pre-processing, textual documents should be linked to the unique identifiers of the concepts and entities it mentions. For example, given a question about Paris (France), the system should not retrieve passages containing the keyword “Paris”, the way textual QA systems typically do, but instead retrieve only those

---

<sup>7</sup><http://www.w3.org/RDF/>

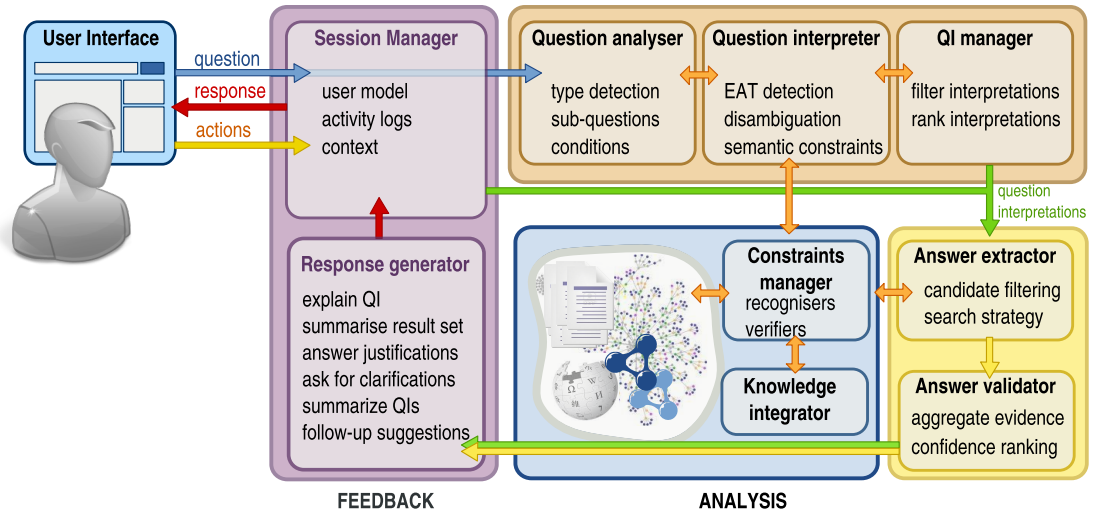


Figure 3.1: Generic architecture for open-domain QA

snippets which refer to the French capital (increased precision), even if not mentioning its name explicitly (increased recall).

One of the key ideas behind the proposed architecture is its ability to address ambiguity. Rather than employing a pipeline of standard NLP tools to disambiguate the input and produce the most probable output, the system must be aware of different sources of ambiguity and be able to deal with them directly during processing. Humans use common sense and their knowledge of the world when understanding and interpreting questions. To compensate for the lack of such abilities, during the analysis phase the system must generate all possible interpretations and rank them based on prior likelihood, their potential results and the context of the session. In the feedback phase, the system should be able to explain its ‘understanding’ of the question, and the criteria justifying the answers.

The three challenges posed by this architecture are: mapping natural language questions to semantic structures (‘understanding’ the question), representing available information according to the semantic model (finding and validating answers, integrating data sources), and generating metadata to enable feedback and interaction.

### 3.5.1 Analysis phase

In the **Analysis phase** the system performs question analysis and searches its knowledge base to identify possible answers and validate the correct ones (Figure 3.1). This phase corresponds to the main processing steps in the standard textual QA approach: question analysis, document retrieval, answer extraction and validation. Using a concept-centric representation enables atomic conditions in the questions to be addressed independently by specialised components. A linguistic analysis of the natural language request is initially carried out by the **question analyser** to determine the question type and to identify possible sub-questions and conditions. This yields probable grammatical clues regarding the structure of the question. The **question interpreter** then tries to map parts of the question to high-level semantic constraints. A possible “understanding” of the question consists of a composition of semantic constraints, information regarding the expected answer type and regarding the entities and the concepts mentioned by the question. Multiple question interpretations (QIs) are generated to reflect both uncertainty regarding the type of the constraints and how they should be combined. All the generated QIs are ranked and filtered by the QI manager enabling the system to address the most likely ones first.

For each generated QI, the system will search its knowledge base to find correct answers: named entities which satisfy the constraints specified by the question interpretation. The **answer extractor** no longer relies on a keyword-based IR engine, but on constraints verifiers: high-level processing primitives which make use of specialised methods and resources to check which named entities satisfy a particular type of constraint. Depending on the nature of these verifiers different strategies need to be employed to minimise processing time. Because multiple verifiers are used, the **answer validator** needs to normalise and aggregate multiple confidence scores to filter the list of answers.

During **question interpretation** the key issue is ambiguity: the same span of text can be part of several nested constraints and it could be read in different ways. Classic concepts

such as question type and expected answer type remain of interest, but they are part of the interpretation, and are no longer unique (fixed in the first step). The analysis tries to find **instances** (i.e., entities, either singleton - *Portugal*, or ambiguous - *Paris*), **concepts** (i.e., classes and types, either titles of category pages or RDF instances of WordNet and Cyc, and **relations** (based on the name of the relation, e.g., *bornIn*). Exact matches have the highest score, but additional hits may come from fuzzy matches: anchor text of hyperlinks, disambiguation links or RDF aliases. From verbs, the system can use lexico-semantic heuristics to find possible relations and properites from the RDF data.

The system also tries to find prepositional phrases or subordinate clauses and then identify the correct antecedent. The question is viewed as a sequence of constraints that have to be combined in a meaningful way. This is particularly true for encyclopaedic questions which are structurally complex.

Depending on the interpretation, the system can synthesise a simplified question, e.g. removing a constraint, and pass it to the question analyser. This can be used, for example, to compare the structure of the two questions. This shows that this architecture is not a strictly feed-forward pipeline, in contrast to the typical approach adopted by open-domain QA systems.

In this phase three types of ambiguity need to be addressed:

- **class ambiguity:** in an open domain it is very likely that a simple phrase used to specify a type of named entities can be interpreted in multiple ways by the system, depending on the breadth of its coverage, for example, “football player” is an ambiguous type because football covers several sports: *association football* (soccer), *American football* or *Australian football*. Usually the phrase needs to be automatically mapped to an intersection of more generic classes, which is challenging in an open domain;
- **instance ambiguity:** the larger the knowledge base the more collusions between entity names (eponyms), e.g., “Bush” can refer to persons, brand names, locations,

band names, an aircraft carrier and other entities. When an ambiguous name is found in the question, the system needs to address all the possible entities; the problem also manifests when searching for factual evidence in textual documents.

- **relation ambiguity:** this affects the way concepts and entities are inter-connected and the type of evidence the system needs to search for. An “European athlete” is an athlete competing for an European country, but an “European citizen” refers to a person’s citizenship.

If a question contains such phrases, the **question interpreter** should create several corresponding interpretations and answer them separately. The answer extractor is also affected by ambiguity when enforcing the semantic constraints. Using multiple, probabilistic constraint verifiers means that the **answer validator** needs to aggregate the resulting evidence, create a ranked list of answers and justify their correctness. While instance ambiguities are easier to deal with, others may require an ontology language that is both expressive and efficient so such inference is feasible/tractable. Systems may use mappings to OpenCyc to perform the inference required to select valid interpretations.

On the basis of the discussion above, it can be seen that the question analysis, document retrieval and answer extraction are mixed. This is natural: in order to understand the question, the system has to interpret the variables in the context of its knowledge base, and its interpretation can change depending on the data it discovers. Depending on the context, a question can have more than one meaning and a system must use methods that allow more than one interpretation at the same time.

### 3.5.2 Feedback phase

The **FEEDBACK PHASE** is the one responsible with presenting the results to the user, and is coordinated by the **session manager**. Depending on the purpose of the system this can mean a highly interactive Web interface, in which the user can disambiguate himself among the interpretations, modify the question’s constraints and give feedback to the

system, or just a simple interface in which the top ranked results are displayed with supporting information, optionally allowing users to dismiss erroneous results.

The response generator receives the question interpretations generated by the system together with their answers. It will generate a response which contains an explanation of the most likely QI ans well as provide justification of their answers. The system can summarize these if they are too many, or suggest ways in which the question can be altered. The remainder of the QIs can be used to show additional interpretations to facilitate disambiguation by the user.

An important step is ranking the interpretations based on the interaction with the user (session-sensitive context), the user model (collaborative filtering), the confidence of the answers found (the top answers should be presented first), and the similarity of the interpretations (group the interpretations that treat Paris as a city separately to those that treat Paris as a person).

If the selected interpretation of the question does not yield answers, the system can lower the thresholds (allow less probable entities), or relax the constraints to find answers and give feedback to the user. Whenever it produces results, the QA system needs to give informative feedback regarding which interpretation was used to create the returned set of answers.

Given the complexities of QA, exposing this type of information can benefit both the user experience and the error analysis: it is much easier to track individual errors back to their sources and to improve the system by analysing user actions. Collecting such information has been transformational in Web-search, product ranking, collaborative filtering and other domains.

The focus of this research is on the analysis phase, since it represents a novel approach to open-domain QA, a shift from a textual to a conceptual approach. The feedback phase is something less explored in QA and research is needed regarding this new functionality: how is feedback meta-data generated, how is it presented to the user, how much information is the user willing to accept, and so on.

### 3.5.3 Challenges

A crucial aspect of any QA system is the type of questions it can address. The conceptual approach proposed is focused on embedding generic world knowledge and multiple information sources at an encyclopaedic level. Achieving good performance would perhaps create the necessary conditions for the next generation of QA systems. The proposed Encyclopaedic QA uses questions that are expressed as a series of constraints combined with logical operands and perhaps aggregates. The simplest of questions (those having one or two constraints) correspond to factoid questions in standard textual QA approaches, while average questions would use *and*, *or*, *not* to combine several constraints in complex structures (arbitrary nesting). Questions using quantifiers and aggregates could be too computationally demanding and their semantics could be unclear under open domain assumptions of uncertainty and incompleteness. However an interesting type of questions are those requiring the system to generalise: *How high can planes fly?*, *What do novelists write?* or *What characterises Renaissance composers?* or those using aggregates: *Which rivers flow through more than 2 national capitals?*

One challenge posed by complex questions is decomposition. Existing work is focused on transforming a compound question into a series of (usually two) questions (Hartrumpf et al., 2009). The generated questions are processed sequentially and their results are combined according to the operand identified in the first step (previous work only use a limited set of combinations). The QA approach proposed can deal with more complex cases of question decomposition because it does not limit the number or type of constraints, nor their recursive nesting. Creating a correct question interpretation is not only a NL parsing problem: constituents also need to be assigned semantic constraints. As the new architecture relies on semantic relations between concepts it can exploit any source of world knowledge to interpret questions. For example “Romanian architects” can refer to the homonym category, while “Romanian volcanoes” does not have such a corresponding category. It can be represented as  $isA(?x, Volcano) \wedge locatedIn(?x, Romania)$ . The *isA* predicate must use subcategories as well as textual

extraction to determine whether or not an entity is a volcano. The *locatedIn* extractors must check the geographical constraint.

One of the challenges of such a system is the **expressiveness of the semantic representation**. This concerns both the ability to interpret questions as well as the corresponding mechanisms necessary to find answers. A key issue is whether or not a system is able to recognise questions or fragments of questions which it cannot deal with. Explaining the interpretations derived can help the user understand the limitations of the system and rephrase his question. The chosen formalism has to be generic enough to allow consistent coverage over a variety of topics and also accommodate the seamless addition of domain specific resources. The question answering system could aggregate a number of specialised components for a better treatment of popular domains (e.g., regarding sports events or geographic constraints), and also fall-back to a textual QA system or Web search engine.

A related challenge is that of storing and querying knowledge, as both coverage and time constraints are crucial from the user's point of view. In terms of automatic inference, an expressive formalism may not be able to scale very well, while a simpler, robust approach could provide better coverage and be more useful. Materialising all inferable information is likely to have unrealistic space requirements, whereas query time inference might not meet the time constraints expected by the user. One way to tackle this problem is to distinguish necessary pre-conditions (which can be queried efficiently in a database) and sufficient conditions (precise, but computationally expensive). In questions having several constraints applying pre-conditions first could significantly reduce the number of candidates. These can be further verified in parallel, allowing horizontal scalability of the system. One research issue is whether an automatic scheduling of processing steps is possible and what type of information is necessary to implement it.

When enforcing constraints, the system is likely to have multiple, overlapping sources. For example, there are various ways in which a person's nationality can be found in Wikipedia ranging from surface patterns for free text to semi-structured data such as

infoboxes or categories, in addition to structured data sources such as Freebase, or domain dependent databases. The system needs to be able to accommodate new sources, new IE capabilities and also deal with inconsistencies and contradictory evidence. A related issue is that of information transience: many properties vary in time (continuously or discretely). The knowledge base must record these changes and mark which value is considered “current”. Approximative values also need to be dealt with consistently. The system must also be able to deal with incompleteness of information: not all attributes and relations are known for all entities. The algorithms employed need to be based on these assumptions.

Until a significant amount of world knowledge is represented using adequate semantic formalisms, Wikipedia will be used as a proxy for this information. An important part of the system is how to represent world knowledge as a semantic graph which allows “browsing” for semantically relevant data, instead of using keyword-based retrieval and *tf-idf* word weighting. The hyper-link structure of Wikipedia can present a starting point, but not all links are marked in a Wikipedia page due to stylistic concerns. To recover the missing links, named entity disambiguation (see Section 2.4.3) needs to be performed. An important enhancement is to consider weighted links in order to capture the strength of association between entities (and use it to estimate confidence). Applying a spreading activation algorithm on this graph could help detect which topics are most relevant to a user’s question, to better rank interpretations and generate feedback-meta data.

An advantage of working at a semantic layer is that language becomes less important. The knowledge base itself is language independent. It allows mappings to different languages via the IE framework. Entities, categories or attributes no longer need translation or transliteration, instead being identified by their URI<sup>8</sup>: the names and aliases are just labels used in a particular language. This makes it possible to extract answers from other language versions of Wikipedia which have more information about a given topic. Different language versions can be used for validating and ranking answers.

---

<sup>8</sup>Uniform Resource Identifier

There are some similarities between recent QA approaches (Section 2.2) such as the DeepQA architecture used in Watson (Ferrucci et al., 2010) and the architecture employed by FREyA (Damljanovic et al., 2011) and the novel semantic QA architecture developed in this thesis and implemented in the EQUAL system (Chapter 4). These independent developments which work to incorporate more semantic information and multiple sources of evidence into QA provide further evidence that it is time for a shift in the field, and that this new/semantic approach can produce significantly better performance on more complex types of questions.

### 3.6 Conclusions

This chapter argues that a paradigm shift is needed in open-domain QA and proposes a novel approach which uses encyclopaedic knowledge. It was argued that the standard evaluation campaigns, focused on questions which can be answered by single snippets, have unwillingly biased research towards a textual approach characterised by the use of keyword-based IR engines and by the reliance on lexical similarity between the question and the answer bearing snippet. For this reason, despite the complexity of the task, little progress was made into adding semantic world knowledge or common-sense inference. The ability of the standard non-interactive pipeline architecture to deal with NL ambiguity is questioned, as the propagation of errors along the QA pipeline limits overall performance.

Two changes are proposed to address these issues. Firstly, encyclopaedic QA is seen as new evaluation test-bed focused on open list questions composed of several constraints which require information fusion from several documents and basic world understanding. In addition to synthetic benchmarks, new evaluation methods should be employed to reflect real world performance of QA as an information access tool, as experienced by users. Secondly, a paradigmatic shift is proposed for QA systems from words to concepts. The core of the system should be based on Semantic Web technologies which allow the integration of various information sources and knowledge bases, enabling QA systems to take advantage of advances in domains such as KR and IE.

A new architecture is proposed consisting of two major components: **Analysis**, which deals with interpreting questions and finding answers, and **Feedback**, which is responsible with interacting with the user, explaining the results, and asking for clarifications. The central notion is that of semantic interpretation: the formal representation of the question's meaning, mediating the user's needs and the system's capabilities, enabling ambiguity resolution and user feedback. The major challenges for this approach were then outlined. The next chapter describes two implementations of the **Analysis** component of the proposed architecture.

## Chapter 4

# EQUAL: Encyclopaedic QUestion

## Answering system for List questions

### 4.1 Overview

The previous chapter argued for a shift in focus from factoid questions towards encyclopaedic questions, open list questions whose answers are named entities. These explicitly ask for a more or less specific type of named entities and usually contain additional conditions or constraints the entities must satisfy. To answer this type of questions, QA systems must fuse information from various sources and establish the correctness of each candidate answer, a more difficult problem than standard factoid QA where a single correct answer must be ranked in the top results returned. The additional challenge posed by encyclopaedic QA is to combine simple facts in order to deduce answers which are not explicitly present in a single textual snippet. To do so, systems need a repository of named entities to which they can link textual documents, database entries, knowledge bases and other sources of information. Wikipedia is a perfect candidate for providing this type of information.

This chapter presents EQUAL – Encyclopaedic QUestion Answering for List questions, a question answering system which implements the first phase of the QA architecture proposed in this thesis, the one responsible with interpreting questions and finding answers. As discussed in Section 3.5, the proposed approach revolves around concepts, not

words. EQUAL relies on structural information extracted from Wikipedia to answer open-list questions. Instead of using it as a textual collection, EQUAL views the encyclopaedia as a graph of entities: pages are nodes and their relations are edges.

EQUAL achieved the highest score in two question answering tasks, GikiP (Santos et al., 2009) and GikiCLEF (Santos and Cabral, 2009*a,b*) which are briefly described in Section 4.2. Unlike the standard textual QA approach, EQUAL does not rely on identifying the answer within a text snippet by using keyword retrieval. Instead, it explores the Wikipedia page graph extracting and aggregating information from multiple documents and enforcing semantic constraints. Section 4.3 describes the first version of the system which was employed in the GikiP pilot task. This system was further developed and extended with more features for the GikiCLEF competition. The improved system is described in Section 4.4. The results in the two competitions are analysed in Section 4.5. Based on an error analysis carried out on the GikiCLEF data, Section 4.6 discusses challenges posed by the new approach.

## 4.2 Geographic question answering

The previous chapter argued that in order to advance the state of the art in QA it is necessary to use more complex questions. Questions proposed by encyclopaedic QA are difficult because they are open-list questions with multiple constraints. Information supporting each answer is unlikely to be found in a single paragraph and simple ranking methods exploiting redundancy of information on the Web can no longer assume a unique correct answer. List questions have been previously employed in competitions such as CLEF and TREC, but they only accounted for a small fraction of the total number of questions and they could usually be answered by a single paragraph in the target collection, i.e., an enumeration. Two of the recent evaluation campaigns focusing on encyclopaedic list questions were GikiP (Santos et al., 2009) and GikiCLEF (Santos and Cabral, 2009*b*) and are described next.

|  |
|--|
| GP3: Portuguese rivers that flow through cities with more than 150000 inhabitants<br><i>Douro, Mondego</i>                     |
| GP9: Composers of Renaissance music born in Germany<br><i>Arnolt Schlick, Christoph Demantius, Conrad Paumann (12 more...)</i> |
| GP6: Which Australian mountains are higher than 2000m?<br><i>Mount Kosciuszko, Mawson Peak, Mount Jagungal, Mount Townsend</i> |
| (a) GikiP 2008 topics  |
| GC09-01: List the Italian places which Ernest Hemingway visited during his life.   |
| GC09-04: Name Romanian poets who published volumes with ballads until 1941.  |
| GC09-09: Name places where Goethe fell in love.  |
| GC09-11: What Belgians won the Ronde van Vlaanderen exactly twice?   |
| GC09-29: Places above the Arctic circle with a population larger than 100,000 people   |
| GC09-37: Which Norwegian musicians were convicted for burning churches?  |
| (b) GikiCLEF 2009 topics   |

Figure 4.1: Sample of GikiP and GikiCLEF topics

**GikiP** was a pilot QA task that took place in CLEF 2008 and combined QA and geographic IR. The requests for information consisted of natural language topics, phrased either as a question or as an imperative retrieval directive, which had some sort of geographical constraint. The answer was supposed to be a set of results: Wikipedia articles describing the entities that correspond to the information need expressed in the topic (see Figure 4.1a). One of the aims of GikiP was to create truly multi-lingual systems that are able to exploit information from Wikipedia in any of the three participating languages: English, German and Portuguese. To this end, human translations of the topics were made available, enabling participants to exploit the multi-lingual encyclopaedia and perform multilingual processing. Each topic also had a brief narrative description of the information requested. These were not used by the systems, but instead helped human assessors judge answers' correctness. The GikiP pilot consisted of 23 topics: 8 for development and 15 for testing.

**GikiCLEF** was another evaluation task organised at CLEF 2009 and was a successor of GikiP. As its predecessor, it also consisted of geographically challenging information requests (see Figure 4.1b). Systems had to search the Wikipedia collections and return answers as lists of document titles. GikiCLEF emphasised the multi-lingual component: 10 language versions of Wikipedia were used: Bulgarian, Dutch, English, German,

Italian, Norwegian (both Bokmål and Nynorsk), Portuguese, Romanian and Spanish. The evaluation campaign used 24 development topics and 50 test topics, all translated into the 10 languages of the competition.

### 4.3 EQUAL in GikiP

This section describes the first implementation of EQUAL which participated in GikiP. The first step of the method consists of pre-processing the Wikipedia dump distributed by the task organisers and transforming it into an entity graph. Some articles denote named entities, others refer to events or topics, while categories and list pages denote possibly ambiguous sets of entities (see Section 2.3). For each node of the graph, two types of information were extracted from its corresponding article: node properties, e.g., title, infobox attributes, plain text, alternative names; and links, i.e., interwiki links for crosslingual information, wiki links for semantic relatedness and category links for semantic type information. The infobox attributes were extracted using regular expressions from the wiki source text and added as additional node properties. Links between nodes were not labelled, e.g., no distinction was made between topic— and list—categories because this information is not explicit in the Wikipedia markup.

At query time, the system analyses the information associated with a node to extract simple facts, such as population size of a city, elevation of a mountain or date of birth of a person, or simple binary relations, such as the city or the country a person was born in. Usually this information was extracted from infoboxes, but sometimes facts needed to be extracted from the plain text of the article, using lexical patterns.

In the GikiP pilot task, the **feedback** phase is not important because there is no human interaction, thus the system only implemented the **analysis** phase of the proposed architecture. Processing consists of two steps: the *question analysis* and *answer extraction*. In the first step, the structure of the question is analysed and split in two parts: the **domain** of possible answers which specifies the expected answer type, and the **filter**

which contains the conditions that the correct answers must satisfy. The split is made based on simple linguistic information: the first noun phrase indicates the type of named entities sought by the question (the domain). The first relative pronoun or verb which follows this phrase marks the beginning of the filter. The system checks what semantic constraints are compatible with each sub-part of the question. Based on an analysis of the training questions, both parts may consist of one or two constraints.

The **domain**, the first part of the question, contains the phrase used to specify the type of named entities sought, also known as the expected answer type. The system maps it to the title of a Wikipedia category. This phrase contains a noun in plural form which is matched to a generic category. Then a subcategory is chosen, whose title contains the most of the remaining words and their synonyms retrieved from WordNet. The initial set of candidate answers consists of all the articles directly linked to this subcategory (rank I) and all the articles linked to any subcategory (rank II). This breadth-first distinction was employed to limit ambiguity, because not all categories assigned to an article convey a type relation (see Section 2.3); some categories mark an article as relevant to a particular subject or topic. In Wikipedia, both types of category links are marked up in the same way. For example, in GikiP topic **GP4** *Which Swiss cantons border Germany?* the domain is "Swiss cantons" which is mapped to *Category:Cantons of Switzerland*, which lists all the relevant candidate articles(rank I). This category also has several topic sub-categories, e.g., *Category:Canton of Bern* which contains articles such as *Emmental (cheese)*, *Bernese Mountain Dog* or *Swiss peasant war of 1653* which have rank II. These articles should not be considered candidate answers.

The **filter** consists of one or two conditions that a candidate answer needs to satisfy. Two types of filters were used for GikiP: *entity* and *attribute*. The entity filter is identified in questions which mention an explicit named entity; it denotes that some relation must exist between this named entity and the candidate answers. For example in topic **GP4** *Which Swiss cantons border Germany?* the filter is "border Germany". Because it contains a named entity, this is considered an entity filter. The most likely article in

Wikipedia that the name can refer to is then identified. In the GikiP there usually was a straightforward match between the article title and the entity name. To satisfy an entity filter, a candidate article must be connected to the article corresponding to the named entity. To check this, the system searched either for a wiki link between the two articles or at least for an explicit mention to each other in their corresponding text. Eight of the fifteen topics had an entity filter, e.g. **GP1** *Which waterfalls were used in the film "The last of the Mohicans"?*, **GP12** *Places where Goethe lived*, or **GP14** *Brazilian architects who designed buildings in Europe*.

