D1.2.2.1.3 Benchmarking of annotation tools

Diana Maynard (University of Sheffield)

with contributions from:
Stamatia Dasiopoulou (ITI-CERTH), Stefania Costache (L3S)
Kai Eckert, Heiner Stuckenschmidt (University of Mannheim)
Martin Dzbor (Open University), Siegfried Handschuh (NUIG)

Abstract.
EU-IST Network of Excellence (NoE) IST-2004-507482 KWEB
Deliverable D1.2.2.1 (WP1.2/2.1)

This deliverable investigates methods and results for benchmarking annotation tools. We define first some criteria for benchmarking, including both performance and usability issues, and examine those factors which are particularly important for a user in an industrial setting to be able to determine which is the most suitable tool for their use. We also look particularly at two issues: scalability of the tool, which is most important if the tools are to be used in a real industrial setting rather than just as research prototypes; and the best way to evaluate performance. We then perform a series of experiments on the annotation tools, and discuss the results, finally drawing some conclusions about the future of annotation tools.

Document Identifier	KWEB/2007/D1.2.2.1.3/v1.0
Project	KWEB EU-IST-2004-507482
Version	v1.0
Date	October 30, 2007
State	final
Distribution	public

Copyright © 2007 The contributors

Knowledge Web Consortium

This document is part of a research project funded by the IST Programme of the Commission of the European Communities as project number IST-2004-507482.

University of Innsbruck (UIBK) - Coordinator
Institute of Computer Science
Technikerstrasse 13
A-6020 Innsbruck
Austria
Contact person: Dieter Fensel
E-mail address: dieter.fensel@uibk.ac.at

École Polytechnique Fédérale de Lausanne (EPFL)
Computer Science Department
Swiss Federal Institute of Technology
IN (Ecublens), CH-1015 Lausanne
Switzerland
Contact person: Boi Faltings
E-mail address: boi.faltings@epfl.ch

France Telecom (FT)
4 Rue du Clos Courtel
35512 Cesson Sévigné
France. PO Box 91226
Contact person : Alain Leger
E-mail address: alain.leger@rd.francetelecom.com

Freie Universität Berlin (FU Berlin)
Takustrasse 9
14195 Berlin
Germany
Contact person: Robert Tolksdorf
E-mail address: tolk@inf.fu-berlin.de

Free University of Bozen-Bolzano (FUB)
Piazza Domenicani 3
39100 Bolzano
Italy
Contact person: Enrico Franconi
E-mail address: franconi@inf.unibz.it

Institut National de Recherche en Informatique et en Automatique (INRIA)
ZIRST - 655 avenue de l'Europe -
Montbonnot Saint Martin
38334 Saint-Ismier
France
Contact person: Jérôme Euzenat
E-mail address: Jerome.Euzenat@inrialpes.fr

Centre for Research and Technology Hellas / Informatics and Telematics Institute (ITI-CERTH)
1st km Thermi - Panorama road
57001 Thermi-Thessaloniki
Greece. Po Box 361
Contact person: Michael G. Strintzis
E-mail address: strintzi@iti.gr

Learning Lab Lower Saxony (L3S)
Expo Plaza 1
30539 Hannover
Germany
Contact person: Wolfgang Nejdl
E-mail address: nejdl@learninglab.de

National University of Ireland Galway (NUIG)
National University of Ireland
Science and Technology Building
University Road
Galway
Ireland
Contact person: Christoph Bussler
E-mail address: chris.bussler@deri.ie

The Open University (OU)
Knowledge Media Institute
The Open University
Milton Keynes, MK7 6AA
United Kingdom
Contact person: Enrico Motta
E-mail address: e.motta@open.ac.uk

Universidad Politécnica de Madrid (UPM)
Campus de Montegancedo sn
28660 Boadilla del Monte
Spain
Contact person: Asunción Gómez Pérez
E-mail address: asun@fi.upm.es

University of Karlsruhe (UKARL)
Institut für Angewandte Informatik und Formale Beschreibungsverfahren - AIFB
Universität Karlsruhe
D-76128 Karlsruhe
Germany
Contact person: Rudi Studer
E-mail address: studer@aifb.uni-karlsruhe.de

University of Liverpool (UniLiv)
Chadwick Building, Peach Street
L697ZF Liverpool
United Kingdom
Contact person: Michael Wooldridge
E-mail address: M.J.Wooldridge@csc.liv.ac.uk

University of Manchester (UoM)
Room 2.32. Kilburn Building, Department of Computer Science, University of Manchester, Oxford Road
Manchester, M13 9PL
United Kingdom
Contact person: Carole Goble
E-mail address: carole@cs.man.ac.uk

University of Sheffield (USFD)
Regent Court, 211 Portobello street
S14DP Sheffield
United Kingdom
Contact person: Hamish Cunningham
E-mail address: hamish@dcs.shef.ac.uk

University of Trento (UniTn)
Via Sommarive 14
38050 Trento
Italy
Contact person: Fausto Giunchiglia
E-mail address: fausto@dit.unitn.it

Vrije Universiteit Amsterdam (VUA)
De Boelelaan 1081a
1081HV. Amsterdam
The Netherlands
Contact person: Frank van Harmelen
E-mail address: Frank.van.Harmelen@cs.vu.nl

Vrije Universiteit Brussel (VUB)
Pleinlaan 2, Building G10
1050 Brussels
Belgium
Contact person: Robert Meersman
E-mail address: robert.meersman@vub.ac.be

Work package participants

The following partners have taken an active part in the work leading to the elaboration of this document, even if they might not have directly contributed to writing parts of this document:

France Telecom
Institut National de Recherche en Informatique et en Automatique
Learning Lab Lower Saxony
National University of Ireland Galway
The Open University
Universidad Politécnica de Madrid
University of Karlsruhe
University of Sheffield

Changes

Version	Date	Author	Changes
1.0	21.09.07	Diana Maynard	First version
1.1	09.10.07	Diana Maynard	Comments from QA
1.2	29.10.07	Martin Dzbor	Input from OU
1.3	30.10.07	Diana Maynard	Comments from QC

Executive Summary

This deliverable investigates methods and results for benchmarking annotation tools. We define first some criteria for benchmarking, including both performance and usability issues, and examine those factors which are particularly important for a user in an industrial setting to be able to determine which is the most suitable tool for their use. We also look particularly at two issues: scalability of the tool, which is most important if the tools are to be used in a real industrial setting rather than just as research prototypes; and the best way to evaluate performance. We then perform a series of experiments on the annotation tools, and discuss the results, finally drawing some conclusions about the future of annotation tools.

Contents

Chapter 1

Introduction

Ontology-based annotation tools are used to populate ontologies with instances from text, and/or to annotate text with conceptual information from an ontology. This task forms an important part of ontology creation and management, by enabling us to combine and associate existing ontologies, perform more detailed analysis of the text, and to extract deeper and more accurate knowledge. Figure 1.1 shows the relationship between text, annotations and ontologies.

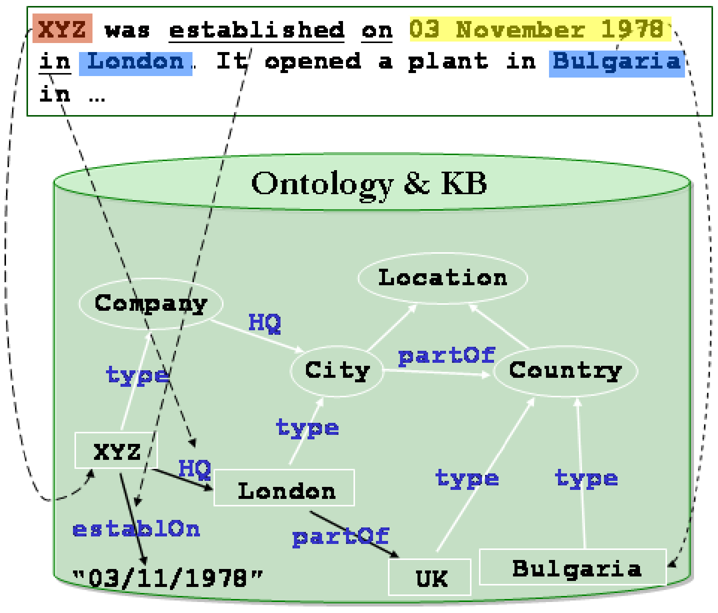


Figure 1.1: Text, annotations and ontologies

Ontology population is a crucial part of knowledge base construction and maintenance that enables us to relate text to ontologies, providing on the one hand a customised ontology related to the data and domain with which we are concerned, and on the other hand a richer ontology which can be used for a variety of semantic web-related tasks such as knowledge management, information retrieval, question answering, semantic desktop applications, and so on.

In this deliverable we define textual annotation as the process (or result) generally performed by means of some kind of ontology-based information extraction (OBIE). This consists of identifying the key terms in the text (such as named entities and technical terms) and then relating them to concepts in the ontology. Typically, the core information extraction is carried out by linguistic pre-processing (tokenisation, POS tagging etc.), followed by a named entity recognition component, such as a gazetteer and rule-based grammar or machine learning techniques.

Multimedia annotation has a similar goal to textual annotation but is, of course, concerned with annotating multimedia information. Due to the complexity of the task, it is mainly performed manually, although research in areas such as image recognition has paved the way for automatic indexing of multimedia – see for example work done in the context of the EU projects MUMIS [SCB$^+$03] and Prestospace [DTCP05]. However, the current state-of-the-art is such that only manual multimedia annotation tools are really viable currently, except for certain very specific tools.

There have been several previous reports claiming to compare and/or evaluate different annotation tools. However, these largely examine only the performance of the information extraction component and have been based on a single set of task-specific data (see for example [SP05]). What all these reports ignore is the fact that there are many more aspects to benchmarking a tool than just measuring its performance in a particular situation. We aim to rectify this by analysing many different factors, such as usability, accessibility, scalability and interoperability, as well as performance. Our aim is to try to aid a user who has to choose which tool would be most appropriate for his/her needs: we therefore need to take into account a variety of different factors other than just performance. We thus also examine the suitability of the tools for different purposes. This is an important point and will be stressed throughout the deliverable: to claim that any one tool is a priori *better than another* makes no real sense here, because it depends on many factors such as who will be using it, what they will be using it for, and which features they care about most (e.g. speed vs. usability).

In this deliverable, we investigate mechanisms for benchmarking both textual and multimedia annotation tools, specifying a set of criteria by which they can be compared. We look at various different factors – not only performance, but also usability, accessibility, scalability and interoperability – and compare the annotation tools according to such criteria. In terms of performance, we first look at a new metric for evaluating ontology-based information extraction (the technology behind most of the annotation tools) which gives us a better indication of the performance levels than traditional metrics. We also investigate various metrics for evaluating automatic document indexing, which use similar techniques. We then compare the performance of the various tools using a variety of criteria (some tools being more suited to different types of evaluation). Finally we try to summarise our results, draw some conclusions about the tools, and give some indications as to the future of annotation tools.