An attribute filter constrains the values of a particular property a candidate answer needs to have, e.g. **GP6** *Which Australian mountains are higher than 2000m?*, **GP9** *Composers of Renaissance music born in Germany?* or **GP10** *Polynesian islands with more than 5000 inhabitants*. To satisfy an attribute filter, the system tries to extract the corresponding fact from the body of the article or from its infobox, and if successful, compares its value with the criterion specified in the question. In the GikiP test set, the attributes were related to the geographical domain: *height/elevation* (1 topic), *nationality* (2 topics), *population* (3 topics) and *length* (1 topic). Candidate articles from which the fact could not be extracted were dismissed. The selection criterion consists of one or more reference values specified by the question. The value extracted from the candidate document is compared with the reference values. The comparison can be numeric: *moreThan* (5 topics), or set-based: *inList* (1 topic), *not\_inList* (1 topic).

Results found in the English Wikipedia were mapped using the inter-wiki links into the other two languages, thus EQUAL is essentially a mono-lingual system. To exploit cross-lingual redundancy and rank the best results, the system should search for candidate answers in each of the three languages and then combine the three resulting sets. This approach was deemed infeasible for two reasons. Firstly, in a realistic setting, users are unlikely to translate their own questions in more languages, therefore, relying on user-translated topics limits the scope of the system. Secondly, different language versions of Wikipedia have different coverage and structure, usually yielding distinct sets of answers

that only partially overlap. Considerable amounts of training data are necessary to create a reliable answer validation, i.e., not just intersection/union of the sets.

Analysis of the results reveals advantages of this approach over traditional textual QA systems:

- ability to process complex entity types: the expected answer type is not limited to gazetteer lists or generic NER super-tags – information present in the category graph is used instead, e.g., ‘waterfalls’, ‘Dutch painters’ or ‘composers of Renaissance music’;
- multilingual processing: by navigating the graph using inter-wiki links, the system can search several language versions for information, e.g., it is possible that the German language version of Wikipedia is more complete with regard to topics such as Goethe and German cities. Checking wiki links (entity filter) and category links (domain) is largely language independent, while extracting facts (attribute filter) requires language dependent resources;
- semi-structured information analysis: in addition to textual information, semi-structured data, RDF repositories and databases can be exploited as a source of reliable attributes and properties, which can be integrated in the system (see Section 2.4.4).

The system implements the architecture proposed in Section 3.5 using a rather small set of generic constraints, primarily because of the limited amount of training data. Although more expressive filters are needed to correctly address all the GikiP questions, the system ranked first amongst 8 submissions by 3 participants, outperforming both manual and semi-automatic submissions (Santos et al., 2009). The evaluation results are presented in Section 4.5.

## 4.4 EQUAL in GikiCLEF

For the second competition, EQUAL was extended by adding new semantic constraints which enable more expressive question interpretations to be generated. Because GikiCLEF

is a non-interactive task, the system implements the analysis phase. To tackle ambiguity and uncertainty, the system may generate several question interpretations which are addressed sequentially. This approach relies on the assumption that an erroneous interpretation will likely yield either too many results or no results at all.

The task of interpreting questions is difficult due to the ambiguity which characterises natural language: very similar natural language expressions can have very different meanings, while the same meaning can be expressed in a variety of forms. Closed domain QA systems have a narrow scope and rely on an unambiguous specification, such as a database schema or an ontology, to limit ambiguities. In open-domain QA, even a simple question can have several interpretations, each corresponding to a different ‘understanding’. In GikiCLEF a question interpretation is a composition of semantic constraints involving entities, relations and properties. An analysis of the sample topics was carried out to determine a set of generic constraints relevant to the entity-graph model employed by EQUAL. These are listed below:

- **EAT** (expected answer type): indicates the type of entities sought in the question; only entities of this type are considered valid answer candidates;
- **type**: constraint which indicates a type that an entity/article must have;
- **entity**: verifies that some connection exists between two entities, usually a candidate answer and a named entity mentioned in the question;
- **relation**: verifies that a specified relation holds between a pair of entities, typically denoted by verbs, e.g., *born in*, *played for* or *moved to*;
- **property**: restricts a set of entities to a subset characterised by a certain property, e.g., *population size*, *height* or *surface area*;
- **geographic**: checks if an article is included in or part of a larger geographic entity, e.g., *a country* or *a region*;
- **temporal**: constrains the temporal interval of a relation or event;
- **introduction**: marks the phrase used to start the question, this chunk is removed from the question.

EQUAL uses a simplified sequential model of the general architecture (see Section 3.5), allowing it to stop when one of the interpretations yields results. The system is based on a rule-based heuristic approach which was developed using the GikiCLEF 2009 training question set.

#### 4.4.1 Question parsing

In EQUAL, the question analyser examines the structure of a GikiCLEF topic using linguistic information which results in a set of possible constituents of the input question. This relies on certain classes of words that are typically delimiters, i.e., prepositions, relative pronouns, conjunctions, adverbs and verbs, and the output of the Stanford Parser<sup>1</sup> (Klein and Manning, 2003).

The chunking algorithm determines which of the chunks indicates the semantic type of the expected answers. In general, this constituent is marked by an interrogative or a demonstrative pronoun, but in the GikiCLEF collection, after the redundant introduction is removed, the first chunk always corresponds to the EAT constraint and it always contains a noun in plural form. For example, topic **GC09-18** *In which Tuscan provinces is Chianti produced?* is split in two: “Tuscan provinces” and “Chianti produced”. The irrelevant words, *in*, *which* and *is* are ignored. The first constituent is the type-phrase, and its head noun is extracted separately. The second constituent corresponds to conditions and is split into chunks, see Table 4.1 (part of speech information and syntactic information produced by Stanford parser is not shown).

#### 4.4.2 Semantic interpretation and constraints

The question interpreter is responsible with determining which are the compatible semantic constraint types for each chunk (sub-filter) and creating corresponding semantic interpretations. To do so, this module needs to deal with ambiguities in the context of the

---

<sup>1</sup><http://nlp.stanford.edu/software/lex-parser.shtml>

Table 4.1: Decomposition of topics into chunks by the question analyser

|             |  |
|-------------|--|
| topic       | Name Romanian writers who were living in USA in 2003                     |
| type-phrase | Romanian writers   |
| type-noun   | writers  |
| conditions  | living in USA in 2003  |
| chunks      | (VP living) (PP in USA) (PP in 2003)                                     |
| topic       | Which Brazilian football players play in clubs in the Iberian Peninsula? |
| type-phrase | Brazilian football players   |
| type-noun   | football players   |
| conditions  | play in clubs in the Iberian Peninsula                                   |
| chunks      | (VP play) (PP in clubs) (PP in the Iberian Peninsula)                    |
| topic       | In which Tuscan provinces is Chianti produced?                           |
| type-phrase | Tuscan provinces   |
| type-noun   | provinces  |
| conditions  | Chianti produced   |
| chunks      | (NP Chianti) (VP produced)   |

semantic constraints that the system ‘understands’ and the way the relevant information is present in Wikipedia and any associated data sources. The main ambiguities that are dealt with during this phase are:

- **referential ambiguity:** the constraint refers to entities or concepts which need to be disambiguated with respect to the knowledge base (“*Who is the mayor of Paris*”? – several possible cities vs. “*Who is the director of Paris*”? – a 2008 film);
- **structural ambiguity:** the same constraint can be attached to more than one head, e.g., “*Brazilian architects who designed buildings in Europe*” has a prepositional phrase attachment ambiguity: the question most likely refers to buildings which are located in Europe, but it is possible that buildings outside of Europe were designed by an architect while living in a European country. Unlike humans who rely on intuition, the system needs to consider both possibilities.
- **type ambiguity:** the constraint can be interpreted in more than one way (“*What novels did Orwell publish before 1984?*” – temporal constraint = year; entity constraint = novel).

The system examines each syntactic constituent produced by the question analyser to determine its possible semantic type. For example phrases such as *greater than* or *more than* indicate a numeric property constraint, while the preposition *in* can indicate several types of constraints, e.g., geographical *in USA*, temporal *in 2003* or *in World War II*, or even an entity constraint *in Paris (film)*. For each type of constraint, EQUAL has a set of specific rules which test if a given chunk contains the necessary information. The geographic constraint needs the name of a geographic region, usually a country, and a preposition which specifying inclusion or exclusion (*in*, *within* vs. *outside of*). A temporal constraint requires a reference period (a year, a century, a span of time given by two years) and one of the following prepositions *before*, *after*, *during* and *in*. An entity constraint only requires the presence of a named entity, while a type constraint needs a noun in a plural form which can be matched to a Wikipedia category.

Once the compatible constraint types are determined for each chunk, question interpretations are generated. The parse tree given by the syntactic parser is used to assign the most likely dependent for each chunk. This information is used to determine how to combine the semantic constraints. For example, the relation constraint links the candidate answers with a type or an entity constraint. Geographical constraints are typically applied to the chunk preceding them, while the temporal constraint refers to the relation constraint. EQUAL uses a very generic model and cannot take advantage of ontologies the way closed domain QA systems do. It exploits the fact that in GikiCLEF there are usually there are only two or three constraints, so it generates all possible combinations.

An interactive QA system can ask the user for clarifications, e.g., by providing a ranked list of alternatives. In the non-interactive paradigm, a system needs to select the choice most likely to match the expectations of the user. Where there is ambiguity, the distinct interpretations are addressed sequentially. EQUAL analyses all the wiki links in Wikipedia to determine for each anchor text which is the most frequent target article. This information is used to determine the most likely article for each entity name

mentioned by the question and to prioritise the generated interpretations. These are passed to the answer extractor.

[Name] || Romanian writers || [who] [were] living | in USA | in 2003  
 $\{_{intro}Name\} \{_{eat}Romanian\ writers\} \{_{rel}living\} \{_{geo}in\ USA\} \{_{temp}in\ 2003\}$

Figure 4.2: Example: decomposition of question GC09-32 into constraints

#### 4.4.3 Answer extraction

The GikiCLEF question set is composed of entity list questions. While these are regarded as more difficult than factoid questions (most factoid questions can be rephrased as list questions which have exactly one correct answer), they have three important characteristics which make them accessible:

1. encyclopaedic domain — since all questions are asking about entities represented in Wikipedia, certain assumptions regarding the “domain” can be made;
2. structural uniformity — most questions have few constraints combined using a small set of syntactic patterns;
3. articles as answers — the system can search for/validate candidate answers using disambiguated links in Wikipedia, instead of ambiguous entity names (the abbreviation “USA” is used for more than thirty entities).

From Wikipedia’s redirect and disambiguation pages, entity aliases are extracted to be used at the question analysis stage. Together with anchor text statistics they help rank the most likely entities referred to in the question. The articles describing these entities are indexed and the links between them are stored in a database, allowing access to the set of pages that are linked to/from a given article. These are used for the entity filter, and as a first step in the relation filter. The categories and their links are stored in a separate table for navigating the category graph.

The category folksonomy is used to derive type information for entities. Being a folksonomy, the relations between categories are not specified, which results in

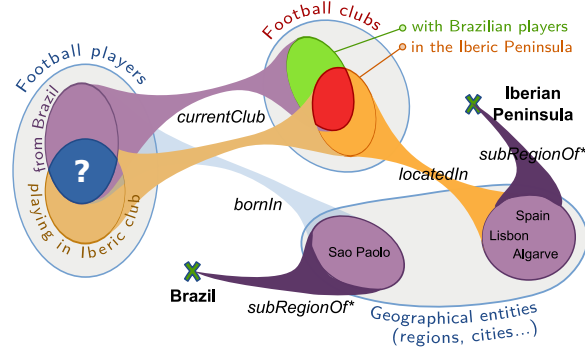
ambiguities. There are both type and topic categories, but this distinction is not explicit in the markup. Although most of the time a direct link from an article to a category denotes a type relation, precautions should be taken when assuming transitivity. Section 2.4 describes DBpedia and Yago, which use more advanced methods to disambiguate these links. Because these tools were still being developed, EQUAL employs a simple rule which limits transitivity: subcategories are processed in a breadth-first order, proceeding to the next level only if no answers have been found yet. Transitivity filtering refers to categories which do not have a plural noun in their title, or which have a corresponding article with exactly the same name (e.g., *London*). This rule is designed to limit the recursive exploration of subcategories in the Wikipedia graph.

Apart from the straightforward query for the set of categories assigned to a page, EQUAL can also check if the page is directly or indirectly within a specified category, by starting with this category and exploring, level by level, the pages indirectly connected to category it, with or without transitivity filtering.

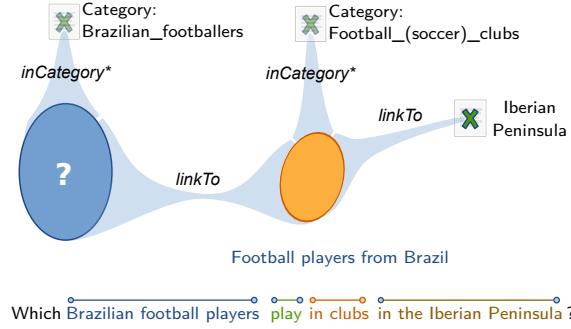
The advantage of the categories graph is that it exhibits various granularity levels. For example, it is possible to directly map the question span “Brazilian football players” to the category *Brazilian footballers*, instead of having to deduce the semantics of the individual words, such as checking which “football players” were “born in Brazil” (see Figure 4.3a).

#### 4.4.4 Constraint verifiers

EQUAL explores the Wikipedia graph verifying which entities satisfy the constraints of the current question interpretation. The semantics of the constraints themselves in the context of Wikipedia are defined by *constraint verifiers*, i.e., the actual implementation which verifies whether a particular constraint holds. A constraint has several specialised verifiers to take advantage of all the existing sources of information. For example, in Wikipedia, *geographic containment* can be expressed using demonym modifiers (e.g., *Nepalese*), categories (e.g., *Mountains of Nepal*), tables, infoboxes and text. However, the system could also use a geographical Web Service to verify relations such as containment,



(a) Semantic interpretation using DBpedia relations



(b) Semantic interpretation by EQUAL

Figure 4.3: Example of semantic question interpretation for the question: *Which  $\{_{eat} \text{Brazilian football players}\} \{_{rel} \text{play in}\} \{_{type} \text{clubs}\} \{_{geo} \text{in the Iberian Peninsula}\}$ ?*

neighbouring and distance using an article's geo coordinates. The following constraints verifiers were used:

**Type constraint.** It is used to determine the set of named entities which have a specific type. It is typically used to determine the initial set of candidate answers sought by the question, i.e., the expected answer type. The system first matches the phrase used to denote this type to a Wikipedia category. If the match is not straightforward, the system considers the first noun in plural form as the most generic hypernym to select a category. For example, *Which eight-thousands are at least partly in Nepal?* is directly matched to *Category:Eight-thousands*, *List the 5 Italian regions with a special statute.* is partially matched to *Category:Regions of Italy*, and *List the left side tributaries of the Po river* is only matched to *Category:Tributaries*. Apart from the EAT, some questions can have a second constraint, for example in *Which Portuguese rivers flow through cities with more than 150.000 inhabitants.*

**Entity constraint.** Filters a set of articles based on which of them have some connection

to a given named entity. This entity is usually mentioned in the question, e.g. *In which Tuscan provinces is Chianti produced?*: the set of candidate articles which are provinces in Tuscany are filtered based on having links with *Chianti*. To enforce this type of constraint the system checks for each article whether or not it mentions or is mentioned by the article corresponding to the given named entity, in this case Chianti. EQUAL uses both the links table and also the article text to identify mentions, to account for cases where the link was not explicitly marked by the Wikipedia editors.

**Relation constraint.** This constraint is identified in questions that specify a relation between the entities, and usually co-occurs alongside an entity constraint or a type constraint. EQUAL must then confirm that the context of the mention page is ‘compatible’ with the constraint identified in the question. Initial experiments revealed that in many cases testing the compatibility needs more than synonym expansion or WordNet similarity measures. Therefore this constraint was ignored in the GikiCLEF test set. This means that for some topics, such as GC09-01: *List the Italian places which Ernest Hemingway visited during his life* and GC09-09: *Name places where Goethe fell in love*, EQUAL uses a more general interpretation, which usually leads to inaccurate answers and a decrease in precision.

**Temporal constraint.** When interpreting questions that have a time constraint, this is applied to further filter the candidates satisfying the relation constraint. The implementation uses a temporal annotator developed for the QALL-ME project<sup>2</sup> to transform temporal expressions to time intervals, and then tests if the context of the mention is compatible with the constraint from the question. This approach has similar limitations as the relation constraint: matches in the context can be irrelevant to the actual entity mention, leading to false positives. Therefore, for GikiCLEF EQUAL used a simpler test: if the constraint is one year, then it must be textually present somewhere in the article’s content. All other forms of temporal constraints are ignored, even if in other interpretations the question span could be considered an entity constraint, e.g., GC09-42:

---

<sup>2</sup><http://qallme.fbk.eu/>

“Political parties in the National Council of Austria which were founded **after the end of World War II**”.

**Property constraint.** EQUAL uses both the infobox content and the article text to extract numeric attributes. The comparatives and other quantifiers from the question are observed. If the noun denoting the property is not found or a numeric value cannot be extracted, the constraint is not satisfied and the candidate article is discarded (favouring precision).

**Geographic constraint.** This constraint is used to check if an article describes an entity located in or associated with a country. Instead of processing the text of the article, the fastest way to check this is to examine the article’s categories. There are two ways in which category titles capture this information. The first way is to combine a demonym with the type in plural form, for example ‘Romanian poets’, ‘Dutch classical violinists’, ‘Brazilian footballers’ and ‘European countries’. The system uses a large list of demonyms extracted from Wikipedia to deal with former countries names, e.g., ‘Mercian people’ or ‘Northumbrian folkloric beings’. The second way is to combine the hypernym in plural form with the country name: \$TYPE (BY|OF|FROM|IN) \$COUNTRY for example ‘Mountains of Nepal’, ‘Monarchies in Europe’, ‘World Heritage Sites in France’ or ‘Rivers of Brazil’. This strategy is also used when searching for the best category match for the EAT. If such a pattern does not apply, as a fall-back strategy, EQUAL interprets this as an entity constraint.

**Introduction constraint.** A list of interrogative pronouns (e.g., *which*), imperative verbs (e.g., *list*, *name*) and declarative constructions (e.g., *I’m interested...*) which was compiled from the QALL-ME benchmark (Cabrio et al., 2008) is used to identify the spans of text that typically introduce a question. These spans can be considered list-question markers, but in the context of GikiCLEF this information is redundant and these spans are ignored in all subsequent processing.

The results achieved by the systems are presented in the next section. Insights into the challenges posed by the approach are described in Section 4.6 based on an error analysis.

## 4.5 Results

### 4.5.1 Evaluation metrics

The organisers of GikiP proposed an evaluation measure which rewards *precision* (returning correct entity names of the adequate type, not the documents where the answer can be found), *recall* (finding many correct entities) and *multilinguality* (ability to find answers in several languages). All questions are open list questions. In this section, an answer refers to one named entity returned by the system in response to a question. A system's final score  $S_1 = \frac{1}{N} \cdot \sum_{i=1}^N S_1(i)$  is the average of its scores per topic, which are computed regardless of language, according to the following formula:  $S_1(i) = mult \cdot C_i \cdot C_i / N_i$  where  $C_i = \sum_{lang} C_i^{lang}$  is the number of correct answers for topic  $i$  across all languages,  $N_i = \sum_{lang} N_i^{lang}$  is the number of answers given by the system, and *mult* is the bonus factor representing the number of languages processed by the system (possible values are 1,2 and 3).

The GikiCLEF organisers have chosen topics that have a strong cultural/national bias (Santos et al., 2010), i.e., one language version of Wikipedia is very likely to contain answers, while other language versions include few or even no answers. This bias was used in order to reward systems that process multilingual pages. The evaluation measure initially used in GikiP was also adjusted to better reflect the number of languages in which answers are found for each topic, because the constant factor *mult* represents in fact just an upper bound accounting for the number of languages a system is processing regardless of whether or not it finds correct answers. Therefore, the score was computed per language instead of per topic using the formula  $S_2^{lang} = C^{lang} \cdot C^{lang} / N^{lang}$ , where  $C^{lang} = \sum_i C_i^{lang}$  and  $N^{lang} = \sum_i N_i^{lang}$ . The final score of a system is the sum of its individual language scores  $S_2 = \sum_{lang} S_2^{lang}$ .

Despite providing an accurate ranking of the systems' performance, there are three main criticisms of the official GikiCLEF scoring measure: 1) bias towards precision, 2)

Table 4.2: GikiP official results for all participants

| # | System | Answers | Correct | Precision | Languages | Score |
|---|--------|---------|---------|-----------|-----------|-------|
| 1 | c11    | 123     | 94      | 0.63      | 3         | 16.14 |
| 2 | c1     | 218     | 120     | 0.55      | 2         | 10.71 |
| 3 | c7     | 79      | 9       | 0.11      | 3         | 0.70  |
| 4 | c8     | 76      | 7       | 0.08      | 3         | 0.56  |
| 5 | c9     | 84      | 7       | 0.07      | 3         | 0.50  |
| 6 | c5     | 199     | 9       | 0.04      | 3         | 0.34  |
| 7 | c10    | 189     | 9       | 0.05      | 3         | 0.32  |
| 8 | c6     | 171     | 7       | 0.04      | 3         | 0.26  |

non-normalised range and 3) bias towards topics with many answers. The bias towards precision stems from the fact that not all correct answers are known, thus recall cannot be computed. Since the score range is not normalised, performance is not comparable on distinct topics sets, i.e., development and testing. In addition to the official score, performance for list questions can be measured using metrics traditionally used in IR-based, such as precision, recall and F-measure. To compute recall, the number of known correct answers in the answer pool is used as a reference, although this is only a lower bound for the absolute number of correct answers. Scores for a particular language can be computed using both micro- and macro-averaging. For example micro-averaged precision gives the probability that a random named entity returned by the system is correct, while macro-averaged precision shows the expected performance for a random question:

$$\begin{aligned} \text{micro-averaged precision } P_{micro}^{en} &= \frac{\sum_{i=1}^n C_i^{en}}{\sum_{i=1}^n N_i^{en}} \text{ and recall } R_{micro}^{en} = \frac{\sum_{i=1}^n C_i^{en}}{\sum_{i=1}^n K_i^{en}}, \\ \text{macro-averaged precision } P_{macro}^{en} &= \frac{1}{n} \sum_{i=1}^n \frac{C_i^{en}}{N_i^{en}} \text{ and recall } R_{macro}^{en} = \frac{1}{n} \sum_{i=1}^n \frac{C_i^{en}}{K_i^{en}}. \end{aligned}$$

#### 4.5.2 GikiP results

In GikiP, despite having a few generic constraints, the first version of the EQUAL system, referred to as c11 in the official results in Table 4.2, outperformed both automatic and semi-automatic participants. The most likely reason for this success is that most property constraints are easy to be verified using infobox attributes extracted by simple regular expressions from the wikisource text of the article. In addition, by using the entire text

Table 4.3: GikiP normalised results

| System        | micro-averaging |              |                | macro-averaging |              |                |
|---------------|-----------------|--------------|----------------|-----------------|--------------|----------------|
|               | P               | RR           | F <sub>1</sub> | P               | RR           | F <sub>1</sub> |
| RENOIR - c1   | 0.550           | <b>0.690</b> | 0.612          | 0.551           | <b>0.696</b> | <b>0.615</b>   |
| EQUAL - c11   | <b>0.764</b>    | 0.540        | <b>0.633</b>   | <b>0.634</b>    | 0.573        | 0.602          |
| GIRSA-WP - c7 | 0.114           | 0.052        | 0.071          | 0.107           | 0.060        | 0.077          |

of an article as support for unspecified relations also affords high precision due to the use of fine-grained types. For example, a page which describes a *Swiss canton* and also has a link to *Germany* is very likely to be a correct answer for GP4 *Which Swiss cantons border Germany?* The two most common causes for errors are inaccurate category mappings, e.g., “wars” in GP5 *Name all wars that occurred on Greek soil*, and insufficient expressiveness of the interpretations created for GP5 and GP15 *French bridges which were in construction between 1980 and 1990*. The verifiers implemented are precise, at the cost of recall: for 10/15 topics, the system returned no false positive answers (100% precision), but it only finds the complete set of answers for 3/15 topics (low recall).

The third criticism of the official scoring measure  $S_1$  is illustrated by Table 4.4: the five topics (30% of the test set) having the most correct answers contribute almost 80% of the total score of the system. This is also true for the second ranked system. In the case of perfect output, topic GP7 yields a higher score than nine other topics combined.

GIRSA-WP (GIRSA for Wikipedia) is a fully-automatic, hybrid system which merges results from InSicht, an open-domain QA system (Hartrumpf, 2005), with a list of relevant documents retrieved by GIRSA, a system for textual geographic IR (Leveling and Hartrumpf, 2008). To select answers from this list, the system first extracts the generic concept which indicates the expected answer type, e.g., for topic GP4 *Which Swiss cantons border Germany?* the extracted concept is *canton*, which is an artificial geographical entity denoting a kind of regional institution. Then, each document title is parsed by WOCADI (Hartrumpf, 2003), a syntactico-semantic parser for German text, in order to match its ontological sort and the semantic features to those of the concept

extracted from the topic. The documents for which this match is successful are returned as answers.