The outline of this deliverable is thus as follows: in this chapter we give an introduction to the task and describe the annotation tools under investigation. Chapter 2 examines the issues of usability, interoperability and accessibility and compares the tools according to a set of criteria. The following two chapters look at metrics for evaluating the performance of annotation tools (Chapter 3) and for evaluating the performance of document indexing (Chapter 4). Chapter 5 then evaluates the performance of annotation tools in various ways. We also look at the problem of scaling annotation tools, which have largely beeen developed as research projects, up to the demands of the real world in Chapter 6. Finally, we conclude in Chapter 7 with a summary of our findings and some thoughts on the future in store for annotation tools.

In the remainder of this chapter, we give a short introduction to the annotation tools that we have investigated in this task. We include a variety of textual and multimedia annotation tools, providing a brief overview of each tool and outlining its main distinctive features. Because the main aim of this deliverable is to give an idea to industrial users of which tool best suits their purpose, we try to provide here some general background information about each tool.

1.1 Textual annotation tools

In Knowledge Web deliverable D2.1.4 [GCMW$^+$05], we proposed to benchmark 5 ontology annotation tools – KIM, GATE, MnM, OntoMat and Melita. We later decided to replace Melita with Magpie, partly because of technical issues regarding Melita, and partly because Magpie has come to the forefront in recent years and has been developed further within Knowledge Web (via the ASPL application).

We have also included Beagle++, which annotates a variety of textual desktop resources for the purposes of indexing and search. While on the one hand it does indeed perform textual annotation, on the other hand it is rather different from the other textual annotation tools such as GATE, and therefore cannot be compared directly. We include it here as an example of a different kind of annotation tool, which nevertheless is still important to benchmark (though there are not really any other similar existing tools against which we can compare it).

The textual annotation tools we have chosen to investigate in this task are therefore GATE, KIM, OntoMat, MnM, Magpie and Beagle++. These have been chosen for a number of reasons. First, they all perform in some way annotation of textual data with respect to an ontology, and are all XML-based. Second, they are all open source, readily available and do not require extensive training to use. There are many other systems available, but most of them are commercial and/or black box systems which are impossible to investigate and to perform experiments with. The tools have also been chosen for their diversity: MnM is a very basic tool which was developed some years ago largely as proof of concept. It is no longer maintained, so is a good reflection of the initial state-of-the-art, and in some way can act as a baseline. GATE, KIM are quite generic tools, which are actively maintained and developed, and are used as the basis of many other annotation systems. OntoMat and Magpie were also developed as quite generic tools, but in slightly different ways: OntoMat requires coupling with an information extraction engine in order to perform automatic annotation, while Magpie – like GATE – can be tuned to a number of different applications.

1.1.1 GATE

GATE [CMBT02] is an architecture for language processing which contains, amongst other things, tools and resources to perform textual annotation both manually and automatically (using information extraction techniques). GATE comes with a default information extraction system called ANNIE [MTU$^+$01], which is designed to be a starting point for adaptation to a specific domain, language, ontology or application. The ANNIE system identifies generic concepts such as persons, locations, organisation, dates, etc., so most tasks require the development of new rules or adaptation of existing rules in order to find instances of different kinds of concepts.

ANNIE consists of the following processing resources: a document structure analyser which parses different input files into GATE documents; a tokeniser which splits a text into different kinds of words; a sentence splitter which segments the document into sentences; a part-of-speech tagger which associates POS tags to words and symbols; a named entity recognition sub-system composed of a gazetteer lookup component and a rule-based pattern matching engine, and a coreference resolution algorithm. Other components which are sometimes necessary, depending on the text and task, are a morphological analyser which produces a root and affix for each word in the document; parsers which associate syntactic and semantic structures with sentences. The named entity system in GATE is a rule-based system developed using a pattern-matching engine called JAPE [CMT00] which is ontologically aware, making the mapping of entities into ontological classes possible during entity recognition. Figure 1.2 depicts a typical ANNIE pipeline.

GATE's Ontology-Based Information Extraction (OBIE) system has been used for ontology-based annotation in a variety of contexts and applications: for example financial business intelligence [MSY$^+$07], monitoring employment opportunities and business intelligence in the chemical engineering industry [MYKK05], species identification in the fisheries domain in the NeOn project[1], annotation of news texts in the SEKT project [PABC05], identification of keywords in academic publications, and so on. GATE is used also by many other research projects and companies, for example by Garlik[2] (giving con-

[1]http://www.neon-project.org
[2]http://www.garlik.com

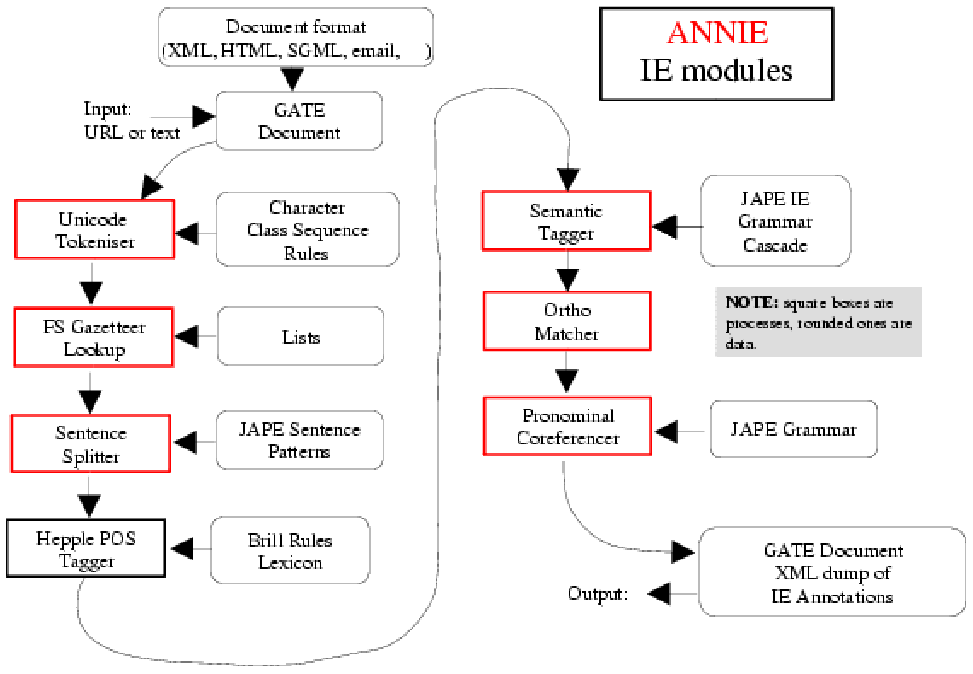


Figure 1.2: Architecture of ANNIE

sumers control over their personal data), Fizzback[3] (enabling companies to capture instant feedback from their consumers) and Innovantage[4] (providing business intelligence in the employment market). Its core components are also used in many other annotation tools such as KIM, Magpie, and MnM. It is also possible to use GATE solely as a manual annotation tool: current research involves developing components for collaborative annotation by multiple users simultaneously, as part of the EU NeOn project. GATE sources can be downloaded from `http://gate.ac.uk`.

1.1.2 KIM

KIM [PKK+04] offers an end-to-end, extendable system which addresses the complete cycle of metadata creation, storage, and semantic-based search and includes a set of front-ends for online use, that offer semantically enhanced browsing. KIM contains an instance base (KIMO) which has been pre-populated with 200,000 entities. KIM has a special ontology enrichment stage where new instances found in the text are added to the ontology. This often involves a disambiguation step, because many instances could be added in more than one place. For example, "Paris" could be an instance of the country France or the state of Texas. The disambiguation process uses an Entity Ranking algorithm, which involves priority ordering of entities with the same label based on corpus statistics.