To deal with the complex topics in GikiP, InSicht employs question decomposition (Hartrumpf, 2008) to generate several simple questions, e.g., topic GP4 *Which Swiss cantons border Germany?* is decomposed into subquestions like *Name a canton in Switzerland*, which yields subanswers such as Aargau, Basel and so on, and for each of these a revised question is generated, for example *Does Aargau border Germany?*

RENOIR is an interactive tool where query procedures are executed, generating partial and final results for each GikiP topic (Santos et al., 2008, 2009). RENOIR makes extensive use of REMBRANDT (Cardoso, 2008), a named entity recognition module for Portuguese and English which uses nine main categories: person, organisation, local, timestamp, value, abstraction, thing, masterpiece and event. For each topic the user creates a sequence of query procedures. These processing actions can be executed automatically, semi-automatically or manually by the user. There are four types of actions employed: automatic **retrieval actions**, e.g. find articles mentioning a *term*, pertaining to a given *category* or linking to a given *article*; semi-automatic **mapping actions**, i.e., matching named entities to Wikipedia articles; automatic **NER actions**, to process documents text with REMBRANDT, and **filtering actions**, e.g. automatically checking for a NER category or manually validating if a candidate article is a correct answer. RENOIR is conceptually very similar to EQUAL from the point of view of these high level actions, however it relies entirely on the user creating the adequate processing sequence, whereas EQUAL interprets the question and finds the answers automatically. Both approaches tend to over-simplify the evidence needed to validate answers, e.g., in topic GP11 *Which plays of Shakespeare take place in an Italian setting?* where a reference to a place in Italy is enough for the system to consider that a given Shakespearean play actually happens in Italy, although this is not a sufficient condition.

We have carried out an analysis of the top three submissions using the  $F_1$  score. Table 4.3 reports Precision **P**, pseudo-Recall **RR** and  $F_1$ -measure computed using the set of known

Table 4.4: GikiP results for EQUAL (c11)

| Topic     | Answers | Correct | Precision | Languages | Score | $K_i$ |
|-----------|---------|---------|-----------|-----------|-------|-------|
| GP10      | 0       | 0       | 0         | 3         | 0     | 2     |
| GP9       | 0       | 0       | 0         | 3         | 0     | 15    |
| GP5       | 0       | 0       | 0         | 3         | 0     | 19    |
| GP15      | 10      | 2       | 0.20      | 3         | 1.20  | 2     |
| GP8       | 8       | 2       | 0.25      | 3         | 1.50  | 2     |
| GP1       | 1       | 1       | 1.00      | 3         | 3.00  | 1     |
| GP2       | 5       | 3       | 0.60      | 3         | 5.40  | 7     |
| GP13      | 2       | 2       | 1.00      | 3         | 6.00  | 4     |
| GP3       | 3       | 3       | 1.00      | 3         | 9.00  | 8     |
| GP14      | 3       | 3       | 1.00      | 3         | 9.00  | 6     |
| GP6       | 6       | 6       | 1.00      | 3         | 18.00 | 7     |
| GP4       | 24      | 15      | 0.63      | 3         | 28.13 | 19    |
| GP11      | 25      | 21      | 0.84      | 3         | 52.92 | 24    |
| GP7       | 18      | 18      | 1.00      | 3         | 54.00 | 33    |
| GP12      | 18      | 18      | 1.00      | 3         | 54.00 | 25    |
| total c11 | 123     | 94      | 0.63      | 3         | 16.14 | 174   |

correct answers (both micro- and macro-averages). These measures show that the semi-automatic approach employed by c1-RENOIR (Santos and Cardoso, 2008) is also competitive, achieving better recall but lower precision. The precision drop between micro and macro averaging is due to the three topics for which EQUAL does not return any answer. RENOIR demonstrates more stability on the two measures. Interestingly, from a total of 174 answers only 40 are found by both systems, suggesting that the two approaches are complementary.

### 4.5.3 GikiCLEF results

EQUAL achieved the best performance also for GikiCLEF amongst 17 runs by 8 participants (see Table 4.5). The second highest ranked system, GREASE/XLDB (Cardoso et al., 2009) is a semi-automatic submission consisting of hand crafted SPARQL queries on the DBpedia dataset. The third submission, CHESIRE (Larson, 2009), consists of documents retrieved by sending manually built queries to an advanced geographic textual retrieval engine. The system ranked fourth, GIRSA-WP (Hartrumpf and Leveling, 2010), uses a recursive question decomposition approach which combines results from a

Table 4.5: GikiCLEF results for all languages

| #  | System       | Answers | Correct    | Precision   | Score         |
|----|--------------|---------|------------|-------------|---------------|
| 1  | EQUAL        | 813     | <b>385</b> | 0.47        | <b>181.93</b> |
| 2  | GREASE/XLDB  | 1161    | 332        | 0.29        | 96.01         |
| 3  | Cheshire     | 564     | 211        | 0.37        | 80.92         |
| 4  | GIRSA-WP_1   | 38      | 31         | <b>0.82</b> | 24.76         |
| 5  | GIRSA-WP_2   | 985     | 142        | 0.14        | 23.39         |
| 6  | GIRSA-WP_3   | 994     | 107        | 0.11        | 14.52         |
| 7  | JoostER_1    | 638     | 36         | 0.06        | 2.41          |
| 8  | GikiTALP_3   | 356     | 26         | 0.07        | 1.90          |
| 9  | GikiTALP_2   | 295     | 20         | 0.07        | 1.36          |
| 10 | GikiTALP_1   | 526     | 18         | 0.03        | 0.70          |
| 11 | bbk_ufrgs_1  | 726     | 8          | 0.01        | 0.09          |
| 12 | UAICGIKI09_1 | 6420    | 8          | 0.00        | 0.02          |
| 13 | bbk_ufrgs_2  | 734     | 3          | 0.00        | 0.01          |
| 14 | UAICGIKI09_2 | 1133    | 2          | 0.00        | 0.01          |
| 15 | JoostER_2    | 272     | 0          | 0.00        | 0.00          |
| 16 | bbk_ufrgs_3  | 686     | 0          | 0.00        | 0.00          |
| 17 | UAICGIKI09_3 | 4910    | 0          | 0.00        | 0.00          |

geographic information retrieval system with those of a semantic question answering system. This fully automatic system achieves a good performance, but at lower levels of precision and recall. The other systems are essentially traditional textual QA systems which were more-or-less adapted for this task. Their performance shows that the traditional architecture is ill suited to address complex questions.

To deal with all ten languages of the GikiCLEF competition, EQUAL processed English due to the better coverage it affords (Santos and Cabral, 2009b). The human assessors classified the answers returned by EQUAL as follows: 69 correct, 10 unjustified and 59 incorrect, yielding a precision of 50% and a score of 34.5 (see Table 4.6). By mapping these answers from English to the other nine languages using the inter-wiki links, the cumulative results is 385 correct out of a total of 813 answers: precision 47%, score 181.93 (see Table 4.5).

GREASE/XLDB is a prototype system exploring the use of Semantic Web technologies for answering GikiCLEF topics. The system uses three knowledge sources: the DBpedia v3.2 dataset (Auer et al., 2008); WikiWGO, a geographic ontology combining Wikipedia with

a geographic knowledge base (Chaves et al., 2005), and articles from the Portuguese Wikipedia which were processed by HENDRIX, a newly developed named-entity recognition module based on Conditional Random Fields, which was trained to recognise four categories: places, organisations, events and people. These entities are automatically matched to concepts of the WikiWGO ontology. The system relies on the user to manually create SPARQL queries corresponding to GikiCLEF topics. For example, for the training topic *Which romanian writers were born in Bucharest?* the following query was created:

```
SELECT ?RomanianWriters WHERE {  
  ?RomanianWriters skos:subject <http://dbpedia.org/resource/Category:Romanian_writers> .  
  ?RomanianWriters dbpedia-owl:birthplace <http://dbpedia.org/resource/Bucharest>  
}
```

Because these queries were manually created, the results are not directly comparable to those of automatic submissions, but this study provides interesting insights regarding the coverage of the DBpedia dataset for GikiCLEF topics. For example, English Wikipedia infoboxes were only useful in finding answers for 9 out of 45 topics because in most of the cases the relevant information is found in the text of the article and not in the structured dataset; similarly, in the Portuguese language version, answers were only found for 6 topics. The authors also showed that multilingual capabilities are important for achieving robust performance: the Portuguese language version of Wikipedia has sufficient information to validate only 27.6% of all the answers found by the GikiCLEF participants. This type of cultural aspects relevant to GikiCLEF are further discussed in Santos et al. (2010).

Cheshire (Larson, 2009) is an interactive approach which employed a state-of-the-art cross-lingual information retrieval engine (Chen and Gey, 2004) using a version of the Logistic Regression (LR) algorithm (Cooper et al., 1992). The title, body and anchor (a) tags were extracted from the HTML documents when indexing the collections. When executing queries, the system employs a blind relevance feedback algorithm based on the probabilistic term relevance weighting formula developed by Robertson and Sparck Jones (1988). The algorithm typically involves two stages. First, an initial search using the

original topic statement is performed, after which a the top 10 terms with the highest relevance weights are selected from the 10 top-ranked documents. The selected terms are then weighted and merged with the initial query to formulate a new, expanded query which is submitted against the same collection to produce a final ranked list of documents.

The queries were formulated by a human user interactively, a task which “involved a lot of time spent reading pages and deciding whether or not the page was relevant.” Because “each question took literally hours of work using the interactive system,” only 22 out of the 50 topics were addressed before the submission deadline (Larson, 2009). The results of this submission illustrate that a good IR engine can be used to search for the correct answers but only if the user is willing to spend time to repeatedly refine the queries.

GIRSA-WP (Hartrumpf and Leveling, 2009) is an improved version of the fully automatic system which participated in GikiP (Santos et al., 2009). It uses a recursive question decomposition approach which generates a sequence of simple intermediate questions (list, factoid and yes/no) which are answered by combining a QA system with an IR engine. For example, for topic **GC09-07** *What capitals of Dutch provinces received their town privileges before the fourteenth century?*, GIRSA-WP generates the following sequence of questions:

- *Name capitals of Dutch provinces.* This is recursively decomposed into:
- *Name Dutch provinces.* This question yields a list of entities such as *Zeeland*, which are used replace the phrase ‘Dutch provinces’ in question a) generating new questions such as:
- *Name capitals of Zeeland.* This factoid questions yield one answer each, in this case Middelburg, which are used to rephrase the original topic:
- *Did Middelburg receive its town privileges before the fourteenth century?*

Using this approach allows GIRSA-WP to gather the supporting evidence for each answer it produces. The system submitted three runs: run 1 only used the results from the InSicht

QA system, while runs 2 and 3 combined these with the results returned by the GIRSA IR engine using different weighting parameters. The intermediate questions generated by GIRSA-WP test one atomic fact or a relation, which resembles the semantic constraint verifiers, which suggests that EQUAL could also generate intermediate questions and use a robust factoid QA system to verify constraints.

JoostER (Bouma and Duarte, 2009) is an automatic textual QA system for Dutch and Spanish which adopts some of the ideas proposed by EQUAL. JoostER uses a linguistic analysis consisting of shallow parsing the topic to determine the EAT and to extract the additional constraints. These are used to build a query which results in a set of relevant documents. The system incorporates additional knowledge sources (WordNet, Yago, DBpedia) to improve the mapping of the EAT to the most relevant Wikipedia categories, which enables the system to filter out documents which do not match the EAT. Instead of only using the EAT phrase, JoostER uses query expansion to increase recall. One way is to include additional hypernyms from DBpedia, e.g., in a topic about *German artists* it adds types such as *German writers*, *German comedians* or *German actors*. Another way is to include synonyms extracted from WordNet, from the redirect links and from a list of country demonyms. An IR engine retrieves documents which are relevant to the expanded query, and these are filtered based on their type. To do this for a Spanish article, the system uses the cross-lingual links to the corresponding English page, retrieve its types from the Yago/DBpedia dataset and match them to the types extracted from the topic. The Spanish system was implemented after GikiCLEF and its performance would have ranked it 4<sup>th</sup> which is the best performance reported by an automatic textual QA system.

The remainder of submissions<sup>3</sup> are simple textual QA systems which automatically convert the GikiCLEF topics into keyword queries and consider the top-ranked documents as answers. These systems usually achieved poor results because they focus on retrieving

---

<sup>3</sup>these systems are described in more detail at [http://clef.isti.cnr.it/2009/working\\_notes/CLEF2009WN-Contents.html](http://clef.isti.cnr.it/2009/working_notes/CLEF2009WN-Contents.html)

relevant documents, rather than correct entities. GikiTALP is a simple system based on a full-text information retrieval engine, Sphinx<sup>4</sup>, which does not use explicit geographical knowledge. It addressed two of the ten languages, English and Spanish. The topic texts were considered as keyword queries and minimal NLP was employed, such as lemmatisation and stop-word removal. BBK-UFRGS is similar system for Portuguese which uses GATE<sup>5</sup> and a list of geographical place names<sup>6</sup> to analyse topics. The weight of any geographic entities mentioned in the topic was boosted when building keyword queries. The titles of the most relevant articles returned by an IR engine, Zettair<sup>7</sup>, were considered answers. Since the list was not filtering based on the semantic type specified by the topic, the results were modest.

UAIC uses a textual QA approach which maps the EAT to person, location, organisation or date. Instead of retrieving paragraphs to extract and validate answers, it uses document-level retrieval. The nouns, verbs and named entities in the topic are used to generate a Lucene<sup>8</sup> query, e.g., for the topic *List the Italian places where Ernest Hemingway visited during his life*, it builds the query (places^2 place) +Italian +Ernest +Hemingway (visited^2 visit) during life (title:Italian title:Ernest title:Hemingway title:Italian Ernest Hemingway) and marks the EAT as location. The relevance score of the relevant articles is changed to reflect whether or not the title mentions an entities matching the EAT. Because very broad entity categories are used, and because GikiCLEF requires the title of the article to be a named entity of the specific type sought by the topic, the UAIC system which processed Romanian and Spanish collections had very low precision.

EQUAL ranked first in GikiCLEF, ahead of semi-automatic, automatic and standard textual QA approaches (Santos and Cabral, 2009a), proving that questions previously considered too difficult to answer can be successfully addressed using Wikipedia as

---

<sup>4</sup><http://www.sphinxsearch.com/>

<sup>5</sup><http://gate.ac.uk/>

<sup>6</sup><http://snowball.tartarus.org/>

<sup>7</sup>[www.seg.rmit.edu.au/zettair/](http://www.seg.rmit.edu.au/zettair/)

<sup>8</sup>

Table 4.6: GikiCLEF results for English

| #  | System       | Answers | Correct   | Precision   | Score        |
|----|--------------|---------|-----------|-------------|--------------|
| 1  | EQUAL        | 138     | <b>69</b> | 0.50        | <b>34.50</b> |
| 2  | Cheshire     | 139     | 56        | 0.40        | 22.56        |
| 3  | GREASE/XLDB  | 198     | 52        | 0.26        | 13.66        |
| 4  | GIRSA-WP_1   | 5       | 3         | <b>0.60</b> | 1.80         |
| 5  | GikiTALP_3   | 296     | 22        | 0.07        | 1.64         |
| 6  | JoostER      | 136     | 14        | 0.10        | 1.44         |
| 7  | GIRSA-WP_3   | 141     | 14        | 0.10        | 1.39         |
| 8  | GikiTALP_2   | 295     | 20        | 0.07        | 1.36         |
| 9  | GIRSA-WP_2   | 129     | 11        | 0.09        | 0.94         |
| 10 | GikiTALP_1   | 383     | 16        | 0.04        | 0.67         |
| 11 | UAICGIKI09_2 | 642     | 1         | 0.00        | 0.00         |
| 12 | UAICGIKI09_3 | 491     | 0         | 0.00        | 0.00         |

a repository of world knowledge. As well as precision, the system also had the best performance in terms of the number of correct answers found. However, the results reflect the bias towards precision: the system did not return any answers for 25 (half) of the questions. Another interesting observation is that the systems ranked 2<sup>nd</sup> and 3<sup>rd</sup> are semi-automatic systems: a user is responsible with manually creating queries and checking the results in an iterative fashion. Despite being an automatic system, EQUAL outperforms both of them achieving higher precision and higher recall.

In spite of the positive results, the prototype needs further improvements to be able to answer more questions and to increase its recall. Although it uses more expressive semantic constraints to its predecessor, EQUAL demonstrated a decrease in performance for this task, reflecting the fact that the GikiCLEF topics were more difficult than the ones used in GikiP.

In GikiCLEF, both micro and macro averaging are needed because of the way answers are distributed over topics. Micro averaging gives each answer the same weight, favouring systems that solve the topics with most answers. For example the top 10 topics with most answers account for 50% of the entire answer pool. The most prolific topic has as many answers as the 17 least prolific topics combined. Because of this distribution, a system which answers a prolific topic by chance can score much higher than a system which

Table 4.7: GikiCLEF normalised results for English

| #<br>System       | micro-averaging |       |       | macro-averaging |       |       |
|-------------------|-----------------|-------|-------|-----------------|-------|-------|
|                   | P               | RR    | F1    | P               | RR    | F1    |
| <i>EQUAL (25)</i> | 0.790           | 0.800 | 0.795 | 0.570           | 0.670 | 0.616 |
| <i>EQUAL (50)</i> | 0.395           | 0.400 | 0.397 | 0.570           | 0.410 | 0.477 |

answers perfectly many, but less prolific topics. Macro averaging gives each topic the same weight, favouring the systems which answer topics with fewer answers.

Table 4.7 shows the results without considering whether correct answers were justified or not. Recall (RR) is computed relative to the entire set of English answers assessed as correct by human judges, called the English answer pool. For the set of answered questions, performance is satisfactory despite most of EQUAL’s modules being relatively simple. While the prototype clearly outperformed standard textual QA systems (Santos and Cabral, 2009a), the competition uncovered some limitations of the current implementation. The fact that it did not return any answers for half of the questions suggests that its components have a limited coverage and that the system is not robust: more verifiers are necessary, as is ability to combine answers found by similar interpretations.

## 4.6 Discussion

EQUAL implements a simplified version of the architecture proposed in Chapter 3. The most important simplification is due to the non-interactive nature of the task. Lacking the feedback stage, EQUAL generates alternative question interpretations and addresses them sequentially, until one yields results. The underlying hypothesis is that only a correct interpretation of the question can have results. This bias also reflects the official scoring measure which is based on precision. This section discusses some issues arising from this implementation, identifying solutions for future integration.

The constraint verifiers employed are generic, in order to avoid over-fitting the sample topics. Each has a method to be instantiated using a chunk from the question and a

method to test a candidate solution. The implementations are quite general and achieve a good balance between accuracy and coverage, but more specialised implementations will lead to better performance. Currently, at least one of the verifiers must match for a constraint to hold. This is a conservative behaviour favouring precision, given that the training set was relatively small and only few verifiers were implemented. A future challenge is to automatically learn verifiers from either seed examples or from user feedback.

Analysis of the results revealed some of the limitations of the current verifiers. For example, in **GC09-09** *Name places where Goethe fell in love*, the system cannot distinguish between the places where Goethe **lived**, those he **visited** or those where he **fell in love**. The relation constraint verifier looks at the positions of the trigger words (*fell, love*) in relation to the link to a candidate answer, but is not a specialised extractor for this particular relation. Instead of the boolean model currently employed, EQUAL needs to adopt a fuzzy model to estimate the likelihood that an entity satisfies a constraint, enabling several partial clues to jointly contribute towards validating a constraint. Ranking its confidence means that EQUAL can provide more useful results to its users and improve its answers, e.g., using an on-line learning algorithm exploiting user feedback. It is arguably more useful for a user to see an answer with a low confidence score and marked as such, instead of a blank screen. One of the future research directions is to use a probabilistic model and perform belief propagation to rank answers and interpretations.

A drawback of the current implementation is the computational effort needed to test all the verifiers at query time, especially those analysing the entire text of an article. Answering questions with a large number of candidate articles was aborted if no result was identified within a given time limit. To reduce runtime complexity, it is necessary to optimise the order of evaluating verifiers and to enable the system to first use ‘pre-verifiers’, representing necessary conditions that can be tested efficiently. Such information can be processed off-line and combined with other structured data, such as DBpedia or Freebase. The time-consuming verifiers should be applied to each candidate in a second step, which can be easily parallelised to further reduce the answering time.

EQUAL currently explores direct evidence (information mentioned in the article corresponding to the entity). To enable indirect evidence, EQUAL needs a reliable disambiguation method to recover links which are not present in articles due to Wikipedia style guidelines. The next chapter studies existing tools and describes a new method for enriching the Wikipedia link graph. This method can be also applied to external text: for example large newswire corpora can be automatically linked to Wikipedia and used by textual verifiers that search for indirect evidence. This can be performed at indexing time, to support fast retrieval of relevant information.

The primary cause for imprecise answers is due to the fact that EQUAL uses few semantic constraints: it lacks the expressive power required to accurately represent all the questions. Sometimes, the generated interpretation misses relevant information from the question. For example, in **GC09-11** *What Belgians won the Ronde van Vlaanderen exactly twice?*, EQUAL returns **all** the Belgian winners, because it cannot ‘understand’ the constraint *exactly twice*. One feature that needs to be added to EQUAL is support for aggregates and quantifiers. Universal quantifiers have significant implications regarding tractability. In terms of verifying aggregates, EQUAL will need to combine textual extractors, i.e., finding an explicit textual mention such as ‘they had five children’, with set aggregates, i.e., finding a list or a table mentioning the children or determining articles corresponding to each child and then derive the cardinality of this set.

An important factor impacting the correctness of results is *vagueness*. For example “Romanian writers” can mean writers of Romanian nationality, but also Romanian-language writers of different nationality, such as Moldovan writers, as well as writers of Romanian descent that only wrote in other languages, e.g., German. By using information from Wikipedia, the results provided by the system will reflect this vagueness, therefore answers’ correctness should be considered in a lenient fashion: systems cannot be expected to outperform their own sources. For example, for topic **GC09-19** *Name mountains in Chile with permanent snow*, the system only found one of the total of 14 answers judged correct, because its verifiers looked for an explicit reference to permanent

snow. It is sometimes debatable what kind of proof should be accepted as valid; for certain topics the judges had difficulties in reaching consensus (Santos and Cabral, 2009a), suggesting that the task is also difficult for humans, and that a lenient correctness criterion is necessary in an open-domain task. If EQUAL provides confidence scores and supporting evidence, the user will find its answers more informative.

The performance is also affected by inconsistencies in Wikipedia. EQUAL assumes that all the articles are assigned to correct and relevant categories, but this is not always the case. Inaccurate categories decrease precision. For example, in **GC09-18** *In which Tuscan provinces is Chianti produced?*, 13 pages were inaccurately assigned to the category *Provinces of Tuscany* at the time of the competition, when in fact they are places in Tuscany, and only 3 pages described actual provinces. Incomplete categories lead to a decrease in recall and the system should consider combining several sources of information to mitigate this. Although information in Wikipedia is continuously improved and updated, such inconsistencies are inherent in a project this size (Hammwöhner, 2007).

Mapping the EAT to categories needs further refinement, as the current mechanism assumes that there is a most relevant category and that the terms in the question are similar to Wikipedia’s folksonomy. This is not always the case, usually because such a category does not exist, e.g., *German-speaking movies* and *Swiss casting show winners*. In **GC09-15** *List the basic elements of the cassata*, the question is asking for the **ingredients** of the Italian dessert (sponge cake, ricotta cheese, candied peel, marzipan, candied fruit, and so on). Even though the system finds mentions of these articles, it discards them because they are not ‘elements’. As well as the confidence of a category mapping, the system should also estimate the likelihood that a ‘correct’ category does not in fact exist.

A bad category mapping is the primary cause for not answering questions: either the mapping is too generic, yielding too many results, or it is incorrect and no results are found. If the number of answers EQUAL finds exceeds the maximum threshold of 30 answers, the entire result-set was dismissed at the expense of recall.

Currently, EQUAL first picks a starting category and then visits its articles. To increase recall, another verifier needs to be added which uses a keyword based retrieval using all the semi-structured data available (such as categories, navigation boxes, infobox contents, list pages, disambiguating terms) to create an alternative, high recall set of candidate answers, and then employs a generic classifier predicting the likelihood that an individual article has the type specified in the question. Such a verifier enables information from more than one category to be used.

## 4.7 Conclusions

This chapter presented EQUAL, an encyclopaedic QA system which implements the analysis phase of the architecture proposed in this thesis to answer complex open-list questions against Wikipedia. It detects different sources of ambiguity creates multiple question interpretations, corresponding to different understandings of the question. The question interpretation consists of a decomposition of the question into constituents which are then assigned to coarse-grained semantic constraints. Instead of retrieving paragraphs, EQUAL explores the Wikipedia page graph to determine which entities are correct answers for a particular interpretation. To enforce its constraints, EQUAL can employ structured, semi-structured and textual resources.

Section 4.5 presented the results achieved in two competitions, where EQUAL significantly outperformed all automatic submissions and compared favourably with semi-automatic approaches. The error analysis carried out revealed some of the challenges facing this architecture. One of the limitations is the reliance on the existing wiki links when creating the Wikipedia link graph. The next chapter will investigate the problem of enriching the Wikipedia markup with additional links to improve the recall of the system.

# Chapter 5

## Semantic Document Analysis

### 5.1 Overview

As a repository of encyclopaedic entities, Wikipedia is a valuable resource for QA systems due to the wide variety of topics covered as well as the semi-structured nature of the data. It was argued in Section 3.3 that plain text alone is too ambiguous for QA applications because standard information retrieval engines yield imprecise data. One way to improve retrieval accuracy is to distinguish distinct entities which share the same name (homographs). Existing wiki links can help alleviate ambiguity, but even for human readers these links are usually incomplete as the editorial guidelines<sup>1</sup> recommend only linking to the most relevant articles, to avoid clutter and redundant information. As a result, Wikipedia markup is not complete, especially from the point of view of automatic processing by computers. As demonstrated by EQUAL in the previous chapter, searching using the existing wiki links yields precise answers but has a negative impact on recall. To increase recall, it is therefore necessary to enrich the existing markup with new, automatically detected links.

This chapter studies the problem of identifying and ranking Wikipedia entities mentioned in a text, a task related to semantic analysis, word sense disambiguation and named entity disambiguation. More specifically, given a textual document  $d$  and an encyclopaedic knowledge base comprising a set of entities  $E$ , the task is to identify and disambiguate

---

<sup>1</sup>[http://en.wikipedia.org/wiki/Wikipedia:Manual\\_of\\_Style](http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style)

textual references  $ref_i \in d$ , linking them to entries in the knowledge base. The assumption is that the document  $d$  already contains a set  $V_0$  of disambiguated links. The name **densification** is used to refer to this task. The links identified and ranked by a densification system allow an arbitrary NLP application to then select those targets that are relevant for the particular task at hand. Densification can be used as a general purpose iterative semantic analysis tool: in each step  $k$  a possibly empty set of validated targets  $V_k$  is used to re-rank the remaining set  $C_k$  of candidates. The task is related to wikification (Mihalcea and Csomai, 2007; Cucerzan, 2007; Milne and Witten, 2008b), which is discussed in more detail in Section 5.2.

Question answering is not the only application which can benefit from this type of semantic analysis. It can also be employed in other NLP applications such as information extraction, text mining, coreference resolution and automatic summarisation. This type of analysis blurs the boundaries between the textual Web documents and Semantic Web meta-data. The analysis can be performed off-line during indexing/pre-processing, when text can be represented in a semantic space induced by Wikipedia articles. Coordinates of a semantic vector represented in this space correspond to Wikipedia article identifiers, weighted according to the disambiguation confidence score and the number of occurrences. Computing the weighted vector is a form of explicit semantic indexing which enables efficient, higher-level semantic retrieval procedures, such as defining queries to retrieve *⟨paragraphs mentioning Manchester United (the association football team) and also mentioning a stadium⟩*. This semantic representation is an alternative to current approaches for semantic indexing, such as latent semantic indexing, probabilistic latent semantic indexing, random indexing or latent Dirichlet allocation, which construct semantic spaces based on word co-occurrence patterns in textual documents. Such methods achieve state-of-the-art performance in modern information retrieval engines, however, these models yield abstract conceptual spaces that are difficult to interpret. Using Wikipedia articles as concepts does not have this limitation. Instead of using mathematical criteria to reduce the dimensionality of the space, the categories in Wikipedia can be used to create a second-order vector space model which is easier to

interpret. The links between articles allow the use of a generalised vector space model, i.e., one in which the coordinates are not orthogonal.

This chapter is structured as follows: Section 5.2 reviews previous methods for disambiguating entity mentions and linking arbitrary text to Wikipedia. Most of them aim for balance between precision and recall. To increase the number of links identified and to also include less prominent entities, a novel task, densification, is proposed in Section 5.3. A two-step method for addressing this task is described in Section 5.4, and Section 5.5 describes three evaluation experiments. In the first, human users are employed to evaluate a sample of automatically predicted links. This experiment investigates the feasibility of developing a large scale gold standard annotation. In the second experiment, a set of Wikipedia articles are used to automatically estimate recall. The proposed method is compared to some of the tools developed by other researchers which are described in Section 5.2. The final experiment investigates the impact of the densification system on question answering, i.e., whether it improves the recall achieved by the EQUAL system. The chapter finishes with conclusions.

## 5.2 Related work

Amongst similar forms of semantic analysis which have been proposed in the past, the most relevant approach is Explicit Semantic Analysis (ESA) (Gabrilovich and Markovitch, 2007). In ESA, the text of Wikipedia articles is used to create an inverted index: for each word  $w$ , the ranked list of documents it appears in is built using *tf-idf* weighting:  $docs(w) = \{(d_w^i, r_i) | i = \overline{1..n}\}$ . Given an input text fragment, the system combines the vectors associated with each of its words, yielding an aggregated vector in the semantic space. The coordinates of this space correspond to Wikipedia articles. The authors demonstrate that this space can be used to measure semantic similarity between two texts, even if their lexical overlap is low. ESA is similar to latent semantic indexing approaches (such as latent semantic analysis) in that each word  $w$  has an associated distribution over topics  $P(topic_i | w)$ . The difference is that the components of the ESA

semantic space are explicitly determined by a subset of Wikipedia articles, whereas the abstract concepts in the latent spaces cannot be easily interpreted, making them less attractive to QA.

Identifying mentions of Wikipedia concepts in texts, called wikification, is an active research area made popular by Wikify! (Mihalcea and Csomai, 2007) and WikipediaMiner (Milne and Witten, 2008b). The potential of Wikipedia as a entity knowledge base was studied by Bunescu and Pasca (2006) and Cucerzan (2007). In addition, more recent tools such as *DBpedia Spotlight* (Mendes et al., 2011), as well as commercial systems such as OpenCalais<sup>2</sup> and AlchemyAPI<sup>3</sup> have also been developed. Wikification is a special case of analysis meant to replicate the current linking style of Wikipedia articles, when only the most important entities are sought.

There are three main challenges for this type of entity disambiguation task: determining possible mentions in the input text, disambiguating for each mention which Wikipedia article is the best target, and selecting a set of links which have a high joint confidence score. The last step is necessary because Wikipedia is incomplete and the correct sense might not be present. In the news collection used in the Entity Linking task at the Text Analysis Conference<sup>4</sup>, 57% of named entity mentions did not have a corresponding Wikipedia article. Depending on the nature of the relatedness measure used to rank the targets, there are two main wikification approaches. The first uses a semantic relatedness measure based on the Wikipedia link graph, while the second uses word-based similarity between text snippets.

WikipediaMiner (Milne and Witten, 2008b) employs a two step pipeline. It considers as a possible mention any word n-gram which has been used as the anchor text of at least one link in Wikipedia. Those mentions which have exactly one possible target make the set of unambiguous targets. This set is used as a reference context for the rest of

---

<sup>2</sup><http://www.opencalais.com>

<sup>3</sup><http://www.alchemyapi.com/>

<sup>4</sup><http://www.nist.gov/tac/about/index.html>

the ambiguous mentions to determine the relatedness of each possible target. Using this relatedness score, the most likely target article is selected for each mention. In the second step, the system decides which of the targets deemed plausible should be linked based on their relatedness to each other as well as to the reference context. A decision tree model is used to predict the confidence of each link, and only the targets exceeding a fixed threshold are selected. The main contribution of WikipediaMiner is the semantic relatedness measure employed, which is based on the Jaccard index of the two articles' sets of neighbours. The system achieves 97% accuracy in the first step (determining the most valid target of an existing link), exceeding the most frequent target baseline of 90%. While promising, this result is measured only for existing links — those previously marked by editors. When dealing with new texts, the system needs to also determine the mentions themselves, which increases the complexity of the decision space with a negative impact on disambiguation accuracy. The selection criterion in the second step is biased towards prominent entities, favouring a nucleus of high confidence and unambiguous targets. The more challenging the disambiguation, the less likely it is to rank high enough to pass the selection threshold. This is an effective way to eliminate errors made during the disambiguation step, but only because recall has less importance for wikification evaluation. To further enrich the interlinking in Wikipedia it is necessary to employ more advanced measures which are able to also link less prominent articles.

Bunescu and Pasca (2006) show that using the cosine similarity between a mention context and the text of each possible target article yields average results (55%). Applying a Support Vector Machine (SVM) classifier with a taxonomy kernel that exploits the correlations between words and Wikipedia categories, e.g., *conducted* has a stronger association with *Category:Composers* than it has with *Category:Wrestlers*. This leads to a 20% error reduction. Performance is less impressive than more recent models, but results are not directly comparable, especially as the most frequent target baseline is not reported. The main limitation of this work is that it only addresses people mentions which represent a small variation of the title of the corresponding article.

Cucerzan (2007) proposes a large scale disambiguation method based on Wikipedia

data which can only be used for named entities, where mentions are identified by an existing named entity recogniser. A vector representation for each article is used to disambiguate between the candidate entities. The coordinates of this context vector are the wiki links collected from the first paragraph of an article together with its categories. Disambiguation is seen as an assignment problem: the best solution is found by maximising the agreement between the context vectors extracted from Wikipedia and the context vector extracted from the document. When applied to Wikipedia articles, the system achieves 88.3% accuracy, 2 points above the most frequent target baseline. This is due to the fact that the majority of entity mentions are not ambiguous, i.e., they only have one possible target. When applied to a newswire corpus, the system achieves 91.4% accuracy, versus a 51.7% baseline.

Mendes et al. (2011) describe DBpedia Spotlight, a supervised memory-based learning approach for disambiguating entities. A dictionary of references is built from the entire contents of Wikipedia: for each article an associated meta-document is built by concatenating the textual snippets surrounding each of its in-links. These meta-documents are indexed by an IR engine. To disambiguate plain text, a query is built for each plausible n-gram and sent to the IR engine, and the highest ranked article retrieved is selected as the best link. In essence, the disambiguated target is the nearest neighbour in a vector space representation using cosine similarity between *tf-idf* weighted word-vectors. To evaluate the disambiguation performance, 155,000 wiki links with ambiguous surface form are set aside for testing, while around 69 million links are used to build the system. A random target baseline only achieves 17.77% accuracy, while the most frequent target baseline achieves 55.12%. The best results were achieved by a regression model which combines the target frequency observed in Wikipedia, i.e., the prior probability of a target, with the cosine similarity score: 80.52%.

thewikimachine<sup>5</sup> uses more than 1 million kernel-based classifier models, each trained to disambiguate an individual word (Bryl et al., 2010). Training data for this supervised

---

<sup>5</sup><http://thewikimachine.fbk.eu/>

WSD task is derived from the textual snippets surrounding wiki links. To allow the use of a domain kernel exploiting linguistic knowledge, the system uses a mapping of Wikipedia articles to WordNet synsets. The performance of the system is evaluated on the ACE05-WIKI Extension (Bentivogli et al., 2010), a dataset which extends the English Automatic Content Extraction<sup>6</sup> (ACE) 2005 dataset with ground-truth links to Wikipedia. The 599 documents in this collection contain a total of 29,300 entity mentions manually linked to the Wikipedia articles they refer to. The system achieves an F1 score of 0.715, compared to 0.587 obtained by WikipediaMiner on the same dataset.

Commercial systems such as OpenCalais<sup>7</sup>, Zemanta API<sup>8</sup> or Alchemy API<sup>9</sup> also allow the linking of arbitrary text to Wikipedia topics, for example for automatic semantic tagging of newswire articles or blog posts. Such systems are primarily focused on generic named entities (such as person, organisation, location, and others), and the most common target is usually used as a disambiguation method. Because of this, commercial systems are ill suited as an off-the-shelf solution for densification and are not particularly relevant for this research. However, they can provide candidate links for plain text documents, which could be used as input for densification.

### 5.3 Densification: task definition

The aim of densification is to identify articles from Wikipedia which are relevant to a snippet of text. Such an article, called *topic* or *target*, is helpful for both human and machine readers. It is usually referred to by a small span of text called *mention*, *anchor* or *keyword*. The following example illustrates some difficulties in automatically identifying mention boundaries in text: in the sentence “*The US President Barack Obama sent an official letter of apology.*” the entire “*The US President Barack Obama*” span refers to the person, but both “*The US President*” and “*Barack Obama*” are viable anchors with the

---

<sup>6</sup><http://www.itl.nist.gov/iad/mig//tests/ace/ace05/index.html>

<sup>7</sup><http://www.opencalais.com/>

<sup>8</sup><http://www.zemanta.com/api/>

<sup>9</sup><http://www.alchemyapi.com/>

same target. The anchor “*the US President*” can also have as target the page *President of the United States* which describes the office, not the incumbent. Other pairs of anchor text and target article which contain relevant information could be, e.g.,  $\langle \textit{president}, \textit{President} \rangle$ ,  $\langle \textit{US}, \textit{United States} \rangle$ ,  $\langle \textit{Obama}, \textit{Obama (surname)} \rangle$  and even  $\langle \textit{The}, \textit{English articles} \rangle$ , even if these are not coreferent in a strict sense.

As illustrated above, the problem is quite complex because a word can be used in hundreds, even thousands of distinct anchors, each linking to tens of distinct pages. In addition, due to its broad coverage, many entities in Wikipedia are homographs (spelled the same), which means that in a text, mentions with an obvious meaning to a human reader can be very ambiguous for computers. For example the Wikipedia page *The (disambiguation)* lists four exact homographs, while most NLP tools typically consider ‘the’ to be the English definite article and discard it when analysing text; the article *President (disambiguation)* lists more than 25 pages which are possible candidate targets, while in WordNet, *president* has only 6 synsets.

On the one hand, densification is more difficult than word sense disambiguation due to this increased ambiguity, but also because there are vastly more entity names than English common nouns. On the other hand, the information contained in a Wikipedia article is substantial compared to a WordNet synset gloss, or to a dictionary definition. This information can be used to re-rank the likelihood of a target page by measuring relatedness to other targets either locally (in the same paragraph), or globally (in the entire document), thereby making the task less daunting.

In a way, densification is similar to wikification, as both tasks are a form of disambiguating entity names by linking them to their corresponding Wikipedia articles. However, there are two main differences between the two tasks. Firstly, wikification aims for strict disambiguation while densification is designed to provide links to relevant Wikipedia pages, enabling other tools to further filter these links to suit their needs. Secondly, densification employs existing links as seed meta-data in the disambiguation context; this enables iterative annotation and integration with other tools, such as named entity disambiguation models.

## 5.4 Densification: approach

Given a textual document  $d$  and a set of  $n$  seed topics  $S = \{s_i | i = 1 \dots n\}$ , the task is to find a set of relevant topics  $T = \{t_j\}$  that are mentioned or referred to in the document. The system must identify candidate topics and rank them by importance, integrating local information extracted from snippets of  $d$  with global information extracted from Wikipedia. The system collects  $\langle \text{anchor text}, \text{target page} \rangle$  pairs from all the wiki links in Wikipedia. Other alternative names are extracted from redirect pages, disambiguation pages and title variants. Any span  $ref_k$  from  $d$  which is a known anchor text (or alternative name) becomes a possible reference. All known targets of a possible reference  $ref_k$  are considered candidate topics  $ct_{k,l}$ . This yields a low precision–high recall list of candidates  $C = \{ct_{k,l} | k = 1 \dots m, l = 1 \dots r_k\}$ , which needs to be filtered.

The system employs a two-step approach: **pruning** and **selection**. In the first step candidate topics are filtered out based on their semantic relatedness to the set of seeds and to each-other. For each mention  $ref_k$  the topics with low overall confidence are removed. Section 5.4.1 describes how machine learning is used to achieve this. This step reduces the number of candidates, while maintaining high recall. In the second step, the system determines which of the remaining candidates should be selected by giving more weight to the local context of the actual topic mentions and by dealing with overlapping references. The aim of this step is to increase precision. It is described further in Section 5.4.2.

### 5.4.1 Candidate pruning

This step is essentially a form of weak disambiguation. For each mention  $ref_k$ , candidate topics that are semantically relevant to the document are passed to the next step, while those with low relevance are removed. The main aims are to discard weak candidates and to decrease complexity by favouring recall over precision.

## Context Weighting

The system relies on the relatedness between candidate topics and the reference set of seed topics (the context). The importance of each seed topic  $s_i$  is first ranked based on occurrences in the document  $d$  as well as frequency information from Wikipedia and average semantic relatedness between seeds. Several ranking methods were tested empirically, as there is no “gold standard” for ranking the set of seed topics to enable the direct measurement of performance. These methods assign a weight to seed as follows:

- **simple** – all seeds have equal weight:  $w_i = 1$
- **tf** – the weight of seed  $s_i$ , is proportional to  $tf_i$ , the number of occurrences of its known anchors in the document (possible seed mentions):  $w_i = \log(1 + tf_i)$
- **tf·idf** – considers possible occurrences in the document as well as the number of Wikipedia articles which have a link to each seed:  $w_i = \log(1 + tf_i) \cdot \log(\frac{W}{inlinks_i})$ , where  $W$  is the number of articles in Wikipedia,  $inlinks_i$  represents the number of articles that link to the topic  $s_i$
- **avg-rel** – the average semantic relatedness to the other seeds, using the measure proposed by Milne and Witten (2008b):  $w_i = \frac{1}{N} \sum_{j=1}^N rel(s_i, s_j)$
- **avg-rel-tf·idf** – combines the previous method with the weights provided by the **tf·idf** method:  $w_i = \frac{\sum_{j=1}^N rel(s_i, s_j) \cdot w_j^{tf·idf}}{\sum_{j=1}^N w_j^{tf·idf}}$
- **links-in** – indicates the number of seed topics that have a direct wiki link to this article:  $w_i = |S \cap inlinks_i|$
- **links-out** – indicates the number of seed topics that article  $s_i$  has a direct wiki link to:  $w_i = |S \cap outlinks_i|$
- **links-all** – indicates the number of seed topics that article  $s_i$  is linked to:  
 $w_i = |S \cap \{outlinks_i \cup inlinks_i\}|$

## Candidate relevance

The next step is to rank the importance of each candidate topic  $ct_{k,l}$ , by considering all  $\{S, W^m, ct_{k,l}\}$  triples independently. Candidates whose relevance does not pass a

threshold value are discarded, pruning the decision space. The threshold is a parameter which controls the trade-off between *precision* (mainly valid candidates are selected) and *recall* (most valid candidates are selected). A point-wise learning to rank approach is employed to order the candidates.

A machine learning dataset is created using features which characterise the context (number of seeds, average weights), the candidate topic (prior link probability, number of references, number of in- and out-links), and the compatibility/relatedness between the candidate topic and the set of seeds. To create training data, a random set of Wikipedia articles is used. For each article, a fraction of its wiki links are considered as seeds, while the remainder make the set of valid targets. Only a subset of all the ambiguous references are used, namely those mentions  $ref_k'$  that have at least one candidate topic amongst the target articles. The resulting data is split into training and development sets to avoid problems that can arise when using cross-validation alone, such as duplicate instances with copies in more than one cross-validation fold. When applied to real-world documents, while the seed set must be supplied as input, the set of valid targets is not known and the model must make predictions for all ambiguous mentions  $ref_k$ . Without a gold standard created specifically for densification, this bias induced by sampling the training data set is unavoidable. By varying the amount of seed data, we can gain insights into the performance of the model.

## Experiments

A random set of 100 Wikipedia articles were selected for training the models, and another random set of 100 articles were used for testing. Point-wise ranking is employed: for each candidate the model produces a confidence score. Then, for each reference a simple criterion is applied to select candidates, e.g., only selecting those that pass a threshold or the top 3 highest ranked candidates per reference or both. The seeds are weighted using the methods mentioned in the previous step, while the features describing the topic are directly extracted from Wikipedia statistics. The most important features are those reflecting the semantic association between a candidate topic and the set of seeds. Two

Table 5.1:  $\chi^2$  ranking of the candidate pruning features used to measure the relatedness between a candidate target and the set of seed topics

| Average $\chi^2$ merit<br>and variation |              | Average<br>rank | Base relatedness<br>measure | Aggregation<br>method | Seed topics'<br>weighting |
|---|--------------|-----------------|-----------------------------|-----------------------|---------------------------|
| 7091.067                                | $\pm 30.662$ | 1               | WM                          | max                   | <i>tf.idf</i>             |
| 6826.216                                | $\pm 22.576$ | 2               | WM                          | top10                 | <i>tf.idf</i>             |
| 5750.507                                | $\pm 26.079$ | 3               | WM                          | avg                   | avg-rel                   |
| 5654.750                                | $\pm 19.512$ | 4               | WM                          | top10                 | <i>tf</i>                 |
| 5526.763                                | $\pm 13.919$ | 5               | WM                          | avg                   | simple                    |
| 5467.932                                | $\pm 16.275$ | 6               | WM                          | avg                   | avg-rel                   |
| 5102.377                                | $\pm 46.585$ | 7               | WM                          | max                   | <i>tf</i>                 |
| 4850.794                                | $\pm 25.915$ | 8               | WM                          | avg                   | links-out                 |
| 4687.179                                | $\pm 20.891$ | 9               | WM                          | avg                   | link-all                  |
| 4643.823                                | $\pm 30.256$ | 10              | WM                          | top10                 | avg-rel                   |
| 4550.655                                | $\pm 28.174$ | 11              | WM                          | avg                   | <i>tf.idf</i>             |
| 4258.057                                | $\pm 46.608$ | 12.7            | WM                          | max                   | avg-rel                   |
| 4237.392                                | $\pm 25.958$ | 12.8            | links                       | count                 | links-in                  |
| 4213.878                                | $\pm 27.175$ | 13.5            | WM                          | avg                   | <i>tf</i>                 |
| 4149.599                                | $\pm 26.924$ | 15              | WM                          | avg                   | links-in                  |
| 4054.404                                | $\pm 20.434$ | 16              | WM                          | top10                 | simple                    |
| 3010.643                                | $\pm 18.823$ | 17.2            | WM                          | top10                 | avg-rel                   |
| 2997.218                                | $\pm 23.362$ | 17.8            | WM                          | max                   | simple                    |
| 2816.212                                | $\pm 36.212$ | 19              | links                       | count                 | links-all                 |
| 2433.655                                | $\pm 32.184$ | 20.5            | links                       | islink                | link-all                  |
| 2434.678                                | $\pm 11.281$ | 20.5            | WM                          | top10                 | links-out                 |
| 2279.353                                | $\pm 20.949$ | 22.1            | WM                          | max                   | avg-rel                   |
| 2214.805                                | $\pm 24.357$ | 22.9            | links                       | count                 | links-out                 |
| 2023.238                                | $\pm 9.356$  | 24              | WM                          | top10                 | link-all                  |
| 1907.707                                | $\pm 13.352$ | 25              | WM                          | max                   | links-in                  |
| 1838.768                                | $\pm 7.598$  | 26              | WM                          | top10                 | links-in                  |
| 1676.377                                | $\pm 13.025$ | 27              | links                       | islink                | links-in                  |
| 1519.758                                | $\pm 16.815$ | 28              | WM                          | max                   | links-out                 |
| 1071.573                                | $\pm 9.190$  | 29              | WM                          | max                   | link-all                  |
| 868.720                                 | $\pm 15.587$ | 30              | links                       | islink                | links-out                 |

base measures are used: WikipediaMiner’s semantic relatedness measure, which is based on the average Jaccard index of the link-neighbourhood of two articles, and a measure that accounts for direct mentions/links between the articles, information which is not used by the previous measure. The base measure needs to be aggregated to account for the fact that the context has several seeds, each mention has several candidates, and each topic has several possible anchors/mentions. For direct-links, two aggregation methods are used: link count (*count*) and has-at-least-a-link indicator (*islink*). For semantic relatedness, average (*avg*), maximum (*max*), and average of top-10 nearest seed topics (*top10*) are used. The different seed topic weighting methods are combined with different relatedness functions with these aggregation methods. The resulting relatedness scores are used as features by the statistical machine learning methods. Table 5.1 shows the  $\chi^2$  ranking of these features.

## Results

For comparison, the open-source WikipediaMiner (WM) package was used to train and test a disambiguation model on the same set of articles (see Table 5.2). The WM model performs slightly worse than the original results reported on a different sample of articles from an older version of Wikipedia. This suggests that the problem is more difficult in the newer version of Wikipedia, due to the increase in size, number of pages/links, and article length. The main direct cause is the fact that, on average, the frequency of the most common sense has decreased. The WM model relies on the prior probability of a target given the anchor, which amounts to a strong bias towards the most frequent target baseline. According to Cucerzan (2007), this baseline yields good results on text from Wikipedia, better than when used on newswire text. Overall, the features obtained using the WM semantic relatedness measure had higher predictive power than those using the link-based measure. None of the aggregation methods consistently outrank the others. As expected, there is a high positive correlation between features resulting from applying different aggregation methods to the same base relatedness measure.