[3]http://www.fizzback.com
[4]http://www.innovantage.co.uk

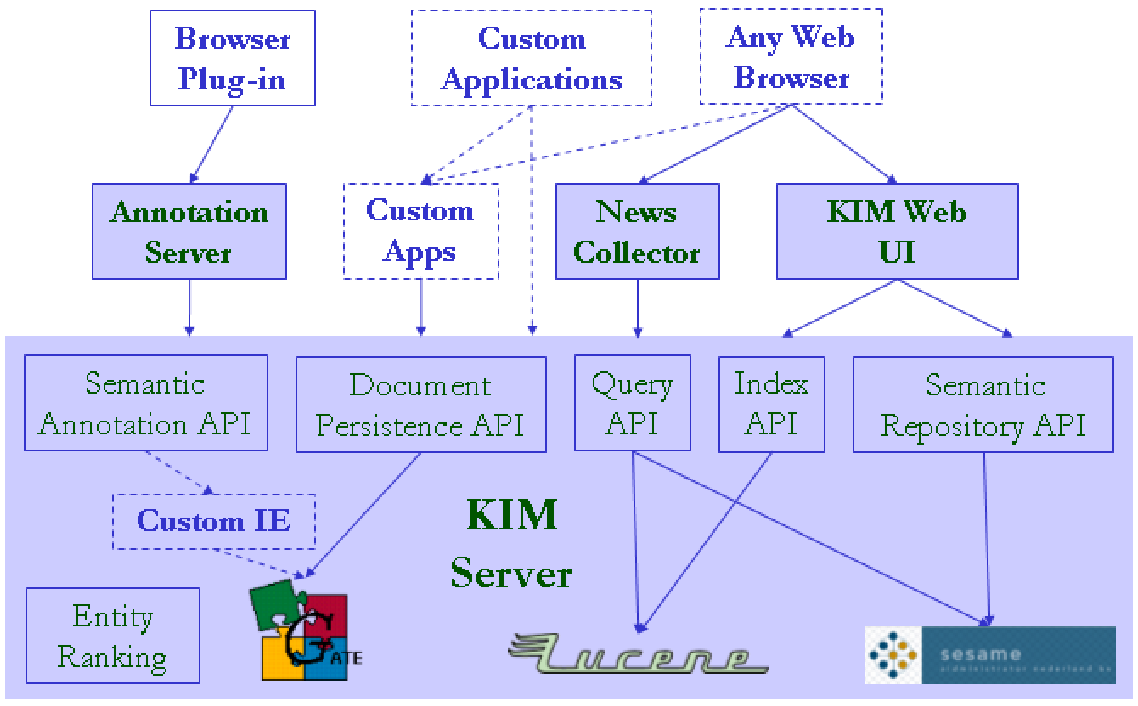


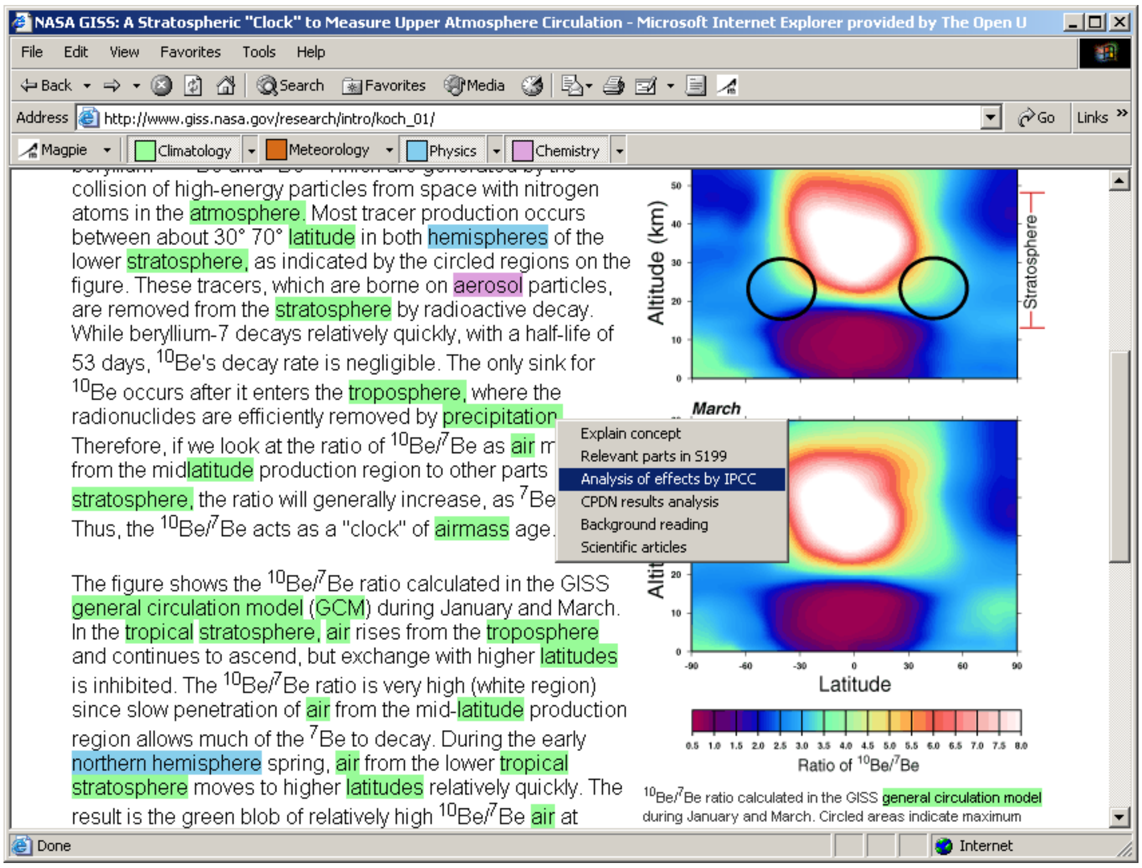
Figure 1.3: Architecture of KIM

The essence of the KIM IE engine is the recognition of named entities with respect to the KIM ontology, which is achieved using a version of ANNIE [MTC+02]. The entity instances all bear unique identifiers that allow annotations to be linked both to the entity type and to the exact individual in the instance base. For new (previously unknown) entities, new identifiers are allocated and assigned; then minimal descriptions are added to the semantic repository. The annotations are kept separately from the content, and an API for their management is provided.

The architecture of KIM (shown in Figure 1.3) consists of the KIM ontology, a knowledge base, the KIM server (with an API for remote access, embedding and integration), and front-ends (a browser plugin for Internet Explorer, the KIM web user interface with various access methods, and the Knowledge Explorer for navigation of the knowledge base). KIM relies on GATE, SESAME and Lucene, but all these, as well as the software platform itself, are domain and task independent.

1.1.3 Magpie

Magpie [DDM04] is a suite of tools which supports the interpretation of webpages and "collaborative sense-making", by annotating a text with instances from a known ontology. These instances can be used as a confidence measure for carrying out some services. The principle behind it is that it uses an ontology to provide a very specific and personalised


NASA GISS: A Stratospheric "Clock" to Measure Upper Atmosphere Circulation - Microsoft Internet Explorer provided by The Open U
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.giss.nasa.gov/research/intro/koch_01/
Go Links
Magpie Climatology Meteorology Physics Chemistry
Altitude (km)
Stratosphere
March
Explain concept
Relevant parts in S199
Analysis of effects by IPCC
CPDN results analysis
Background reading
Scientific articles
Latitude
Ratio of ^{10}Be/^{7}Be
Done
Internet

collision of high-energy particles from space with nitrogen atoms in the atmosphere. Most tracer production occurs between about 30° 70° latitude in both hemispheres of the lower stratosphere, as indicated by the circled regions on the figure. These tracers, which are borne on aerosol particles, are removed from the stratosphere by radioactive decay. While beryllium-7 decays relatively quickly, with a half-life of 53 days, ^{10}Be's decay rate is negligible. The only sink for ^{10}Be occurs after it enters the troposphere, where the radionuclides are efficiently removed by precipitation. Therefore, if we look at the ratio of ^{10}Be/^{7}Be as air m from the midlatitude production region to other parts stratosphere, the ratio will generally increase, as ^{7}Be Thus, the ^{10}Be/^{7}Be acts as a "clock" of airmass age.

The figure shows the ^{10}Be/^{7}Be ratio calculated in the GISS general circulation model (GCM) during January and March. In the tropical stratosphere, air rises from the troposphere and continues to ascend, but exchange with higher latitudes is inhibited. The ^{10}Be/^{7}Be ratio is very high (white region) since slow penetration of air from the mid-latitude production region allows much of the ^{7}Be to decay. During the early northern hemisphere spring, air from the lower tropical stratosphere moves to higher latitudes relatively quickly. The result is the green blob of relatively high ^{10}Be/^{7}Be air at

^{10}Be/^{7}Be ratio calculated in the GISS general circulation model during January and March. Circled areas indicate maximum

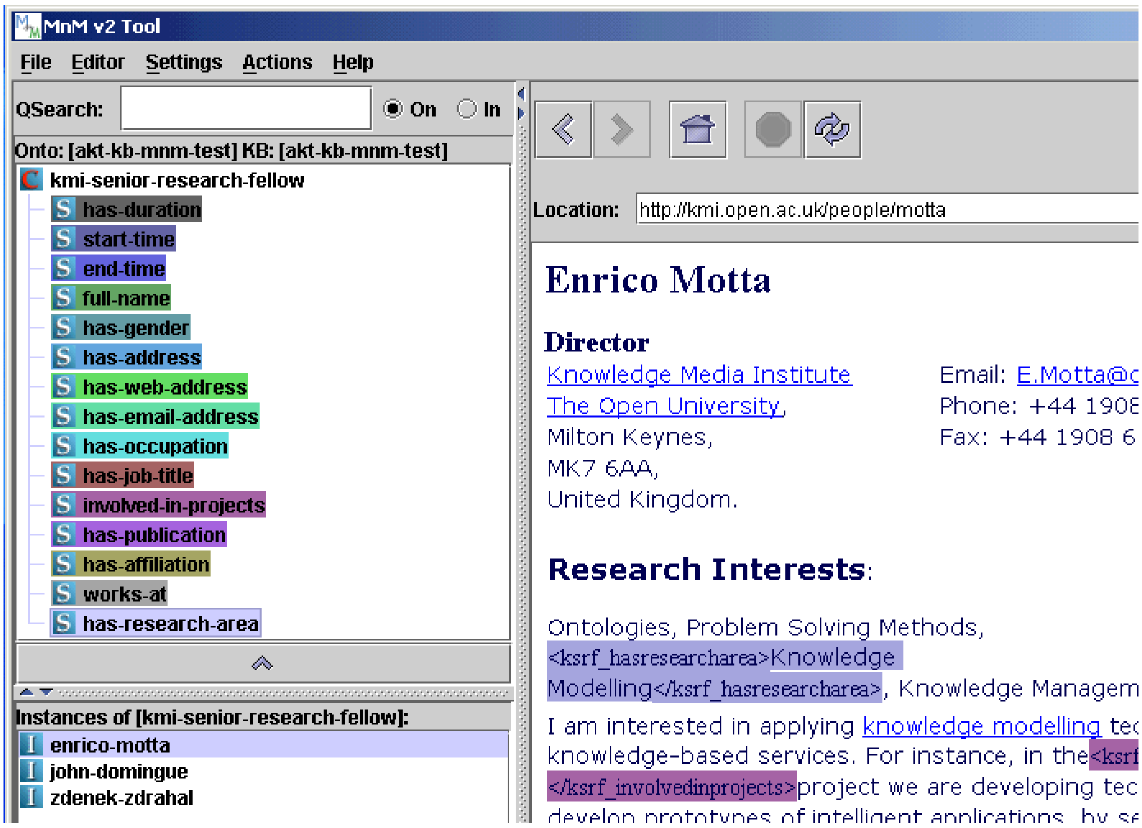
Figure 1.4: Screenshot of MAGPIE

viewpoint of the webpages the user wishes to browse. This is important because different users often have different degrees of knowledge and/or familiarity with the information presented, and have different browsing needs and objectives.

In terms of its named entity recognition capabilities, Magpie relies mainly on simply recognising existing instances in the ontology, which are stored in a simple lexicon to speed up search. The main method used is simple string matching between text and ontology instance, although some lexicon-based pattern matching is also carried out, using components from GATE (for example to combine a known first name of a person with an unknown surname). Figure 1.4 shows a sample scientific text viewed through Magpie. Relevant concepts from climatology, physics and chemistry are highlighted in different colours (with a key in the Magpie toolbar), depicting the annotations produced by the system.

Magpie is used as the basis for various more targeted applications such as ASPL (Advanced Semantic Platform for Learning), which has been well documented in the previous deliverable D3.3.3 [DS05]. While Magpie is a generic platform that can make use of any ontology-derived lexicon, ASPL is one of the Magpie applications; i.e. it uses the same principle of layering semantic information onto a web page and of associating the web services with concepts, etc. Magpie is essentially the engine enabling ASPL, which is a bit more targeted: ASPL uses real world datasets, basically assuming an open world, and tries to design and implement services that can help users in making some analyses in such a world.

1.1.4 MnM



Onto: [akt-kb-mnm-test] KB: [akt-kb-mnm-test]

kmi-senior-research-fellow
- has-duration
- start-time
- end-time
- full-name
- has-gender
- has-address
- has-web-address
- has-email-address
- has-occupation
- has-job-title
- involved-in-projects
- has-publication
- has-affiliation
- works-at
- has-research-area

Instances of [kmi-senior-research-fellow]:
- enrico-motta
- john-domingue
- zdenek-zdrahal

Location: http://kmi.open.ac.uk/people/motta

Enrico Motta

Director

Knowledge Media Institute
The Open University,
Milton Keynes,
MK7 6AA,
United Kingdom.