The best model uses logistic regression and outperforms the state-of-the-art, halving the error rate. Logistic regression is suitable for the binary dataset in this research, as the

Table 5.2: Candidate pruning performance

| Model               | Accuracy | Error Rate | Kappa  |
|---------------------|----------|------------|--------|
| WM-disambig         | 91.33%   | 8.66%      | 0.7551 |
| logistic regression | 95.37%   | 4.06%      | 0.7773 |

confidence value output by the algorithm is actually the probability that the instance is a valid topic. This means that sorting topics based on this confidence score yields a robust ranking, which is desirable as it allows the use of a threshold value to favour recall over precision. Table 5.3 illustrates that by varying the hyper-parameter  $\theta$  the system can be biased towards achieving high recall (few false negatives). This model reaches impressive recall levels (97%) while maintaining reasonable precision (50%). In this dataset, on average, there are 10 candidates for each mention, one of which is considered valid. Using the regression model discards, on average, 8 out of 10 candidates.

Table 5.3: Trade-off between recall and precision for candidate pruning

| Threshold $\theta$ | True pos. | False pos. | False neg. | True neg. | R   | P   | F <sub>0.5</sub> | F <sub>1</sub> | F <sub>2</sub> |
|--------------------|-----------|------------|------------|-----------|-----|-----|------------------|----------------|----------------|
| 0.40               | 9600      | 1300       | 3400       | 104000    | 73% | 87% | <b>84%</b>       | 79%            | 75%            |
| 0.30               | 9900      | 1700       | 3100       | 104000    | 76% | 85% | 83%              | <b>80%</b>     | 78%            |
| 0.05               | 12800     | 13100      | 300        | 92700     | 97% | 49% | 54%              | 65%            | <b>81%</b>     |

#### 5.4.2 Candidate selection

In the second step, the focus shifts from the global-context, overall relatedness to all seeds, to the local-context of each mention. For each mention, a score is computed factoring in the relatedness to seeds and candidates mentioned nearby. The distance between mentions, measured in words/tokens, is used to weight the relatedness score. For each nearby mention only the most related candidate is considered. The length of an anchor is also used as a feature, because longer anchors tend to be less ambiguous. Short mentions overlapped by long mentions must be linked to one of the targets of the surrounding mention, otherwise they are discarded. These criteria become numeric features which can be used to train a classifier or a regression model. Unfortunately, as its markup

is incomplete, information extracted from Wikipedia is no longer sufficient to create training data automatically. Without validation data that can quantify the performance of a particular method, a relevance estimation function – a linear combination of the numeric features – is used to rank the overall relevance of each topic candidate. The system then selects the topics at the top of this list. The number of topics selected can be expressed relative to the total number of candidates or to the number of seeds.

The weights used in the scoring function were established empirically, giving more importance to seeds than to topics. To better determine a set of weights, an extrinsic validation methodology was employed: a supervised document classification task using a subset of the 20-newsgroups dataset<sup>10</sup> (Lang, 1995), consisting of 100 files for each newsgroup. After removing headers and email addresses, the resulting 2000 files average 272.93 words per file. Unlike the usual 60-40 split between training and testing (or the 90-10 split used in 10-fold cross validation), only 10% of the documents are used for training and the remaining 90% of data is used for testing, because such a split forces a document classifier to generalise, i.e., to rely more on semantic relatedness rather than lexical overlap. All documents are first wikified using WM to select the 5 most important topics per document as seeds. Documents are then represented as feature vectors, each coordinate corresponding to a Wikipedia article weighted according to the relevance score computed by the densification system. Different relevance estimation functions are used with different weights. As a baseline, null-relevance densification is used: it passes through the input seeds, but does not add any new topics, which is equivalent to using the wikification system alone. The intuition is that the best performance in this classification task is achieved by choosing a good relevance weighting function.

## 5.5 Evaluation

The main reason that wikification is popular is the availability of large amounts of training and testing data directly from Wikipedia. The existing approaches give much more weight

---

<sup>10</sup><http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html>

to predicting precise prominent targets, rather than to completeness. Densification aims to find new links in addition to the existing ones, therefore both precision and recall need to be calculated to measure performance, which requires creating a reference gold standard annotation, which can be a costly process. Therefore it is necessary to also consider alternative ways to estimate performance.

Section 5.5.1 describes an experiment involving human raters carried out to determine whether or not agreement is high enough to warrant the development of a large scale gold standard annotation. In a second experiment (Section 5.5.2), a set of Wikipedia articles are used to automatically estimate the recall of the proposed densification method by measuring how many of the existing wiki links are also predicted by the system. Section 5.5.3 presents a third experiment which evaluates the impact the densification system has on EQUAL, the question answering system presented in Chapter 4.

### 5.5.1 Human evaluation

To reflect the ability to find less obvious yet relevant links, an experiment involving human annotators was performed to determine the agreement level for this task. Such an annotation can be employed to create a reference gold standard, or to evaluate the performance of systems. For a sample of paragraphs from Wikipedia articles, the set of candidate topics was extracted, excluding those present in the original Wikipedia markup. Human annotators were asked to assess the targets of these links using three criteria: *correctness* of the proposed link, its *relatedness* to the text and the *type* of the target article:

- **correctness** – the proposed target is a plausible disambiguation for the anchor text in this document. This criterion reflects the ability to discard targets which are obviously incorrect using a 3-level scale, see Table 5.4a
- **relatedness** – the content of the target article helps in understanding the current document. This criterion distinguishes easy but irrelevant targets (such as numbers or dates) using a 5-level scale, see Table 5.4b

- **type** – the description included in the target page matches the type of the anchor text. This criterion distinguishes five coarse categories described in Table 5.4c

The experiment focused on four paragraphs of text, totalling 94 links to 48 candidate targets. The six annotators were given an explanation of the scoring scheme as well as examples. In the first step, annotators were shown a short text, usually a paragraph, which they had to read. They then had to score the three criteria for each candidate link. While such an experiment works towards creating a gold standard, the main goal here was to study the three criteria, the agreement between annotators and the types of links found in the sample data. This provides insights into the feasibility of creating a gold standard for this task.

### **Correctness**

Correctness is measured on an ordinal scale, but it can also be considered nominal (categorical) as there are only 3 values. Overall, the six annotators marked 79.17% targets as correct, 14.58% as possible and just 6.25% as incorrect. Because of the skewness of the dataset three inter-rater agreement measures are used: simple agreement, Cohen's kappa (Cohen, 1960) and Krippendorff's alpha (Krippendorff, 2004).

Table 5.5a shows pairwise agreement considering categorical ratings. The level of agreement is high, but a better picture emerges when looking at Cohen's kappa coefficient for pairwise agreement in Table 5.5b. The average level of agreement of 51.75% is considered good for three categories. The average agreement of two raters (5 and 6) seems to be substantially lower than the other four participants, suggesting that they used a narrower sense for 'correctness'. The average kappa for the remaining set of four raters is 63%, which is slightly better. This result can be considered satisfactory, as kappa is affected by prevalence (Sim and Wright, 2005), and on this dataset the codes are not equiprobable, instead their probabilities vary considerably. In addition, kappa does not capture the ordinal nature of the criterion: there are only three cases of 'extreme'

Table 5.4: Criteria assessed by human raters

| (a) Criterion A: <b>correctness</b> of the proposed target page for the keyword |     |   |
|---|-----|---|
| Correct   | I   | the target proposed corresponds to the keyword, e.g., the same title, a synonym or an alternative name  |
| Possible  | II  | the target is related to the keyword, but less exact, e.g., keyword <i>expensive</i> - target <i>Cost</i>   |
| Incorrect   | III | erroneous target, e.g., keyword <i>foreign visitors</i> - target <i>Foreign Policy (magazine)</i>   |
| (b) Criterion B: <b>relatedness</b> of the proposed target page                 |     |   |
| Very relevant   | I   | target page is central to the text, e.g., several valid mentions in the text, or a strong relation to the topic, usually entities or specific terminology |
| Relevant  | II  | target page describes an entity or concept referred to in the text; it should have at least one valid mention in the text                                 |
| Somewhat relevant   | III | target page provides additional information, but less important (e.g., standard terminology, partial entity names, disambiguation pages)                  |
| Not relevant  | IV  | irrelevant, generic terms e.g., years, dates, measurement units, numbers  |
| Other   | V   | if criterion A was deemed Incorrect   |
| (c) Criterion C: <b>type</b> of the proposed target page                        |     |   |
| Named Entity  | I   | people, organizations, places, events, albums, etc. e.g., <i>Paris</i> , <i>Picasso</i> , <i>World War I</i>  |
| Term  | II  | terminology, scientific or technical concepts, common words e.g., <i>public house</i> , <i>capital city</i> , <i>isomer</i> , <i>CPU</i>                  |
| Topic   | III | a generic topic, e.g., <i>Geography of Germany</i> , <i>Characters of Lost</i> , <i>Age of Enlightenment</i> , <i>Partition of India</i>                  |
| Disambiguation  | IV  | a listing of pages with similar titles e.g., <i>Paris (disambiguation)</i>  |
| Other   | V   | none of the above e.g., list pages ( <i>List of sovereign states</i> ) or dates ( <i>2003</i> , <i>21st century</i> , <i>March 01</i> )                   |

Read the text:

Caernarfon is the traditional county town of the historic county of Caernarfonshire. The town is best known for its great stone castle, built by Edward I of England and consequently sometimes seen as a symbol of English domination. Edward's architect, James of St. George, may well have modelled the castle on the walls of Constantinople, possibly being aware of the alternative Welsh name *Caer Gystennin*; in addition, Edward was a supporter of the Crusader cause. On higher ground on the outskirts of the town are the remains of an earlier occupation, the **Segontium Roman Fort**.

Is the target [Segontium](#) correct and/or relevant?

## Segontium

Coordinates: 53.1373°N 4.2659°W

From Wikipedia, the free encyclopedia

**Segontium** is a Roman fort for a Roman auxiliary force, located on the outskirts of Caernarfon in Gwynedd, north Wales.

It probably takes its name from the nearby River Seiont, and may be related to the Segontiaci, a British tribe mentioned by Julius Caesar. The fort was founded by Agricola in 77 or 78 AD after he had conquered the Ordovices. It was the main Roman fort in the north of Roman Wales and was designed to hold about a thousand auxiliary infantry. It was connected by a Roman road to the Roman legionary base at Chester, Deva Victrix. Unlike the more recent Caernarfon Castle alongside the Seiont estuary, Segontium is located on higher ground giving a good view of the Menai Straits.

The original timber defences were rebuilt in stone in the first half of the 2nd century AD. An inscription on an aqueduct from the time of the Emperor Septimius Severus indicates that at that time it was garrisoned by Cohors I Sunicorum, which would have originally been levied among the Sunci of Gallia Belgica.

**Correct** ☒ Correct ☐ Possible ☐ Incorrect ☐ Other

**Related** ☒ Very relevant ☐ Relevant ☐ Somewhat relevant ☐ Not relevant ☐ Other

**Type** ☒ Entity ☐ Term ☐ Topic ☐ Disambiguation ☐ Other

Answer

Comment:

Figure 5.1: User interface employed in the experiment

disagreement – when a target is rated *correct* by the first rater and *incorrect* by the second rater. Two of these cases involve both the two stricter annotators (4 and 5). In the third outlier case, the rater felt that the link was ‘unnecessary’ as the target article was the exact page the text was extracted from, i.e., *Paris*.

Table 5.5c shows pairwise agreement using Krippendorff’s alpha (Ka) (Krippendorff, 2004), a coefficient that generalises a number of inter-rater agreement statistics, such as Scott’s Pi (Scott, 1955), Fleiss’ kappa (Fleiss, 1971) and Spearman’s rank correlation coefficient rho (Spearman, 1904). It can be used for various types of data (including nominal, ordinal and interval), and, like kappa, it also considers the level of chance agreements. For more than two annotators, one can use either the average pairwise Ka (61.71%) or the overall Ka statistic for all annotators (62.30%). The two stricter annotators also stand out using this index, with lower average pairwise agreement. Removing their ratings, the Ka statistic increases from 62.30% (for 6 annotators) to 73.93% (for 4 annotators). A level of 77.30% is reached if the third outlier rating mentioned previously is also removed from the dataset. Agreement above 80% indicates strong agreement, while values over 70% are considered as sufficient agreement.

## **Relatedness**

One of the purposes of this experiment was to determine whether the proposed criterion of relatedness corresponds to a natural attribute of the target that human raters can recognise without special training. The “guidelines” given to the raters described the scale and provided several examples, rather than contain an exhaustive set of strict rules. The criterion was measured using a 5-point scale, but codes 4 (*Not relevant*) and 5 (*Other*) are essentially equivalent. Code 5 should be used for irrelevant targets which had been deemed ‘incorrect’; however, just two raters used this separate code, therefore we map these ratings on a 4-point ordinal scale.

As Table 5.6a shows, the average simple agreement is substantially lower than for the correctness criterion. To reflect the ordinal nature of the scale, a modified version of the

Table 5.5: Agreement for criterion **Correctness**

| (a) Pairwise simple agreement               |       |       |              |       |       |                   |
|---|-------|-------|--------------|-------|-------|-------------------|
|   | 2     | 3     | 4            | 5     | 6     | Annotator average |
| 1   | 87.50 | 87.50 | 83.33        | 81.25 | 83.33 | 84.58             |
| 2   |       | 87.50 | <b>91.67</b> | 83.33 | 83.33 | 86.67             |
| 3   |       |       | 83.33        | 77.08 | 87.50 | 84.58             |
| 4   |       |       |              | 79.17 | 81.25 | 83.75             |
| 5   |       |       |              |       | 72.92 | 78.75             |
| 6   |       |       |              |       |       | 81.67             |
|   |       |       |              |       |       | 83.33             |
| (b) Pairwise Kappa agreement                |       |       |              |       |       |                   |
|   | 2     | 3     | 4            | 5     | 6     | Annotator average |
| 1   | 62.89 | 65.55 | 54.07        | 53.94 | 40.19 | 55.33             |
| 2   |       | 64.36 | <b>76.24</b> | 57.94 | 36.32 | 59.55             |
| 3   |       |       | 54.82        | 44.71 | 57.33 | 57.36             |
| 4   |       |       |              | 49.74 | 36.00 | 54.17             |
| 5   |       |       |              |       | 22.10 | 45.69             |
| 6   |       |       |              |       |       | 38.39             |
|   |       |       |              |       |       | 51.75             |
| (c) Pairwise Krippendorff's alpha (ordinal) |       |       |              |       |       |                   |
|   | 2     | 3     | 4            | 5     | 6     | Annotator average |
| 1   | 70.21 | 70.81 | 70.24        | 58.67 | 45.96 | 63.18             |
| 2   |       | 77.74 | 88.60        | 67.73 | 52.84 | 71.42             |
| 3   |       |       | 67.56        | 55.74 | 55.83 | 65.54             |
| 4   |       |       |              | 67.31 | 44.35 | 67.61             |
| 5   |       |       |              |       | 32.10 | 56.31             |
| 6   |       |       |              |       |       | 46.22             |
|   |       |       |              |       |       | 61.71             |

agreement measure can be used, which allows for one-point disagreements between a pair of raters. Table 5.6b shows that the majority of the pairwise disagreements are one scale point differences in rating. This means that agreement was relatively low when relatedness ranking is seen as a categorical rating task (also reflected by the kappa index in Table 5.6c), but higher agreement levels were achieved when using an ordinal index.

Table 5.6d shows that overall, the level of agreement using Krippendorff's alpha is indeed lower than for the previous criterion, with only half of raters achieving an average pairwise index above 60% (raters 2,4 and 5). The pair that agrees best (raters 4 and 5) have a moderate agreement index of 66.85% while the average pairwise index is 51.84%. This suggests that there is too much subjectivity when assessing the relatedness criterion on a 4-point ordinal scale. Table 5.7 illustrates this point: rating I (Very Relevant) has higher prevalence for annotator 4 (23 items, vs. a median of 10 items); rating II (Relevant) has high prevalence for annotators 1 and 2 (though they agree on this code less than half of the time); while rating IV is used by rater 6 twice as frequently as the overall average. The disagreements are inconsistent even for pairwise decisions. By mapping the ratings to a smaller scale (3-point or even 2-point), simple agreement is the only inter-rater agreement index that increases. Both kappa and alpha decrease slightly, strongly suggesting that there is a high degree of subjectivity. The conclusion is that relatedness is a less generic and a more subjective concept than correctness: whether a target is deemed related to a text or unrelated depends on the task at hand, as well as on the rater's knowledge of the topic. Some raters suggested the concept of relevance as a less ambiguous alternative criterion (targets that are mentioned more often are more relevant), but relevance does not consider the actual contents of the target page.

## Type

The third criterion was designed to be used as a control signal, as the distinction between named entities, terms and topics seemed a natural intuition. Analysis of this data revealed two notable findings. The first finding concerns code *V (Other)* which should to be used for index pages from Wikipedia, such as lists of entities or dates/years. These pages,

Table 5.6: Agreement for criterion **Relatedness**

| (a) Pairwise simple agreement |       |       |       |       |       |                   |
|-------------------------------|-------|-------|-------|-------|-------|-------------------|
|                               | 2     | 3     | 4     | 5     | 6     | Annotator average |
| 1                             | 52.08 | 39.58 | 35.42 | 45.83 | 27.08 | 40.00             |
| 2                             |       | 43.75 | 35.42 | 64.58 | 33.33 | 45.83             |
| 3                             |       |       | 56.25 | 41.67 | 47.92 | 45.83             |
| 4                             |       |       |       | 35.42 | 39.58 | 40.42             |
| 5                             |       |       |       |       | 37.50 | 45.00             |
| 6                             |       |       |       |       |       | 37.08             |
|                               |       |       |       |       |       | 42.36             |

| (b) Pairwise ordinal agreement allowing one scale point mismatches between raters |       |       |       |       |       |                   |
|---|-------|-------|-------|-------|-------|-------------------|
|   | 2     | 3     | 4     | 5     | 6     | Annotator average |
| 1   | 89.58 | 95.83 | 95.83 | 87.50 | 83.33 | 90.42             |
| 2   |       | 87.50 | 89.58 | 93.75 | 81.25 | 88.33             |
| 3   |       |       | 87.50 | 85.42 | 83.33 | 87.92             |
| 4   |       |       |       | 87.50 | 75.00 | 87.08             |
| 5   |       |       |       |       | 89.58 | 88.75             |
| 6   |       |       |       |       |       | 82.50             |
|   |       |       |       |       |       | 87.50             |

| (c) Pairwise Kappa agreement |       |       |       |       |       |                   |
|------------------------------|-------|-------|-------|-------|-------|-------------------|
|                              | 2     | 3     | 4     | 5     | 6     | Annotator average |
| 1                            | 23.97 | 15.02 | 20.26 | 21.80 | 8.94  | 18.00             |
| 2                            |       | 24.21 | 18.87 | 50.84 | 18.25 | 27.23             |
| 3                            |       |       | 39.57 | 22.04 | 30.96 | 26.36             |
| 4                            |       |       |       | 16.92 | 19.26 | 22.97             |
| 5                            |       |       |       |       | 22.08 | 26.74             |
| 6                            |       |       |       |       |       | 19.90             |
|                              |       |       |       |       |       | 23.53             |

| (d) Pairwise Krippendorff's alpha (ordinal) agreement |       |       |       |       |       |                   |
|---|-------|-------|-------|-------|-------|-------------------|
|   | 2     | 3     | 4     | 5     | 6     | Annotator average |
| 1   | 48.59 | 48.38 | 54.94 | 48.59 | 37.65 | 47.63             |
| 2   |       | 44.05 | 52.97 | 66.85 | 52.54 | 53.00             |
| 3   |       |       | 56.33 | 45.41 | 48.29 | 48.49             |
| 4   |       |       |       | 59.50 | 46.34 | 54.01             |
| 5   |       |       |       |       | 64.41 | 56.95             |
| 6   |       |       |       |       |       | 49.85             |
|   |       |       |       |       |       | 51.66             |

Table 5.7: Number of targets per code for criterion **Relatedness**

| Rater   | Code      |           |       |           |
|---------|-----------|-----------|-------|-----------|
|         | I         | II        | III   | IV        |
| 1       | 6         | <b>30</b> | 9     | 3         |
| 2       | 10        | <b>24</b> | 7     | 7         |
| 3       | 12        | 14        | 18    | 4         |
| 4       | <b>23</b> | 5         | 15    | 5         |
| 5       | 11        | 18        | 10    | 9         |
| 6       | 10        | 7         | 16    | <b>15</b> |
| average | 12.00     | 16.33     | 12.50 | 7.17      |
| median  | 10.50     | 16.00     | 12.50 | 6.00      |

while important for aiding navigation and for providing context for a particular target, do not have one main subject (the encyclopaedic entry), which should be the case for the first three codes (I named entity, II term, III topic). Results show that raters 4 and 5 did not use the whole scale to rate targets, suggesting perhaps an omission in the guidelines. The second finding is that raters had very low agreement for code *III (Topic)*. The typical disagreement was for targets such as *Economy*, which could be considered both general terms and topics in human knowledge. Most topics present in the dataset are rather generic and are affected by this problem. More specific topics such as *Economy of France* which are easily distinguishable from terms such as *Gross Domestic Product (GDP)* are less frequent in the data used. To address this issue, the examples given as guideline need to be amended to reflect such borderline cases.

When measuring the agreement on one code vs. all the others, raters agreed best for code *I (Named Entities)* with a Krippendorff's alpha coefficient of 79.4%, while for code *II (Terms)* the agreement was 64%. The other codes, which were used for fewer targets, demonstrated lower levels of agreement: 50.3% for codes IV and V (combined), and 4.62% for code *III (Topic)* due to its very low prevalence. If the coding is mapped onto a 3-category scheme (named entities, terms and other), the agreement amongst all raters is 65% and amongst the first three best raters it is 77.3%. The best pair of annotators (2 and 4) achieve a Krippendorff's alpha coefficient of 84.1% using the original 5-categories scale.

Table 5.8: Agreement for criterion **Type**

| (a) Pairwise simple agreement                         |       |       |       |       |       |                   |
|---|-------|-------|-------|-------|-------|-------------------|
|   | 2     | 3     | 4     | 5     | 6     | Annotator average |
| 1   | 77.08 | 66.67 | 70.83 | 58.33 | 68.75 | 68.33             |
| 2   |       | 83.33 | 89.58 | 72.92 | 79.17 | 80.42             |
| 3   |       |       | 79.17 | 72.92 | 75.00 | 75.42             |
| 4   |       |       |       | 72.92 | 70.83 | 76.67             |
| 5   |       |       |       |       | 75.00 | 70.42             |
| 6   |       |       |       |       |       | 73.75             |
|   |       |       |       |       |       | 74.17             |
| (b) Pairwise Kappa agreement                          |       |       |       |       |       |                   |
|   | 2     | 3     | 4     | 5     | 6     | Annotator average |
| 1   | 65.33 | 47.40 | 57.14 | 39.39 | 51.09 | 52.07             |
| 2   |       | 72.07 | 84.00 | 58.76 | 65.93 | 69.22             |
| 3   |       |       | 66.64 | 57.38 | 56.85 | 60.07             |
| 4   |       |       |       | 59.85 | 54.35 | 64.40             |
| 5   |       |       |       |       | 60.66 | 55.21             |
| 6   |       |       |       |       |       | 57.78             |
|   |       |       |       |       |       | 59.79             |
| (c) Pairwise Krippendorff's alpha (nominal) agreement |       |       |       |       |       |                   |
|   | 2     | 3     | 4     | 5     | 6     | Annotator average |
| 1   | 65.41 | 47.20 | 57.30 | 39.18 | 51.06 | 52.03             |
| 2   |       | 72.28 | 84.14 | 58.85 | 66.04 | 69.34             |
| 3   |       |       | 66.73 | 57.01 | 56.88 | 60.02             |
| 4   |       |       |       | 60.05 | 54.28 | 64.50             |
| 5   |       |       |       |       | 60.61 | 55.14             |
| 6   |       |       |       |       |       | 57.77             |
|   |       |       |       |       |       | 59.80             |

For this pair of raters there is no target they simultaneously consider as code *III* (*Topic*), but all their disagreements occur when one of them uses code *III* (10% of targets), while all the other targets have perfect agreement. This is a positive result, showing that strong agreement can be achieved. The detailed agreement results for the type criterion can be found in Table 5.8.

Overall, the results of this experiment suggest that the level of agreement is average for criteria **relatedness** and **type**, and good for criterion **correctness**. Some of the disagreement can be explained by the fact that the guidelines were not comprehensive, there was no specialised training for the raters and the annotators were not themselves experts. Additionally, the distribution of the scores was skewed, which explains the large difference between simple agreement and measures of agreement adjusted for chance, such as Cohen’s kappa and Krippendorff’s alpha. The experiment was designed to determine the reliability and generality of the first two criteria, i.e., whether they correspond to natural attributes characterising a target. Therefore the guidelines were intentionally rather vague, to allow raters to use their common sense rather than be restricted by a comprehensive set of rules. While the notion of **correctness**, a fuzzy version of ‘identity’ or ‘cross-document coreference’ seems to correspond to this intuition, **relatedness** appears to be a subjective, task-dependent and somewhat artificial concept. This finding casts doubts over the practicality of developing a large-scale, general-purpose gold standard annotation for **relatedness**. Based on these results, voting amongst raters can be used to build a reliable reference annotation for **correctness** and **type** using a slightly amended set of guidelines. Brief training of the raters is also recommended.