Email: E.Motta@c
Phone: +44 1908
Fax: +44 1908 6

Research Interests:

Ontologies, Problem Solving Methods, <ksrf_hasresearcharea>Knowledge Modelling</ksrf_hasresearcharea>, Knowledge Managem

I am interested in applying knowledge modelling tec knowledge-based services. For instance, in the <ksrf </ksrf_involvedinprojects>project we are developing tec develop prototypes of intelligent applications, by se

Figure 1.5: Screenshot of MnM

MnM [MVVD$^+$02] is an annotation tool which provides both automated and semi-automated support for annotating web pages with semantic content. MnM integrates a web browser with an ontology editor and provides open APIs to link to ontology servers and for integrating information extraction tools. This enables a user to mark up web resources by instantiating generic concepts from a standardised terminology (an ontology) for a particular domain. It allows the user the facility to choose ontologies from a variety of sources. All of this can be achieved in a familiar web browser environment.

MnM, along with OntoMat, was one of the earliest supervised IE systems for annotation; it was designed to mark up training data (corpus) for IE tools rather than as an annotation tool per se. This means that it stores marked up documents as tagged versions of the original (storing annotations directly in the document), rather than the RDF formats used by the Semantic Web community. In the Semantic Web, documents and their annotations are usually stored separately, so that documents and annotations can be owned by different people or organisations and stored in different places.

MnM learns how to recognise new objects that require annotation, by learning from a collection of previously annotated documents (markup derived from pre-existing ontologies). The tool works by annotating a training set of text and/or HTML documents and then using this to generate lexical rules which can be used to automatically extract information from another set of documents. The key in the use of MnM is a simple Browse – Markup – Learn – Extract process model, by which users select classes by browsing an ontology, mark web resources in terms of appropriate class slots and then press a *Learn* button in order to generate generic information extraction mechanisms, able to extract examples of the user annotations. The instances derived from this process can be used to populate the ontology (using a WebOnto repository) used in the annotation. This semi-automatic learning process usually requires the mark-up of a considerable collection of documents. The MnM system was built to investigate how this task could be facilitated for domain experts.

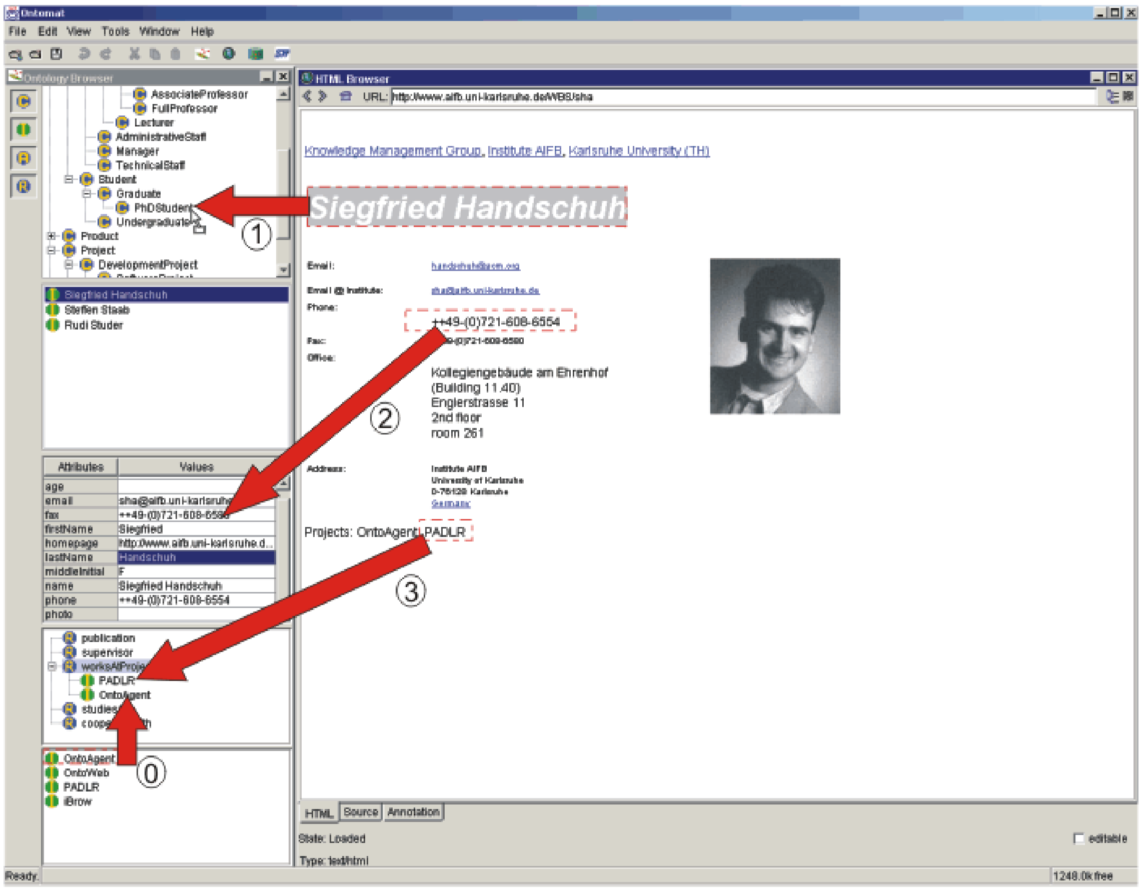
MnM supports a variety of IE plug-ins by the user of a generic API, but it is currently coupled with version 2 of the Amilcare information extraction plugin developed at the University of Sheffield. MnM streamlines the process of tagging the training and testing corpora for use with Amilcare.