### 5.5.2 Wikipedia markup evaluation

This section evaluates the performance of densification by automatically estimating recall. In this experiment, the text of Wikipedia articles is used together with a random sample of the existing links that form the set of seed topics. The set of remaining links, deemed relevant by Wikipedia editors, is considered a biased sample of important links

the system should find and is used as a proxy to estimate recall – a subset of valid targets. For each document, the recall score is averaged over 10 random splits. The main advantage of this method is that it can be applied on a large scale as it does not require manual data. The disadvantage is that precision cannot be estimated automatically, but needs to be estimated by humans, using for example, a method similar to that described in the previous section. Off-the-shelf wikification systems should perform well in this scenario, as only the ‘prominent’ links are considered for evaluation. Different seed ratios were used to study the impact of input set size.

To better understand the performance of the proposed densification model, several other off-the-shelf tools were also tested on the same dataset. These tools were briefly described in Section 5.2, and a summary of their characteristics is outlined in Table 5.9. Almost all tools identify both entities and terms, though not all directly create links to Wikipedia or to the Linked Open Data cloud. Four of the seven systems provide descriptions of their approach: two of them use a link-based measure to rank and disambiguate targets, and two use a word-based approach. A recent analysis of popular linked data entity extractors is given by Rizzo and Troncy (2011).

Table 5.9: Wikification tools

|              | Alchemy<br>API | DBpedia<br>Spotlight | Open<br>Calais | thewiki<br>machine | Wikipedia<br>Miner | Zemanta<br>API | Densifi<br>cation |
|--------------|----------------|----------------------|----------------|--------------------|--------------------|----------------|-------------------|
| Link targets |                |                      |                |                    |                    |                |                   |
| • entities   | ✓              | ✓                    | ✓              | ×                  | ✓                  | ✓              | ✓                 |
| • terms      | ✓              | ✓                    | ×              | ✓                  | ✓                  | ✓              | ✓                 |
| Similarity   |                |                      |                |                    |                    |                |                   |
| • word-based | ?              | ✓                    | ?              | ✓                  | ×                  | ?              | ×                 |
| • link-based | ?              | ×                    | ?              | ×                  | ✓                  | ?              | ✓                 |
| Open source  | ×              | ✓                    | ×              | ×                  | ✓                  | ×              | n/a               |
| Fair-use     | ✓              | ✓                    | ✓              | ✓                  | ✓                  | ✓              | n/a               |
| Quota        | ✓              | ✓                    | ✓              | ✓                  | ×                  | ✓              | n/a               |
| Subscription | ×              | ×                    | ×              | ×                  | ×                  | ×              | n/a               |

Table 5.10 presents the performance achieved by the different wikification tools using

Table 5.10: Performance achieved by wikification tools using different seed ratios

| (a) Macro-averaged Recall (and standard deviation) |               |               |               |               |
|--|---------------|---------------|---------------|---------------|
| System   | sr=0.1        | sr=0.3        | sr=0.5        | sr=0.7        |
| aapi   | 29.97 (15.95) | 30.03 (16.11) | 30.25 (16.16) | 31.54 (17.05) |
| dbb  | 48.68 (16.70) | 48.48 (16.86) | 48.20 (16.26) | 48.91 (18.37) |
| oc   | 25.44 (14.39) | 25.34 (14.22) | 25.13 (14.77) | 27.64 (14.94) |
| twm2   | 8.20 (17.24)  | 8.89 (17.95)  | 9.85 (18.15)  | 12.49 (21.37) |
| wm1  | 61.80 (18.16) | 61.88 (17.88) | 61.13 (17.89) | 61.54 (20.68) |
| wm2  | 61.81 (18.62) | 61.89 (18.26) | 61.19 (18.41) | 61.86 (21.20) |
| zem  | 57.36 (17.40) | 57.53 (17.60) | 57.50 (17.09) | 57.19 (18.88) |

| (b) Macro-averaged Precision (and standard deviation) |               |               |               |               |
|---|---------------|---------------|---------------|---------------|
| System  | sr=0.1        | sr=0.3        | sr=0.5        | sr=0.7        |
| aapi  | 52.86 (15.52) | 47.41 (15.31) | 40.16 (14.67) | 30.13 (13.47) |
| dbb   | 32.89 (11.92) | 27.92 (11.28) | 22.02 (10.03) | 14.95 (8.18)  |
| oc  | 51.93 (15.42) | 45.91 (16.25) | 38.20 (15.96) | 29.28 (14.55) |
| twm2  | 7.97 (8.17)   | 6.90 (7.35)   | 5.55 (6.70)   | 4.28 (4.83)   |
| wm1   | 57.03 (15.00) | 51.50 (15.11) | 43.59 (15.17) | 32.80 (14.42) |
| wm2   | 46.34 (13.94) | 40.79 (13.70) | 33.31 (13.29) | 23.94 (11.92) |
| zem   | 59.85 (11.90) | 54.14 (12.11) | 46.29 (12.71) | 34.69 (11.89) |

| (c) Macro-averaged F <sub>1</sub> (and standard deviation) |               |               |               |               |
|--|---------------|---------------|---------------|---------------|
| System   | sr=0.1        | sr=0.3        | sr=0.5        | sr=0.7        |
| aapi   | 35.70 (15.04) | 34.12 (14.65) | 31.81 (13.44) | 28.27 (12.73) |
| dbb  | 37.12 (10.83) | 33.40 (10.58) | 28.29 (9.32)  | 21.48 (8.34)  |
| oc   | 31.73 (13.94) | 30.24 (13.31) | 27.68 (12.53) | 26.05 (11.11) |
| twm2   | 6.66 (9.34)   | 6.47 (8.65)   | 6.04 (8.09)   | 5.45 (6.35)   |
| wm1  | 57.17 (14.04) | 54.00 (13.68) | 48.60 (13.61) | 40.58 (14.55) |
| wm2  | 50.82 (13.11) | 47.00 (12.51) | 40.96 (12.09) | 32.59 (12.32) |
| zem  | 56.07 (11.95) | 53.14 (11.36) | 48.61 (10.72) | 40.57 (11.20) |

different seed ratios. The first thing that is noticeable is that all systems behave similarly when changing the seed ratio. This is expected as these tools do not employ the reference set of seed entities to aid disambiguation. The recall level is quite stable, as demonstrated by Table 5.10a, because as the seed ratio is increased, the remaining set of valid targets is smaller but the predictions made by the systems do not change. Table 5.10b shows that precision decreases for all the tools when the seed ratio is increased. This effect is due to most targets being used as seeds which leaves fewer valid targets for evaluating precision. Formally, the precision of system  $s$  at seed ratio  $sr_a$  for a document is  $P_{sr_a}^s = \frac{NC_{sr_a}}{N}$ , where  $N$  is the number of predictions made by the system and  $NC_{sr_a}$  is the average number of correct answers. As  $sr$  is increased,  $NC_{sr_a}$  can only decrease or stay unchanged. As  $sr$  approaches 1,  $P_{sr}^s$  reaches zero regardless of system. However, for the purposes of densification, precision calculated as above is ill-suited for assessing performance due to lack of data: systems which produce relevant/correct targets that are not in the original set are penalised. Recall is more stable relative to the seed ratio; while variations do occur, these are small in magnitude. A smaller the set of valid targets means a smaller denominator in the recall formula. The random selection of seeds does impact the actual values, which is why the results are averaged over ten runs.

The most curious behaviour is displayed by thewikimachine tool, which consistently has the lowest performance. This is probably because it tries to disambiguate simple terms using specially trained SVM classifiers, rather than focus on encyclopaedic named entities as seems to be the case with all other tools. This data can be used for a rough comparison of wikification performance, e.g., by examining the results for the minimum seed ratio. Overall, WikipediaMiner reaches the highest recall values, while Zemanta API edges ahead in terms of precision. The differences between these two systems, *wm1* and *zem*, are not statistically significant according to both the paired t-test and Wilcoxon signed rank test. This experiment was not designed to be a benchmark for these tools, but rather to provide insights into understanding performance when only using a subset of the valid targets.

A full comparison between the results obtained by the wikification tools presented

Table 5.11: Densification: results for Wikipedia markup evaluation

| System             | Recall | Precision | (upper bound) |
|--------------------|--------|-----------|---------------|
| WikipediaMiner 1   | 46.60% | 51.07%    | 100.00%       |
| Densification(1:1) | 58.73% | 57.26%    | 100.00%       |
| Densification(2:1) | 71.27% | 24.42%    | 50.00%        |
| Densification(3:1) | 83.23% | 14.51%    | 33.33%        |
| Densification(4:1) | 87.13% | 9.62%     | 25.00%        |

in Table 5.10 and the densification method proposed here is not possible because they address different tasks. However, for a more complete picture, the results of the densification method are estimated in a similar framework as wikification. Table 5.11 shows the results achieved by the densification system when seed ratio is 0.5, in comparison with Wikipedia Miner 1 which performed best amongst the other tools. In this setting, the number of links used as seeds is equal to the number of reference targets which are used to estimate recall. Four different threshold values were used to specify the ratio between the number of topics predicted by densification and the number of targets sought. This shows how recall increases when more densified links are considered, but also illustrates that this experiment is ill suited for estimating precision: considering more links artificially limits precision, only because these links, even if correct, are not present in the set of reference targets. In terms of recall, the densification method performs very well: the average recall is 71.27% when maximum precision is 50% (the number of predicted topics is twice the size of the target set). Average recall increases to 83.23% when maximum precision is 33.33%, and reaches 87.13% for maximum precision 25%.

### 5.5.3 Densification impact on EQUAL

In the CLEF competitions, EQUAL demonstrated the highest performance amongst all submissions, regardless whether they were automatic or they had human involvement, and it retrieved more correct answers than any other participant. An important performance bottleneck for the system was the limited types of semantic constraints: for some questions, although phrased in a simple form, the system did not have the

representation power to accurately interpret the question. The interpreter, despite having few, generic components, managed to return answers for half the questions. As new types of semantic constraints are added and more constraint validators are included in the system, increasingly large amounts of data are necessary to rank the most likely interpretations for the questions involved. This requires a coordinated community effort, especially due to the high degree of interdisciplinary knowledge involved, and also due to the amount of data and resources necessary.

Densification is aimed to address the main performance bottleneck that affects the questions for which EQUAL found at least a *compatible* interpretation. In many such cases, the QA system did not find all of the answers when exploring the Wikipedia link graph, because the constraint verifiers employed depended on wiki links. However, these are meant for human readers, and are used accordingly. In cases where some of the correct answers were overlooked, this was usually due to ‘missing links’. As argued at the beginning of the chapter, the available entity linking tools are focused on prominent named entities. This motivated the research into better linking algorithms, which are able to boost recall. This section focuses on the evaluation of the impact that densification brings to the semantic QA approach employed by EQUAL.

The evaluation is performed on the GikiCLEF 2009 dataset with exactly the same question interpretations generated by the system. This process can provide answers to two questions: a) does the Wikipedia link graph enhanced by densification improve recall? b) how does it affect precision? Results are compared to those achieved by the original system.

EQUAL primarily uses two graph-exploration steps: a) visiting out-links (the entities mentioned in an article) and b) visiting in-links (the articles mentioning an entity). Too much computational effort is necessary to densify the entire Wikipedia beforehand, therefore only those articles which are relevant to each question were processed. It is straightforward to densify an article before collecting its out-links. To augment the set of

in-links for a given entity  $e$ , the inverted index built by EQUAL using Lucene<sup>11</sup> (Hatcher and Gospodnetic, 2004) is used to retrieve candidate documents as follows: for each known anchor  $a$  all articles that contain  $a$  and do not contain a link to  $e$  are retrieved. To further reduce redundant computations, the documents which have a link to known targets  $t_a$  of anchor  $a$  are filtered out because densification would use this target as a seed.

In its original GikiCLEF submission, EQUAL returned 69 correct, 10 unjustified and 59 incorrect answers on the English dataset (see Table 4.6) out of an answer pool containing 118 validated answers. This achieves a micro-averaged precision of 57.24% and a micro-averaged recall of 66.94% relative to the official answers. After running densification, the system retrieves 78 novel results, containing 23 valid answers, more than half of the missing answers. Not all of the additional 55 results retrieved are necessarily incorrect because GikiCLEF used open ended questions. Overall, recall is increased to 86.44% (+19.5%) and precision falls to 47.22% (-10.02%). This suggests that densification does indeed help find additional answers, but the semantic constraints need to become more precise. For example, for the GC09-09 topic *Name places where Goethe fell in love*, the system does not find additional correct answers, but instead retrieves more incorrect ones because the constraint verifier is too generic: it validates all cities that are mentioned in the same paragraph as the trigger word “love”.

Densification is more computationally intensive than WikipediaMiner which achieves the state-of-the-art performance for wikification. The same evaluation was employed to determine how many answers can be found if using WM to recover additional links in Wikipedia articles. In this case, EQUAL finds only 34 additional results, containing 12 valid answers. Wikification also helps find additional answers, but to a lesser extent, as it usually created less links per document than densification (see results in Table 5.12).

Based on these results, densification does help increase recall, but, in the case of EQUAL, it does so at the cost of precision. To fully benefit from the extra information, the QA

---

<sup>11</sup><http://lucene.apache.org/core/>

Table 5.12: Densification: performance impact on EQUAL(micro-averaged)

| System          | Corr. | Incorr. | Total | P             | R             | F <sub>0.5</sub> | F <sub>1</sub> | F <sub>2</sub> |
|-----------------|-------|---------|-------|---------------|---------------|------------------|----------------|----------------|
| EQUAL           | 79    | 59      | 138   | <b>0.5725</b> | 0.6695        | 0.5896           | 0.6172         | 0.6475         |
| +Densification  | 102   | 114     | 216   | 0.4722        | <b>0.8644</b> | 0.5193           | 0.6108         | 0.7413         |
| +WikipediaMiner | 91    | 81      | 172   | 0.5291        | 0.7712        | 0.5645           | 0.6276         | 0.7065         |

system needs to use more advanced constraints and to filter candidate answers using improved information extraction techniques. It was decided not to implement additional constraints in EQUAL as it would have made the results not directly comparable. The set of answers which are not reachable by the current system suggests that, in addition to exploring Wikipedia based on links, the system should also use information extraction methods to search for supporting information in any article’s text. This enables the use of multi-argument constraints which require all arguments to be present in the same context, and not just in the same article/paragraph.

## 5.6 Conclusions

This chapter proposed a new method for identifying and ranking Wikipedia entities mentioned in a text. Experiments show that this method can improve the recall of the semantic QA system EQUAL, by enriching the link graph with additional links that are not present in the original markup of Wikipedia articles. The method uses a set of given seed topics to produce a robust ranking of relevant topics. It consists of two steps: pruning – which creates a high-recall set of candidates, and selection – which filters this set and computes relevance scores.

The first step, pruning, is a fuzzy disambiguation approach based on point-wise ranking models trained on data extracted from Wikipedia. It uses a semantic relatedness measure between two articles (Milne and Witten, 2008a) to identify and discard candidate topics that have little relevance to the initial set of seeds. In the second step, selection, the relevance of a topic is estimated using both the seeds and the remaining candidate topics which are mentioned nearby. Overlapping mentions are also dealt with. As there is no

gold standard dataset, an extrinsic validation approach is used. The method is able to produce robust results, even if only a tenth of data is used for training.

A small-scale experiment was carried out to determine whether or not human agreement was high enough to warrant the annotation of a gold standard for densification. The results demonstrated moderate agreement and suggested that acceptable agreement levels could be possible, provided adequate guidelines are given to the raters and that training is first carried out. The experiment showed that assessing relatedness is a rather subjective task, therefore, instead of a costly annotation effort, a more effective way to measure performance is via extrinsic evaluation. The rating methodology developed can be used to estimate the precision of densification models using crowd-sourcing.

Another experiment was carried out to measure the performance of the densification method using Wikipedia. For each article, a random set of links were considered as seeds while the remainder were used to estimate recall. This setting resembles that of wikification, so the performance was compared against several other wikification systems. On average, the method proposed achieves the best results. The advantage of this method is that it can be run on a large scale with no annotation required, but its disadvantage is that it cannot be used to estimate precision for densification.

Finally, the densification method was applied to the subset of pages used by EQUAL to answer GikiCLEF questions. The micro-averaged recall has increased from 66.94% to 86.44%, suggesting that densification enables the semantic QA system to find correct answers which it had previously missed. The decrease in micro-averaged precision from 57.24% to 47.22% suggest that the semantic components of the system need to be improved to take full advantage of the additional links. To check if a relationship between two entities holds, EQUAL uses either facts from sources such as DBpedia or checks for wiki links between the two corresponding Wikipedia articles. Densification allows the system to recover additional links which were omitted in the original markup, increasing the recall of the QA system. To find the remainder of missing answers, future work will apply densification on entire document collections (including Wikipedia) enabling EQUAL to

use paragraphs where both entities are mentioned as possible evidence for the relationship. This approach could also be employed as a pre-processing step to populate linked open data sets. By disambiguating entity mentions it can increase the precision of existing relation extraction tools.



# Chapter 6

## Concluding Remarks

The main aim of this thesis was to advance open-domain question answering to enable more complex questions to be addressed than currently possible by proposing a paradigm shift from textual to semantic QA systems and by providing a proof-of-concept implementation. This chapter considers the extent to which this has been achieved, by reviewing the thesis and summarising the primary results. Section 6.1 revisits the goals set out in Chapter 1 and Section 6.2 presents the main contributions of the thesis resulting from the achievement of these goals. Section 6.3 provides an overview by summarising each chapter, and Section 6.4 discusses possible future directions for the research.

### 6.1 Research goals revisited

This section revisits the goals presented in Chapter 1 which were necessary steps to fulfil the overall aim of the thesis: the advancement of open-domain QA in dealing with more complex question types than currently possible. Each goal is stated along with a description of how it was achieved.

**Goal 1** was to advance a paradigm shift in designing open-domain QA systems which can address more complex questions. This was achieved in Chapter 3 which revealed that standard textual QA systems are too focused on exploiting information redundancy for answering factoid questions, and argued that these systems are difficult to extend to cover more complex question types. Encyclopaedic questions were suggested as a QA challenge to foster new approaches, because these questions avoid the limitations of textual QA

which were identified in this thesis. A novel semantic architecture using concepts and knowledge representation as its core technology was then proposed. This architecture constitutes a paradigm shift in open-domain QA system design (which typically employs simple surface patterns and information retrieval engines). The architecture allows a QA system to combine information from different sources using simple forms of inference and affords a greater variety of question types, many of which are considered too difficult for current open-domain QA technologies. This approach is also motivated by Chapter 2 which illustrated the potential of Wikipedia for QA research, firstly as a repository of human knowledge and secondly as a central resource for a wide variety of NLP tools in the fields such as information retrieval, information extraction, knowledge representation, and semantic relatedness.

**Goal 2** was to test the viability of the novel approach through its implementation in a proof-of-concept QA system. Chapter 4 described EQUAL, the semantic QA system developed as part of this thesis, and presented its participation in two competitions which employed encyclopaedic questions: GikiP (Santos et al., 2008) and GikiCLEF (Santos and Cabral, 2009a). The system detects different types of ambiguity to decompose the question into constituents which are then assigned to coarse-grained semantic constraints involving entities, types, relations and properties. Instead of retrieving paragraphs, EQUAL explores the semantic graph induced by Wikipedia articles to determine correct answers by enforcing semantic constraints. In both competitions, EQUAL achieved the top results, significantly outperforming standard textual QA systems, proving that the QA architecture proposed in this thesis is able to deal with complex questions.

**Goal 3** was to develop a method which enables QA systems to add new arbitrary texts to its knowledge base by linking named entity mentions to relevant Wikipedia articles. This semantic analysis step is necessary because semantic QA systems cannot otherwise make use of text documents to expand their information coverage. The goal was achieved in Chapter 5 in which a new semantic analysis method was proposed, densification, which aims to produce a more complete set of links to relevant articles than existing approaches

which exhibit a strong bias towards precision rather than completeness. This method was evaluated by applying it to a subset of Wikipedia articles relevant for the GikiCLEF dataset. The new links identified via densification enabled EQUAL to find additional correct answers.

## 6.2 Original contributions

The achievement of the goals described above allowed this thesis to make three key original contributions to research in the field of question answering.

The **first original contribution** of this work is the proposal of a paradigm shift in open-domain QA research. This thesis argued that the standard textual approach is inherently limited to questions whose answers are explicitly present in the reference document collection used. To foster the development of more advanced approaches, the thesis proposed to change the focus of QA research from textual, factoid questions to more complex question types, such as encyclopaedic questions: open-list questions usually composed of several constraints that require a system to aggregate evidence from more than one document or data source and to validate each of the possible answers individually. The thesis then advocated a paradigm shift in the design of open-domain QA systems by moving away from the textual approach which uses words and information retrieval at the core of the QA architecture, towards a semantic approach which uses atomic facts and semi-structured knowledge bases. This affords simple forms of inference, enabling the system to combine information from different sources and to answer more complex questions. To achieve this, a novel architecture for semantic QA systems was proposed, which can address encyclopaedic questions. The architecture contains two main processing phases: *analysis*, responsible for interpreting questions, identifying ambiguities and finding answers, and *feedback*, responsible for facilitating effective interaction with the user.

The **second original contribution** of this work is the development of EQUAL, a new semantic QA system implementing the analysis phase of the novel architecture. It uses

Wikipedia as source of world knowledge and employs simple forms of open-domain inference to answer encyclopaedic questions. EQUAL decomposes a question into a sequence of semantic constraints by combining the output of a syntactic parser with information from Wikipedia. To address natural language ambiguity, the system builds several formal interpretations which contain the constraints identified in the question, and addresses each interpretation separately. To find answers, the system then tests these constraints individually for each candidate answer, considering information from different documents and/or sources. The correctness of an answer is not proved using a logical formalism, instead a confidence-based measure is employed. This measure reflects the validation of constraints from raw natural language, automatically extracted entities, relations and available structured and semi-structured knowledge from Wikipedia and the Semantic Web. When searching for answers, EQUAL uses Wikipedia as an entity-relation semantic graph to find relevant pieces of information. This method affords good precision and allows only pages of a certain type to be considered, but is affected by the incompleteness of the existing markup which is targeted towards human readers. The viability of the proposed approach was demonstrated by EQUAL's participation in two competitions, GikiP at CLEF 2008 and GikiCLEF at CLEF 2009. The system outperformed both standard textual QA systems and semi-automatic approaches.

The **third original contribution** of the thesis is the development of a new semantic analysis method, densification, which transforms raw text into semi-structured information by linking mentions of entities or terms to relevant Wikipedia articles. To answer questions using new document collections, the semantic QA approach employed in this thesis requires that texts are first connected to the entity-relationship graph extracted from Wikipedia, which constitutes the core knowledge base used by the system. The GikiCLEF results showed that the existing markup of wiki links in Wikipedia is incomplete, to avoid redundant information and clutter, and as a result a system needs to automatically recover additional links which were omitted due to stylistic considerations. A review of existing wikification tools revealed their bias towards a few precise links rather than a more complete set of links as needed by QA systems.

In order to address this, a semantic analysis method which disambiguates entities is developed to enrich Wikipedia articles with additional links to other pages. The module increases recall, enabling the system to rely more on the link structure of Wikipedia than on word-based similarity between pages. It also allows new document collections to be linked to the encyclopaedia, further enhancing the coverage of the system. Applying the densification method to the GikiCLEF dataset allowed EQUAL to find additional answers which it had previously missed, proving that the task is useful. However, the number of incorrect answers also increased suggesting that the system needs more accurate semantic constraint verifiers to better filter the additional answer candidates.

### 6.3 Directions for future work

The paradigm shift proposed in this thesis opens up an entirely new array of research avenues, either to improve the semantic QA architecture, e.g., further enhance the types of inference possible and increase the complexity of questions addressed, or to extend an actual QA system such as EQUAL by incorporating new tools, resources and technologies.

In terms of the novel QA architecture proposed in Chapter 3, one of the most obvious directions for future investigation is the second phase: **feedback**. This research falls mainly in the areas of human-computer interfaces, interactive question answering and dialogue systems. As envisioned in Section 3.5, a system needs to be able to synthesise, in very short amounts of time, informative responses which limit the impact of misunderstandings. The interface should explain the interpretation of the question, allow the user to quickly check the facts or rules that were used by the system to validate individual answers, and enable the user to easily change/adjust their question. To achieve these objectives, novel inference engines need to be employed that can deal with uncertain facts, multiple-source data provenance and probabilistic reasoning. This is currently an active research area, but no ready-made solutions yet exist. Presentation of the results also warrants exploration, for example how to combine visual elements with natural language generation algorithms to present the information efficiently. An important feature that

the interface should accommodate is that of automatically collecting evaluation of the system's performance by analysing user actions, without asking for explicit validation assessments. This data is instrumental for improving performance.

Determining how to map semantic constraints to corresponding verifiers poses several problems and deserves further investigation. Very similar constraints might require different verifiers, for example, *Romanian architects* can be matched to the homonym category in Wikipedia, but *Romanian volcanoes* might need to be decomposed into two atomic constraints, e.g., *?x isA volcano* and *?x locatedIn Romania*. This mapping affects consistency, when different verifiers yield contradictory results, but more importantly it also affects latency, the processing time required to apply verifiers and generate an initial response (Nielsen, 2007). Some verifiers are costly in terms of the necessary processing time, such as generic verifiers, which need to examine text in search for facts at query time, or nested verifiers, which need to simultaneously check several facts. In the case of complex questions, the system should plan the order in which verifiers are checked so that the more restrictive filters are applied first. In addition, the system could use pre-verifiers, necessary but insufficient conditions that can be tested efficiently, to narrow down the list of candidate answers which require further processing. Another approach for reducing latency is to rank the likelihood of each question interpretation. This requires the system to combine confidence scores for the different disambiguations it performs when interpreting a question. The resulting likelihood score can be particularly useful to prepare the response to the user: the most likely interpretations should be prioritised and their results displayed first, while less likely interpretations can be processed either while the user examines the initial response, or only after the user explicitly indicates it.

More specifically for enhancing systems such as EQUAL, one challenge is the mining of large amounts of semi-structured information from a densified version of Wikipedia. This would allow the creation of virtual, synthetic categories that simplify the expected answer type resolution, for example creating the virtual category *German Renaissance composers* as the intersection of the categories *German people* and *Renaissance composers*

can benefit both interpreting the question and validating candidate answers. The vast amounts of semantically annotated data that can be obtained via densification of the entire Wikipedia would also allow the system to autonomously determine which are the most frequent relations connecting entities of a given type, e.g., footballers and clubs, how these are expressed in text and how to link them to Linked Open Data datasets, extending the coverage of its semantic knowledge base as well as enhancing question interpretation capabilities.

One of main challenges facing the novel approach proposed in Chapter 3 is correctly interpreting the constraints present in questions, which requires the combination of several research areas: syntactic parsing, semantic parsing and automatic reasoning. EQUAL has been successfully applied to encyclopaedic questions, which are a conjunction of atomic constraints. Future research should enhance the capability to interpret and answer questions which use quantifiers and aggregates, in questions such as *Which rivers flow through more than 3 national capitals?*

GikiCLEF demonstrated that different language versions of Wikipedia have different information coverage (Santos and Cabral, 2009a), for example, there is more factual information about Brazil-related topics in the Portuguese version of Wikipedia than in the English version. Another challenge for EQUAL is to further exploit the multilingual nature of Wikipedia. In a first stage, the semantic graphs of different language versions can be merged, to add more links between entities or to collect additional categories. In a second stage, multilingual constraint verifiers can be developed which employ facts originating from data sources written in a language different than that of the question.

A promising research avenue is to further develop densification as a general purpose tool for semantic document analysis and employ it in different NLP applications. One direction previously mentioned is that of incrementally applying densification to Wikipedia itself, further enriching its semantic structure. Such a resource could have significant impact in several fields, being especially well suited for semi-supervised

methods. Another direction is to tailor the behaviour of the algorithm for use in specific NLP applications which can take advantage of the semantic information available.

# References

- Ahn, D., Jijkoun, V., Mishne, G., Müller, K., de Rijke, M. and Schlobach, S. (2005), Using Wikipedia at the TREC QA Track, *in* E. M. Voorhees and L. P. Buckland, eds, ‘Proceedings of the Thirteenth Text REtrieval Conference TREC 2004’, Vol. Special Publication 500-261, National Institute of Standards and Technology (NIST).
- Allan, J. (2005), HARD Track Overview in TREC 2005: High Accuracy Retrieval from Documents, *in* E. M. Voorhees and L. P. Buckland, eds, ‘Proceedings of the Fourteenth Text REtrieval Conference, TREC 2005’, Vol. Special Publication 500-266, National Institute of Standards and Technology (NIST).
- Alpert, J. and Hajaj, N. (2008), ‘We knew the web was big’, <http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html>.
- Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R. and Ives, Z. (2008), DBpedia: A Nucleus for a Web of Open Data, *in* ‘Proceedings of 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference (ISWC+ASWC 2007)’, Lecture Notes in Computer Science, Springer, pp. 722–735.
- Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M. and Etzioni, O. (2007), Open Information Extraction from the Web, *in* M. M. Veloso, ed., ‘Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)’, AAAI Press, Menlo Park, California, pp. 2670–2676.
- Bentivogli, L., Forner, P., Giuliano, C., Marchetti, A., Pianta, E. and Tymoshenko, K. (2010), Extending English ACE 2005 Corpus Annotation with Ground-truth Links to Wikipedia, *in* ‘Proceedings of the 2nd Workshop on The People’s Web Meets NLP: Collaboratively Constructed Semantic Resources’, Coling 2010 Organizing Committee, Beijing, China, pp. 19–27.
- Berners-Lee, T., Cailliau, R., Groff, J.-F. and Pollermann, B. (1992), ‘World-wide web: The information universe’, *Internet Research* 2(1), 52–58.
- Bouma, G. and Duarte, S. (2009), Wikipedia entity retrieval for Dutch and Spanish, *in* C. Peters et al., eds, ‘Cross Language Evaluation Forum: Working Notes for CLEF 2009’, Corfu, Greece.
- Bryl, V., Giuliano, C., Serafini, L. and Tymoshenko, K. (2010), Supporting natural language processing with background knowledge: Coreference resolution case, *in* P. F. Patel-Schneider, Y. Pan, P. Hitzler, P. Mika, L. Zhang, J. Z. Pan, I. Horrocks and B. Glimm, eds, ‘The Semantic Web — ISWC 2010: 9th International Semantic Web

Conference, Shanghai, China, Revised Selected Papers, Part I', Vol. 6496 of *Lecture Notes in Computer Science*, Springer, Berlin, pp. 80–95.

Bunescu, R. C. (2007), Learning for Information Extraction: From Named Entity Recognition and Disambiguation To Relation Extraction, PhD thesis, University of Texas at Austin.

Bunescu, R. C. and Pasca, M. (2006), Using Encyclopedic Knowledge for Named Entity Disambiguation, in 'Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06)', Association for Computational Linguistics, Trento, Italy, pp. 9–16.

Burger, J., Cardie, C., Chaudhri, V., Gaizauskas, R., Harabagiu, S., Israel, D., Jacquemin, C., Lin, C.-Y., Maiorano, S., Miller, G., Moldovan, D., Ogden, B., Prager, J., Riloff, E., Singhal, A., Shrihari, R., Strzalkowski, T., Voorhees, E. and Weishedel, R. (2000), Issues, Tasks and Program Structures to Roadmap Research in Question & Answering (Q&A), Technical report, National Institute of Standards and Technology (NIST).

Cabrio, E., Kouylekov, M., Magnini, B., Negri, M., Hasler, L., Orasan, C., Tomas, D., Vicedo, J. L., Neumann, G. and Weber, C. (2008), The QALL-ME Benchmark: a Multilingual Resource of Annotated Spoken Requests for Question Answering, in N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odjik, S. Piperidis and D. Tapias, eds, 'Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)', European Language Resources Association (ELRA), Marrakech, Morocco, pp. 2525–2530.

Cardoso, N. (2008), Rembrandt - reconhecimento de entidades mencionadas baseado em relações e análise detalhada do texto, in 'Encontro do Segundo HAREM, PROPOR 2008', Aveiro, Portugal.

Cardoso, N., Batista, D., Lopez-Pellicer, F. J. and Silva, M. J. (2009), Where in the Wikipedia is that answer? the XLDB at the GikiCLEF 2009 task, in C. Peters et al., eds, 'Cross Language Evaluation Forum: Working Notes for CLEF 2009', Corfu, Greece.

Cardoso, N., Dornescu, I., Hartrumpf, S. and Leveling, J. (2010), Revamping question answering with a semantic approach over world knowledge, in 'Multiple Language Question Answering 2010 (MLQA10), CLEF LABs 2010'.

Chaves, M. S., Silva, M. J. and Martins, B. (2005), A Geographic Knowledge Base for Semantic Web Applications, in C. A. Heuser, ed., 'Proceedings of the 20th Brazilian Symposium on Databases', Brazil, pp. 40–54.

- Chen, A. and Gey, F. C. (2004), 'Multilingual information retrieval using machine translation, relevance feedback and compounding', *Information Retrieval* 7(1-2), 149–182.
- Chu-Carroll, J., Czuba, K., Prager, J. and Ittycheriah, A. (2003), In question answering, two heads are better than one, in 'NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology', Association for Computational Linguistics, Morristown, NJ, USA, pp. 24–31.
- Chu-Carroll, J., Duboué, P. A., Prager, J. M. and Czuba, K. (2005), IBM's piquant ii in trec 2005, in E. M. Voorhees and L. P. Buckland, eds, 'Proceedings of the Fourteenth Text REtrieval Conference, TREC 2005', National Institute of Standards and Technology (NIST).
- Chu-Carroll, J., Fan, J., Boguraev, B. K., Carmel, D., Sheinwald, D. and Welty, C. (2012), 'Finding needles in the haystack: Search and candidate generation', *IBM Journal of Research and Development* 56(3.4), 6:1 –6:12.
- Clarke, C. L. A., Cormack, G. V. and Lynam, T. R. (2001), Exploiting redundancy in question answering, in 'Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval', ACM Press, pp. 358–365.
- Cohen, J. (1960), 'A coefficient of agreement for nominal scales', *Educational and Psychological Measurement* 20(1), 37–46.
- Cooper, W. S., Gey, F. C. and Dabney, D. P. (1992), Probabilistic retrieval based on staged logistic regression, in 'Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval', SIGIR '92, ACM, New York, NY, USA, pp. 198–210.
- Cucerzan, S. (2007), Large-Scale Named Entity Disambiguation Based on Wikipedia Data, in 'Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)', Association for Computational Linguistics, Prague, Czech Republic, pp. 708–716.
- Damljanovic, D., Agatonovic, M. and Cunningham, H. (2010), Natural language interfaces to ontologies: Combining syntactic analysis and ontology-based lookup through the user interaction., in L. Aroyo, G. Antoniou, E. Hyvönen, A. ten Teije, H. Stuckenschmidt, L. Cabral and T. Tudorache, eds, 'ESWC 2010', Vol. 6088 of *Lecture Notes in Computer Science*, Springer, pp. 106–120.

- Damljanovic, D., Agatonovic, M. and Cunningham, H. (2011), FREyA: an Interactive Way of Querying Linked Data using Natural Language, *in* ‘Proceedings of 1st Workshop on Question Answering over Linked Data (QALD-1), Collocated with the 8th Extended Semantic Web Conference (ESWC 2011)’, Heraklion, Greece.
- d’Aquin, M., Motta, E., Sabou, M., Angeletou, S., Gridinoc, L., Lopez, V. and Guidi, D. (2008), ‘Toward a new generation of semantic web applications’, *IEEE Intelligent Systems* 23(3), 20–28.
- de Vries, A. P., Vercoustre, A.-M., Thom, J. A., Craswell, N. and Lalmas, M. (2008), Overview of the INEX 2007 Entity Ranking Track, *in* ‘Focused Access to XML Documents: 6th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2007. Selected Papers’, Springer-Verlag, Berlin, Heidelberg, pp. 245–251.
- Diekema, A., Yilmazel, O., Chen, J., Harwell, S., He, L. and Liddy, E. D. (2004), Finding answers to complex questions, *in* Maybury (2004b), pp. 141–152.
- Dornescu, I. (2009), EQUAL: Encyclopaedic QQuestion Answering for Lists, *in* F. Borri, A. Nardi and C. Peters, eds, ‘Working Notes for the CLEF 2009 Workshop’, CLEF 2009 Organizing Committee, Corfu, Greece.
- Dornescu, I. (2010), Semantic QA for encyclopaedic questions: EQUAL in GikiCLEF, *in* C. Peters, G. M. Di Nunzio, M. Kurimo, T. Mandl and D. Mostefa, eds, ‘Multilingual Information Access Evaluation I. Text Retrieval Experiments’, Vol. 6241 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Heidelberg, pp. 326–333.
- Dornescu, I., Puşcaşu, G. and Orăsan, C. (2008), University of Wolverhampton at QA@CLEF, *in* F. Borri, A. Nardi and C. Peters, eds, ‘Working Notes for the CLEF 2008 Workshop’, CLEF 2008 Organizing Committee, Aarhus, Denmark.
- Echihabi, A., Hermjakob, U., Hovy, E., Marcu, D., Melz, E. and Ravichandran, D. (2004), How to select an answer string?, *in* T. Strzalkowski and S. Harabagiu, eds, ‘Advances in Textual Question Answering’, Springer, Dordrecht, Netherlands, pp. 383–406.
- Fellbaum, C., ed. (1998), *WordNet: An Electronic Lexical Database*, MIT Press, Cambridge, MA.
- Ferrucci, D. A. (2012), ‘Introduction to "This is Watson"’, *IBM Journal of Research and Development* 56(3.4), 1:1 –1:15.
- Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., Lally, A., Murdock, J. W., Nyberg, E., Prager, J., Schlaefel, N. and Welty, C. (2010), ‘Building Watson: An Overview of the DeepQA Project’, *AI Magazine* 31(3), 59–79.

- Fleiss, J. L. (1971), 'Measuring nominal scale agreement among many raters', *Psychological Bulletin* 76(5), 378–382.
- Gabrilovich, E. and Markovitch, S. (2007), Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis, in 'Proceedings of The Twentieth International Joint Conference for Artificial Intelligence', Hyderabad, India, pp. 1606–1611.
- Giles, J. (2005), 'Internet encyclopaedias go head to head', *Nature* 438(7070), 900–901.
- Gondek, D. C., Lally, A., Kalyanpur, A., Murdock, J. W., Duboue, P. A., Zhang, L., Pan, Y., Qiu, Z. M. and Welty, C. (2012), 'A framework for merging and ranking of answers in deepqa', *IBM Journal of Research and Development* 56(3.4), 14:1 –14:12.
- Green, B. F., Wolf, A. K., Chomsky, C. and Laughery, K. (1986), Baseball: An Automatic Question Answerer, in B. J. Grosz, K. Sparck Jones and B. L. Webber, eds, 'Natural Language Processing', Kaufmann, Los Altos, CA, pp. 545–549.
- Green, B., Wolf, A., Chomsky, C. and Laughery, K. (1961), BASEBALL: an automatic question answerer, in 'Proceedings Western Joint Computer Conference', Vol. 19, pp. 219–224.
- Hammwöhner, R. (2007), Interlingual Aspects Of Wikipedia's Quality, in B. Klein, M. L. Markus and M. A. Robbert, eds, 'Proceedings of the 12th International Conference on Information Quality', p. 39–49.
- Harabagiu, A., Lacatusu, F. and Hickl, A. (2006), Answering Complex Questions with Random Walk Models, in 'Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval', Vol. 2006, New York, NY, USA, pp. 220–227.
- Harabagiu, S. M., Maiorano, S. J. and Pasca, M. A. (2003), 'Open-domain textual question answering techniques', *Natural Language Engineering* 9(03), 231–267.
- Harabagiu, S. and Moldovan, D. (2003), Question Answering, in R. Mitkov, ed., 'The Oxford Handbook of Computational Linguistics', Oxford University Press, chapter 31, pp. 560–582.
- Harabagiu, S., Moldovan, D., Paşca, M., Mihalcea, R., Surdeanu, M., Girju, R., Rus, V., Morarescu, P. and Bunescu, R. (2000), FALCON: boosting knowledge for answer engines, in E. Voorhees and D. Harman, eds, 'Proceedings 9th TExt Retrieval Conference (TREC-9)', National Institute of Standards and Technology (NIST), Gaithersburg, MD, pp. 479–488.

- Harabagiu, S., Moldovan, D., Pasca, M., Mihalcea, R., Surdeanu, M., Girju, R., Rus, V., Lacatusu, F., Morarescu, P. and Bunescu, R. (2001), Answering complex, list and context questions with LCC's Question-Answering Server, *in* E. M. Voorhees and L. P. Buckland, eds, 'Proceedings 10th TExt Retrieval Conference (TREC-10)', National Institute of Standards and Technology (NIST), Gaithersburg, MD, pp. 355–361.
- Hartrumpf, S. (2003), *Hybrid Disambiguation in Natural Language Analysis*, Der Andere Verlag, Osnabrück, Germany.
- Hartrumpf, S. (2005), Question answering using sentence parsing and semantic network matching, *in* 'Proceedings of the 5th conference on Cross-Language Evaluation Forum: multilingual Information Access for Text, Speech and Images', CLEF'04, Springer-Verlag, Berlin, Heidelberg, pp. 512–521.
- Hartrumpf, S. (2008), Semantic decomposition for question answering, *in* 'Proceedings of the 2008 conference on ECAI 2008: 18th European Conference on Artificial Intelligence', IOS Press, Amsterdam, The Netherlands, The Netherlands, pp. 313–317.
- Hartrumpf, S., Glöckner, I. and Leveling, J. (2009), Efficient question answering with question decomposition and multiple answer streams, *in* 'Evaluating Systems for Multilingual and Multimodal Information Access, CLEF 2008', Vol. 5706 of *Lecture Notes in Computer Science*, Springer, pp. 421–428.
- Hartrumpf, S. and Leveling, J. (2009), GIRSA-WP at GikiCLEF: Integration of structured information and decomposition of questions, *in* 'Proceedings of the 10th cross-language evaluation forum conference on Multilingual information access evaluation: text retrieval experiments', *Lecture Notes in Computer Science (LNCS)*, Springer-Verlag, Berlin, Heidelberg.
- Hartrumpf, S. and Leveling, J. (2010), Recursive question decomposition for answering complex geographic questions, *in* C. Peters, G. M. Di Nunzio, M. Kurimo, T. Mandl and D. Mostefa, eds, 'Multilingual Information Access Evaluation I. Text Retrieval Experiments', Vol. 6241 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Heidelberg, pp. 310–317.
- Hatcher, E. and Gospodnetic, O. (2004), 'Lucene in action', *Analysis* 54, 258.
- Hermjakob, U. (2001), Parsing and question classification for question answering, *in* 'Proceedings of the workshop on Open-domain question answering', Association for Computational Linguistics, Morristown, NJ, USA, pp. 1–6.
- Hirschman, L. and Gaizauskas, R. (2001), 'Natural language question answering: the view from here', *Natural Language Engineering* 7(4), 275–300.

- Hovy, E., Gerber, L., Hermjakob, U., Junk, M. and Lin, C. (2000), Question answering in Webclopedia, in E. Voorhees and D. Harman, eds, 'Proceedings 9th Text Retrieval Conference (TREC-9)', National Institute of Standards and Technology (NIST), Gaithersburg, MD, pp. 655–664.
- Ji, H., Grishman, R., Dang, H. T., Griffitt, K. and Ellis, J. (2010), Overview of the tac 2010 knowledge base population track, in H. T. Dang, ed., 'Proceedings of the Third Text Analysis Conference (TAC 2010)', Gaithersburg, Maryland, National Institute of Standards and Technology, pp. 1–24.
- Jijkoun, V. and de Rijke, M. (2007), Overview of the WiQA Task at CLEF 2006, in '7th Workshop of the Cross-Language Evaluation Forum, CLEF 2006. Selected papers', Lecture Notes in Computer Science, Springer, pp. 265–274.
- Jijkoun, V., van Rantwijk, J., Ahn, D., Sang, E. F. T. K. and de Rijke, M. (2006), The University of Amsterdam at CLEF@QA 2006, in A. Nardi, C. Peters and J. Vicedo, eds, 'Working Notes CLEF 2006'.
- Katz, B. (1997), Annotating the world wide web using natural language, in L. Devroye and C. Christment, eds, 'Proceedings of the 5th RIAO Conference on Computer Assisted Information Searching on the Internet (RIAO '97)', pp. 136–159.
- Kazama, J. and Torisawa, K. (2007), Exploiting Wikipedia as External Knowledge for Named Entity Recognition, in 'Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)', Association for Computational Linguistics, Prague, Czech Republic, pp. 698–707.
- Klein, D. and Manning, C. D. (2003), Accurate unlexicalized parsing, in 'ACL', pp. 423–430.
- Ko, J., Si, L. and Nyberg, E. (2010), 'Combining evidence with a probabilistic framework for answer ranking and answer merging in question answering', *Information Processing & Management* 46(5), 541 – 554.
- Krippendorff, K. (2004), *Content Analysis: An Introduction to Its Methodology*, Vol. 79, Sage.
- Lally, A., Prager, J. M., McCord, M. C., Boguraev, B. K., Patwardhan, S., Fan, J., Fodor, P. and Chu-Carroll, J. (2012), 'Question analysis: How watson reads a clue', *IBM Journal of Research and Development* 56(3.4), 2:1 –2:14.
- Lang, K. (1995), Newsweeder: Learning to filter netnews, in 'Proceedings of the Twelfth International Conference on Machine Learning', pp. 331–339.

- Larson, R. R. (2009), Interactive probabilistic search for gikiclef, *in* ‘Proceedings of the 10th cross-language evaluation forum conference on Multilingual information access evaluation: text retrieval experiments’, CLEF’09, Springer-Verlag, Berlin, Heidelberg, pp. 334–341.
- Le Nguyen, M., Nguyen, T. T. and Shimazu, A. (2007), Subtree mining for question classification problem, *in* ‘IJCAI’07: Proceedings of the 20th international joint conference on Artificial intelligence’, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 1695–1700.
- Lenat, D. B. and Guha, R. V. (1989), *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*, 1st edn, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- Lenat, D. B. and Guha, R. V. (1991), ‘The evolution of CycL, the Cyc representation language’, *SIGART Bulletin* 2(3), 84–87.
- Leveling, J. and Hartrumpf, S. (2008), Advances in multilingual and multimodal information retrieval, Springer-Verlag, Berlin, Heidelberg, chapter Inferring Location Names for Geographic Information Retrieval, pp. 773–780.
- Li, X. and Roth, D. (2002), Learning question classifiers, *in* ‘Proceedings of the 19th international conference on Computational linguistics’, Association for Computational Linguistics, Morristown, NJ, USA, pp. 1–7.
- Li, Y., Luk, R. W. P., Ho, E. K. S. and Chung, K. F. (2007), Improving Weak Ad-hoc Queries using Wikipedia as External Corpus, *in* ‘Proceedings of SIGIR ’07’, ACM Press, p. 797–798.
- Lita, L. V., Hunt, W. A. and Nyberg, E. (2004), Resource analysis for question answering, *in* ‘The Companion Volume to the Proceedings of 42st Annual Meeting of the Association for Computational Linguistics’, Association for Computational Linguistics, Morristown, NJ, USA, pp. 162–165.
- Lopez, V., Motta, E. and Uren, V. (2006), Poweraqua: fishing the semantic web, *in* ‘Proceedings of the 3rd European conference on The Semantic Web: research and applications’, ESWC’06, Springer-Verlag, Berlin, Heidelberg, pp. 393–410.
- Lopez, V., Sabou, M. and Motta, E. (2006), Powermap: mapping the real semantic web on the fly, *in* ‘Proceedings of the 5th international conference on The Semantic Web’, ISWC’06, Springer-Verlag, Berlin, Heidelberg, pp. 414–427.

- Lopez, V., Uren, V., Motta, E. and Pasin, M. (2007), ‘AquaLog: An ontology-driven question answering system for organizational semantic intranets’, *Web Semantics: Science, Services and Agents on the World Wide Web* 5(2), 72–105.
- Lopez, V., Uren, V., Sabou, M. and Motta, E. (2011), ‘Is question answering fit for the semantic web?: a survey’, *Semantic web* 2(2), 125–155.
- Lopez, V., Uren, V., Sabou, M. R. and Motta, E. (2009), Cross ontology query answering on the semantic web: an initial evaluation, in ‘Proceedings of the fifth international conference on Knowledge capture’, K-CAP ’09, ACM, New York, NY, USA, pp. 17–24.
- Lynn, J. (2010), ‘Internet users to exceed 2 billion this year’, <http://www.reuters.com/article/2010/10/19/us-telecoms-internet-idUSTRE69I24720101019>.
- Maybury, M. T. (2004a), Question Answering: An Introduction, in *New Directions in Question Answering* Maybury (2004b), pp. 3–18.
- Maybury, M. T., ed. (2004b), *New Directions in Question Answering*, AAAI Press.
- McCord, M. C., Murdock, J. W. and Boguraev, B. K. (2012), ‘Deep parsing in watson’, *IBM Journal of Research and Development* 56(3.4), 3:1 –3:15.
- McGuinness, D. and van Harmelen, F. (2004), ‘OWL Web Ontology Language: Overview’, <http://www.w3.org/TR/owl-features/>.
- Medelyan, O., Milne, D., Legg, C. and Witten, I. H. (2009), ‘Mining meaning from Wikipedia’, *International Journal of Human-Computer Studies* 67(9), 716 – 754.
- Mendes, P. N., Jakob, M., García-Silva, A. and Bizer, C. (2011), DBpedia Spotlight: Shedding Light on the Web of Documents, in ‘Proceedings of the 7th International Conference on Semantic Systems (I-Semantics)’.
- Mihalcea, R. and Csomai, A. (2007), Wikify!: linking documents to encyclopedic knowledge, in ‘CIKM ’07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management’, ACM, New York, NY, USA, pp. 233–242.
- Milne, D. N., Witten, I. H. and Nichols, D. M. (2007), A knowledge-based search engine powered by Wikipedia, in ‘CIKM ’07: Proceedings of the sixteenth ACM conference on Conference on Information and Knowledge Management’, ACM Press, New York, USA, pp. 445–454.
- Milne, D. and Witten, I. (2009), ‘An Open-Source Toolkit for Mining Wikipedia’, <http://www.cs.waikato.ac.nz/~dnk2/publications/AnOpenSourceToolkitForMiningWikipedia.pdf>.

- Milne, D. and Witten, I. H. (2008a), An effective, low-cost measure of semantic relatedness obtained from wikipedia links, *in* 'Proceedings of the first AAAI Workshop on Wikipedia and Artificial Intelligence (WIKIAI'08)', Chicago, IL, pp. 25–30.
- Milne, D. and Witten, I. H. (2008b), Learning to link with Wikipedia, *in* J. G. Shanahan, S. Amer-Yahia, I. Manolescu, Y. Zhang, D. A. Evans, A. Kolcz, K.-S. Choi and A. Chowdhury, eds, 'Proceedings of the 17th ACM conference on Information and knowledge management (CIKM'2008)', ACM, New York, NY, USA, pp. 509–518.
- Mitkov, R. (2002), *Anaphora Resolution*, Studies in Language and Linguistics, Pearson Longman, Essex.
- Moldovan, D., Clark, C., Harabagiu, S. and Hodges, D. (2007), 'Cogex: A semantically and contextually enriched logic prover for question answering', *Journal of Applied Logic* 5(1), 49 – 69. Questions and Answers: Theoretical and Applied Perspectives.
- Mollá, D. and Vicedo, J. L. (2007), 'Question Answering in Restricted Domains: An Overview', *Computational Linguistics* 33(1), 41–61.
- Murdock, J. W., Kalyanpur, A., Welty, C., Fan, J., Ferrucci, D. A., Gondek, D. C., Zhang, L. and Kanayama, H. (2012), 'Typing candidate answers using type coercion', *IBM Journal of Research and Development* 56(3.4), 7:1 –7:13.
- Nielsen, J. (2007), 'Response Times: The 3 Important Limits', <http://www.useit.com/papers/responsetime.html>.
- Nyberg, E., Mitamura, T., Carbonell, J., Callan, J., Collins-thompson, K., Czuba, K., Duggan, M., Hiyakumoto, L., Hu, N., Huang, Y., Ko, J., Lira, L. V., Murtagh, S., Pedro, V. and Svoboda, D. (2003), The JAVELIN Question-Answering System at TREC 2003: A Multi-Strategy Approach with Dynamic Planning, *in* E. M. Voorhees and L. P. Buckland, eds, 'Proceedings of TREC 12', National Institute of Standards and Technology (NIST), Gaithersburg, MD.
- Peters, I. (2009), *Folksonomies. Indexing and Retrieval in Web 2.0*, De Gruyter [u.a.], Berlin.
- Pinto, D., Branstein, M., Coleman, R., Croft, W. B., King, M., Li, W. and Wei, X. (2002), Quasm: a system for question answering using semi-structured data, *in* 'JCDL '02: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries', ACM, New York, NY, USA, pp. 46–55.
- Ponzetto, S. P. and Strube, M. (2006), Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution, *in* R. C. Moore, J. A. Bilmes, J. Chu-Carroll and M. Sanderson, eds, 'Proceedings of the main conference on Human

- Language Technology Conference of the North American Chapter of the Association of Computational Linguistics’, Association for Computational Linguistics, The Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 192–199.
- Ponzetto, S. P. and Strube, M. (2007a), Deriving a Large-Scale Taxonomy from Wikipedia, *in* ‘Proceedings of the 22nd AAAI Conference on Artificial Intelligence’, Vol. 2, AAAI Press, pp. 1440–1445.
- Ponzetto, S. P. and Strube, M. (2007b), ‘Knowledge Derived From Wikipedia For Computing Semantic Relatedness’, *Journal Artificial Intelligence Resources (JAIR)* **30**, 181–212.
- Potthast, M., Stein, B. and Anderka, M. (2008), A Wikipedia-Based Multilingual Retrieval Model, *in* C. Macdonald, I. Ounis, V. Plachouras, I. Ruthven and R. W. White, eds, ‘30th European Conference on IR Research, ECIR 2008’, Vol. 4956 of *Lecture Notes in Computer Science*, Springer, pp. 522–530.
- Prager, J. M., Chu-Carroll, J., Czuba, K., Welty, C. A., Ittycheriah, A. and Mahindru, R. (2003), IBM’s PIQUANT in TREC2003, *in* E. M. Voorhees and L. P. Buckland, eds, ‘TREC’, National Institute of Standards and Technology (NIST), Gaithersburg, MD, pp. 283–292.
- Radev, D., Fan, W., Qi, H., Wu, H. and Grewal, A. (2005), ‘Probabilistic question answering on the Web’, *Journal of the American Society for Information Science and Technology* **56**(6), 408–419.
- Rizzo, G. and Troncy, R. (2011), NERD: Evaluating Named Entity Recognition Tools in the Web of Data, *in* ‘Proceedings of the ISWC’11 Workshop on Web Scale Knowledge Extraction (WEKEX’11)’, Bonn, Germany, pp. 1–16.
- Robertson, S. E. and Sparck Jones, K. (1988), Document retrieval systems, Taylor Graham Publishing, London, UK, UK, chapter Relevance weighting of search terms, pp. 143–160.
- Ruiz-Casado, M., Alfonseca, E. and Castells, P. (2006), From Wikipedia to Semantic Relationships: a Semi-automated Annotation Approach., *in* M. Völkel and S. Schaffert, eds, ‘Proceedings of the First Workshop on Semantic Wikis – From Wiki To Semantics’, Vol. 206, CEUR-WS.org, pp. 139 – 152.
- Ruiz-Casado, M., Alfonseca, E. and Castells, P. (2007), ‘Automatising the learning of lexical patterns: An application to the enrichment of WordNet by extracting semantic relationships from Wikipedia’, *Data Knowledge Engineering* **61**(3), 484–499.

- Sabou, M., D'Aquin, M. and Motta, E. (2008), *Journal on data semantics xi*, Springer-Verlag, Berlin, Heidelberg, chapter Exploring the Semantic Web as Background Knowledge for Ontology Matching, pp. 156–190.
- Sadun, E. and Sande, S. (2012), *Talking to Siri: Learning the Language of Apple's Intelligent Assistant*, Que Publishing.
- Santos, D. and Cabral, L. M. (2009a), GikiCLEF: Crosscultural Issues in an International Setting: Asking non-English-centered Questions to Wikipedia, in F. Borri, A. Nardi and C. Peters, eds, 'Working notes for CLEF', Corfu, Greece.
- Santos, D. and Cabral, L. M. (2009b), GikiCLEF: Expectations and Lessons Learned, in C. Peters, G. M. D. Nunzio, M. Kurimo, D. Mostefa, A. Peñas and G. Roda, eds, 'CLEF (1)', Vol. 6241 of *Lecture Notes in Computer Science*, Springer, pp. 212–222.
- Santos, D., Cabral, L. M., Forascu, C., Forner, P., Gey, F. C., Lamm, K., Mandl, T., Osenova, P., Peñas, A., Rodrigo, Á., Schulz, J. M., Skalban, Y. and Sang, E. T. K. (2010), GikiCLEF: Crosscultural Issues in Multilingual Information Access, in N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odiijk, S. Piperidis, M. Rosner and D. Tapias, eds, 'Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)', European Language Resources Association (ELRA), Valletta, Malta.
- Santos, D. and Cardoso, N. (2008), GikiP: evaluating geographical answers from Wikipedia, in C. Jones and R. Purves, eds, 'Proceedings of the 2nd international workshop on Geographic Information Retrieval (GIR)', ACM, New York, NY, USA, pp. 59–60.
- Santos, D., Cardoso, N., Carvalho, P., Dornescu, I., Hartrumpf, S., Leveling, J. and Skalban, Y. (2008), Getting geographical answers from Wikipedia: the GikiP pilot at CLEF, in F. Borri, A. Nardi and C. Peters, eds, 'Working Notes for the CLEF 2008 Workshop', CLEF 2008 Organizing Committee, Aarhus, Denmark.
- Santos, D., Cardoso, N., Carvalho, P., Dornescu, I., Hartrumpf, S., Leveling, J. and Skalban, Y. (2009), GikiP at GeoCLEF 2008: Joining GIR and QA Forces for Querying Wikipedia, in C. Peters, T. Deselaers, N. Ferro, J. Gonzalo, A. Peñas, G. J. F. Jones, M. Kurimo, T. Mandl and V. Petras, eds, 'Proceedings of the 9th Cross-language evaluation forum conference on Evaluating systems for multilingual and multimodal information access', Vol. 5706 of *Lecture Notes in Computer Science*, Springer, pp. 894–905.
- Schlaefter, N. (2007), *A Semantic Approach to Question Answering*, VDM Verlag Dr. Mueller.

- Schlaefer, N., Giesemann, P., Schaaf, T. and Waibel, A. (2006), A Pattern Learning Approach to Question Answering within the Ephyra Framework., in P. Sojka, I. Kopecek and K. Pala, eds, 'Text, Speech and Dialogue', Vol. 4188 of *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, pp. 687–694.
- Schlobach, S., Olsthoorn, M. and de Rijke, M. (2004), Type Checking in Open-Domain Question Answering, in L. de Mántaras R. and S. L., eds, 'Proceedings of European Conference on Artificial Intelligence', Vol. 2006, IOS Press, Amsterdam, pp. 398–402.
- Scott, W. A. (1955), 'Reliability of content analysis: The case of nominal scale coding', *Public Opinion Quarterly* 19(3), 321–325.
- Sekine, S., Sudo, K. and Nobata, C. (2002), Extended Named Entity Hierarchy, in M. G. Rodríguez and C. P. S. Araujo, eds, 'Proceedings of 3<sup>rd</sup> International Conference on Language Resources and Evaluation (LREC'02)', European Language Resources Association (ELRA), Canary Islands, Spain, pp. 1818–1824.
- Sim, J. and Wright, C. C. (2005), 'The kappa statistic in reliability studies: use, interpretation, and sample size requirements.', *Physical Therapy* 85(3), 257–268.
- Spearman, C. (1904), 'The proof and measurement of association between two rings', *American Journal of Psychology* (15), 72–101.
- Specia, L. and Motta, E. (2007), Integrating folksonomies with the semantic web, in 'Proceedings of the 4th European conference on The Semantic Web: Research and Applications', ESWC '07, Springer-Verlag, Berlin, Heidelberg, pp. 624–639.
- Strube, M. and Ponzetto, S. P. (2006), WikiRelate! Computing Semantic Relatedness Using Wikipedia, in 'Proceedings of the 21st AAAI Conference on Artificial Intelligence', AAAI Press, Menlo Park, California, p. 1419–1424.
- Săcăleanu, B., Orăsan, C., Spurk, C., Ou, S., Ferrandez, O., Kouylekov, M. and Negri, M. (2008), Entailment-based Question Answering for Structured Data, in 'Coling 2008: Companion volume: Posters and Demonstrations', Coling 2008 Organizing Committee, Manchester, UK, pp. 29–32.
- Suchanek, F., Kasneci, G. and Weikum, G. (2007a), Yago: A Large Ontology From Wikipedia and WordNet, Research Report MPI-I-2007-5-003, Max-Planck-Institut für Informatik, Saarbrücken, Germany.
- Suchanek, F. M., Kasneci, G. and Weikum, G. (2007b), Yago: a core of semantic knowledge - Unifying WordNet and Wikipedia, in 'WWW '07: Proceedings of the 16th international conference on World Wide Web', ACM Press, New York, NY, USA, pp. 697–706.

- Tan, B. and Peng, F. (2008), Unsupervised query segmentation using generative language models and wikipedia, in ‘WWW ’08: Proceeding of the 17th international conference on World Wide Web’, ACM, New York, NY, USA, pp. 347–356.
- Thelwall, M. (2004), *Link Analysis: An Information Science Approach*, Emerald Group Publishing.
- Toral, A. and Muñoz, R. (2006), A proposal to automatically build and maintain gazetteers for Named Entity Recognition by using Wikipedia, in ‘Proceedings of the workshop on NEW TEXT Wikis and blogs and other dynamic text sources’, Association for Computational Linguistics, Trento, pp. 56–61.
- Tunstall-Pedoe, W. (2010), ‘True knowledge: Open-domain question answering using structured knowledge and inference’, *AI Magazine* 31(3).
- Vander Wal, T. (2007), ‘Folksonomy coinage and definition’, <http://vanderwal.net/folksonomy.html>.
- Vercoustre, A.-M., Pehcevski, J. and Thom, J. A. (2007), Using Wikipedia Categories and Links in Entity Ranking, in ‘INEX’, pp. 321–335.
- Voorhees, E. M. (2001), ‘The TREC question answering track’, *Natural Language Engineering* 7(4), 361–378.
- Wang, C., Kalyanpur, A., Fan, J., Boguraev, B. K. and Gondek, D. C. (2012), ‘Relation extraction and scoring in deepqa’, *IBM Journal of Research and Development* 56(3.4), 9:1–9:12.
- Wang, G., Yu, Y. and Zhu, H. (2007), PORE: Positive-Only Relation Extraction from Wikipedia Text, in K. Aberer, K.-S. Choi, N. F. Noy, D. Allemang, K.-I. Lee, L. J. B. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber and P. Cudré-Mauroux, eds, ‘ISWC/ASWC’, Vol. 4825 of *Lecture Notes in Computer Science*, Springer, pp. 580–594.
- Wang, G., Zhang, H., Wang, H. and Yu, Y. (2007), Enhancing Relation Extraction by Eliciting Selectional Constraint Features from Wikipedia, in Z. Kedad, N. Lammari, E. Métais, F. Meziane and Y. Rezgui, eds, ‘NLDB’, Vol. 4592 of *Lecture Notes in Computer Science*, Springer, pp. 329–340.
- Watanabe, Y., Asahara, M. and Matsumoto, Y. (2007), A Graph-Based Approach to Named Entity Categorization in Wikipedia Using Conditional Random Fields, in ‘Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)’, Association for Computational Linguistics, Prague, Czech Republic, pp. 649–657.

- Wood, A. and Struthers, K. (2010), 'Pathology education, wikipedia and the net generation.', *Medical teacher* 32(7).
- Woods, W. A. (1977), Lunar Rocks in Natural English: Explorations in Natural Language Question Answering, in A. Zampolli, ed., 'Linguistic Structures Processing', North-Holland, Amsterdam, pp. 521–569.
- Wu, F., Hoffmann, R. and Weld, D. S. (2008), Augmenting Wikipedia-Extraction with Results from the Web, in R. Bunescu, E. Gabrilovich and R. Mihalcea, eds, "Wikipedia and Artificial Intelligence: An Evolving Synergy" Workshop at 23rd AAAI Conference', Technical Report WS-08-15, AAAI Press, Menlo Park, California, pp. 55–60.
- Wu, F. and Weld, D. S. (2007), Autonomously semantifying Wikipedia, in M. J. Silva, A. H. F. Laender, R. A. Baeza-Yates, D. L. McGuinness, B. Olstad, Ø. H. Olsen and A. O. Falcão, eds, 'CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management', ACM, New York, USA, pp. 41–50.
- Wu, F. and Weld, D. S. (2010), Open Information Extraction Using Wikipedia, in 'Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics', Association for Computational Linguistics, Uppsala, Sweden, pp. 118–127.
- Yang, X. and Su, J. (2007), Coreference Resolution Using Semantic Relatedness Information from Automatically Discovered Patterns, in A. Zaenen and A. van den Bosch, eds, 'Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics', Association for Computational Linguistics, Prague, Czech Republic, pp. 528–535.
- Zesch, T. and Gurevych, I. (2007), Analysis of the Wikipedia Category Graph for NLP Applications, in 'Proceedings of the Second Workshop on TextGraphs: Graph-Based Algorithms for Natural Language Processing', Association for Computational Linguistics, Rochester, NY, USA, pp. 1–8.
- Zhang, D. and Lee, W. S. (2003), Question classification using support vector machines, in 'SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval', ACM, New York, NY, USA, pp. 26–32.
- Zirn, C., Nastase, V. and Strube, M. (2008), Distinguishing between Instances and Classes in the Wikipedia Taxonomy, in 'ESWC', pp. 376–387.



# Appendix A:

## Previously Published Work

Some of the work described in this thesis has been previously published in proceedings of peer-reviewed conferences. This appendix provides a short description of these papers and explains their contribution to this thesis:

Santos, D., Cardoso, N., Carvalho, P., **Dornescu, I.**, Hartrumpf, S., Leveling, J. and Skalban, Y. (2008), Getting geographical answers from Wikipedia: the GikiP pilot at CLEF, *in* F. Borri, A. Nardi and C. Peters, eds, ‘Working Notes for the CLEF 2008 Workshop’, CLEF 2008 Organizing Committee, Aarhus, Denmark.

This paper describes the GikiP pilot that took place at GeoCLEF 2008. The article gives an overview of the GikiP competition, defining the task, presenting evaluation measures, describing the results achieved by participants and discussing challenges encountered. My contribution to this paper was a description of the approach employed by the first version of the QA system EQUAL. The results reported in this paper are presented in Chapter 4, Section 4.5.

Santos, D., Cardoso, N., Carvalho, P., **Dornescu, I.**, Hartrumpf, S., Leveling, J. and Skalban, Y. (2009), GikiP at GeoCLEF 2008: Joining GIR and QA Forces for Querying Wikipedia, *in* ‘Proceedings of the 9th Cross-language evaluation forum conference on Evaluating systems for multilingual and multimodal information access’, Vol. 5706 of LNCS, Springer-Verlag, Berlin Heidelberg, pp. 894–905.

This is a revised version of the previous paper, published in 2009, where my contribution to this paper was also a description of EQUAL. This description was further expanded in Section 4.3.

**Dornescu, I.** (2009), EQUAL: Encyclopaedic QUestion Answering for Lists, *in* F. Borri, A. Nardi and C. Peters, eds, ‘Working Notes for CLEF’, Corfu, Greece.

This paper presents the second version of EQUAL, and its participation in the GikiCLEF 2009 task. The system is described in the context of a more general architecture for semantic QA which formed the basis of the architecture proposed in Chapter 3, Section 3.5.

**Dornescu, I.** (2010), Semantic QA for Encyclopaedic Question: EQUAL in GikiCLEF, *in* C. Peters et al., eds, ‘Proceedings of the 10th cross-language evaluation forum conference on Multilingual information access evaluation: text retrieval experiments’, Vol. 6241 of LNCS, Springer-Verlag, Berlin Heidelberg, pp. 326–333.

This is a revised version of the previous paper published in 2010 describing the participation of EQUAL in GikiCLEF. It also includes the results of an error analysis that was carried out. these results were used in Chapter 4, Section 4.5 and Section 4.6, which elaborate on the error analysis presented in the paper. The description of the semantic constraints was expanded in Section 4.4 and the challenges reported in the paper were further developed in Section 4.6.

Cardoso, N., **Dornescu, I.**, Hartrumpf, S. and Leveling, J. (2010), Revamping question answering with a semantic approach over world knowledge, *in* ‘CLEF Labs 2010, Multiple Language Question Answering 2010 (MLQA10), Padua, Italy’. This is a position paper presented at CLEF Labs 2010 which presents a brief summary of the main ideas behind the top 3 semantic approaches employed in GikiCLEF. Some of these ideas are echoed in Chapter 3.

# Appendix B:

## GikiP Topics

Table 1: GikiP 2008 topics (English)

| ID   | English topic  |
|------|--|
| GP1  | Which waterfalls are used in the film “The Last of the Mohicans”?                                  |
| GP2  | Which Vienna circle members or visitors were born outside the Austria-Hungarian empire or Germany? |
| GP3  | Portuguese rivers that flow through cities with more than 150,000 inhabitants                      |
| GP4  | Which Swiss cantons border Germany?  |
| GP5  | Name all wars that occurred on Greek soil.   |
| GP6  | Which Australian mountains are higher than 2000 m?   |
| GP7  | African capitals with a population of two million inhabitants or more                              |
| GP8  | Suspension bridges in Brazil   |
| GP9  | Composers of Renaissance music born in Germany   |
| GP10 | Polynesian islands with more than 5,000 inhabitants  |
| GP11 | Which plays of Shakespeare take place in an Italian setting?                                       |
| GP12 | Places where Goethe lived  |
| GP13 | Which navigable rivers in Afghanistan are longer than 1000 km?                                     |
| GP14 | Brazilian architects who designed buildings in Europe  |
| GP15 | French bridges which were in construction between 1980 and 1990                                    |



# Appendix C:

## GikiCLEF Topics

Table 2: GikiCLEF 2009 topics (English)

| ID      | Topic (English)  |
|---------|--|
| GC09-01 | List the Italian places where Ernest Hemingway visited during his life.  |
| GC09-02 | Which countries have the white, green and red colors in their national flag?   |
| GC09-03 | In which countries outside Bulgaria are there published opinions on Petar Dunov's (Beinsa Duno's) ideas?             |
| GC09-04 | Name Romanian poets who published volumes with ballads until 1941.   |
| GC09-05 | Which written fictional works of non-Romanian authors have as subject the Carpathians mountains?                     |
| GC09-06 | Which Dutch violinists held the post of concertmaster at the Royal Concertgebouw Orchestra in the twentieth century? |
| GC09-07 | What capitals of Dutch provinces received their town privileges before the fourteenth century?                       |
| GC09-08 | Which authors were born in and write about the Bohemian Forest?  |
| GC09-09 | Name places where Goethe fell in love.   |
| GC09-10 | Which Flemish towns hosted a restaurant with two or three Michelin stars in 2008?                                    |
| GC09-11 | What Belgians won the Ronde van Vlaanderen exactly twice?  |
| GC09-12 | Present monarchies in Europe headed by a woman.  |
| GC09-13 | Romantic and realist European novelists of the XIXth century who died of tuberculosis.                               |
| GC09-14 | Name rare diseases with dedicated research centers in Europe.  |
| GC09-15 | List the basic elements of the cassata.  |
| GC09-16 | In which European countries is the bidet commonly used?  |
| GC09-17 | List the 5 Italian regions with a special statute.   |
| GC09-18 | In which Tuscan provinces is Chianti produced?   |
| GC09-19 | Name mountains in Chile with permanent snow.   |
| GC09-20 | List the name of the sections of the North-Western Alps.   |
| GC09-21 | List the left side tributaries of the Po river.  |
| GC09-22 | Which South American national football teams use the yellow color?   |
| GC09-23 | Name American museums which have any Picasso painting.   |
| GC09-24 | Which countries have won a futsal European championship played in Spain?   |

*continued on the next page*

| ID      | Topic (English)   |
|---------|---|
| GC09-25 | Name Spanish drivers who have driven in Minardi.  |
| GC09-26 | Which Bulgarian fighters were awarded the "Diamond belt"?   |
| GC09-27 | Which Dutch bands are named after a Bulgarian footballer?   |
| GC09-28 | Find coastal states with Petrobras refineries.  |
| GC09-29 | Places above the Arctic circle with a population larger than 100,000 people                           |
| GC09-30 | Which Japanese automakers companies have manufacturing or assembling factories in Europe?             |
| GC09-31 | Which countries have Italian as an official language?   |
| GC09-32 | Name Romanian writers who were living in USA in 2003.   |
| GC09-33 | What European Union countries have national parks in the Alps?  |
| GC09-34 | What eight-thousanders are at least partially in Nepal?   |
| GC09-35 | Which Romanian mountains are declared biosphere reserves?   |
| GC09-36 | Name Romanian caves where Paleolithic human fossil remains were found.                                |
| GC09-37 | Which Norwegian musicians were convicted for burning churches?  |
| GC09-38 | Which Norwegian waterfalls are higher than 200m?  |
| GC09-39 | National team football players from Scandinavia with sons who have played for English clubs.          |
| GC09-40 | Which rivers in North Rhine Westphalia are approximately 10km long?                                   |
| GC09-41 | Chefs born in Austria who have received a Michelin Star.  |
| GC09-42 | Political parties in the National Council of Austria which were founded after the end of World War II |
| GC09-43 | Austrian ski resorts with a total ski trail length of at least 100 km                                 |
| GC09-44 | Find Austrian grape varieties with a vineyard area below 100 ha.                                      |
| GC09-45 | Find Swiss casting show winners.  |
| GC09-46 | German writers who are Honorary Citizens in Switzerland.  |
| GC09-47 | Which cities in Germany have more than one university?  |
| GC09-48 | Which German-speaking movies have been nominated for an Oscar?  |
| GC09-49 | Formula One drivers who moved to Switzerland.   |
| GC09-50 | Which Swiss people were Olympic medalists in snowboarding at the Winter Olympic Games in 2006?        |