Figure 1.5 shows an example of using MnM to populate the AKT reference ontology. In the picture, the home page of Enrico Motta has been marked up with slots from the class `kmi-senior-research-fellow`. This markup can be used to directly modify the instance `enrico-motta`, or as the basis for developing information extraction mechanisms.

1.1.5 OntoMat

OntoMat-Annotizer [HSC02] is a user-friendly interactive annotation tool for web pages. Along with MnM, it was one of the earliest annotation tools [HSM01]. It supports the user in the task of creating and maintaining ontology-based OWL markups, i.e. creating instances, attributes and relationships. It includes an ontology browser for the exploration of the ontology and instances, and an HTML browser that displays the annotated text. It is Java-based and provides a plugin interface for extensions. The intended user is the individual annotator, i.e. somebody who wants to enrich their web pages with OWL metadata. Instead of manually annotating the page with a text editor, OntoMat allows the annotator to highlight relevant parts of the web page and create new instances via drag'n'drop interactions. It supports the metadata creation phase of the lifecycle, and is used in the OntoAgent project[5] amongst others.

[5] www.aifb.uni-karlsruhe.de/WBS/aeb/ontoagent/



Attributes	Values
age	
email	sha@aifb.uni-karlsruhe
fax	++49-(0)721-608-6580
firstName	Siegfried
homepage	http://www.aifb.uni-karlsruhe.d...
lastName	Handschuh
middleInitial	F
name	Siegfried Handschuh
phone	++49-(0)721-608-6554
photo	

Knowledge Management Group, Institute AIFB, Karlsruhe University (TH)

Siegfried Handschuh

Email:	[illegible]
Email @ Institute:	[illegible]
Phone:	++49-(0)721-608-6554
Fax:	[illegible]
Office:	Kollegiengebäude am Ehrenhof (Building 11.40) Englerstrasse 11 2nd floor room 261
Address:	Institute AIFB University of Karlsruhe D-76128 Karlsruhe Germany

Projects: OntoAgent, PADLR

Figure 1.6: Annotation with OntoMat

Figure 1.6 depicts an example of annotation with OntoMat. Here, for example, the user selects a piece of text and drags it to the relevant concept in the ontology in order to annotate it. Automatic annotation can also be carried out if the relevant infrastructure is present – the results will look similar. OntoMat sources can be downloaded from `http://projects.semwebcentral.org/projects/ontomat/`.

1.1.6 Beagle^{++}

Beagle^{++} is a desktop search engine enhanced with semantic annotations for the resources available on a user's dektop: files, emails, publications, visited web pages. Also, it provides additional embedded applications which enhance the semantic functionalities provided by adding more links between the available resources, e.g. a file is related to the email that it was attached to.

As a basis for the Beagle^{++} environment, the open source Beagle desktop search engine[6] for Linux is used, which is extended with advanced searching and ranking capabilities exploiting annotated information. Figure 1.7 illustrates the overall Beagle^{++} architecture.

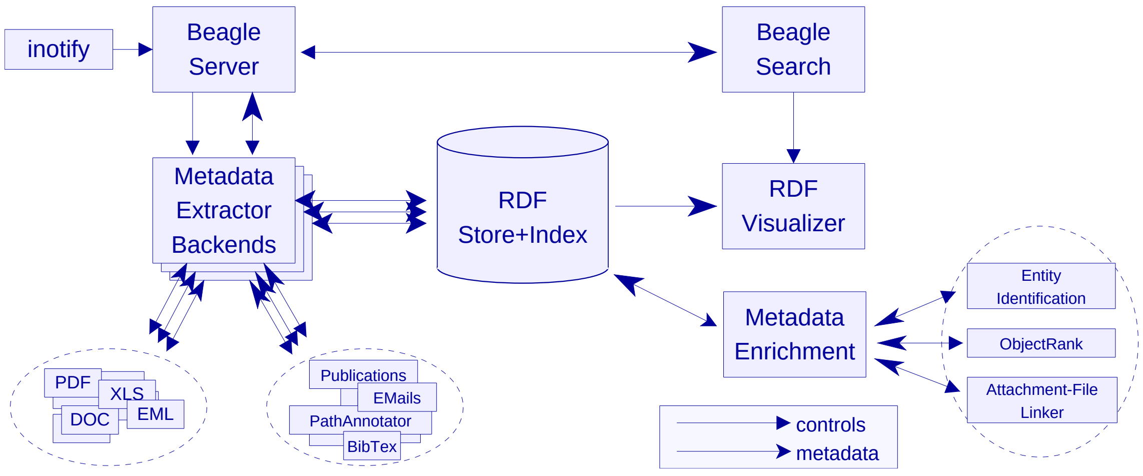


Figure 1.7: Beagle^{++} Architecture Overview

All file system events (files being created, modified or deleted) caught by the *inotify*-enabled Linux Kernel are sent to the Beagle Server, which requests the *Metadata Extraction Backends* to annotate the corresponding resources. Extracted content and metadata information are stored and indexed in a central RDF store. Three *Metadata Enrichment* modules perform specific algorithms on the stored RDF data and enrich these with new metadata:

Scientific Publications In the research community, many papers are available in PDF format. Although PDF allows basic metadata annotations like title and authors, this is rarely used. A tool has been developed to extract metadata from files using the publicly available Citeseer[7] and DBLP[8] databases. If the title is found in the databases, the user will receive additional annotations like authors or conference, year of publication, etc. from these databases.

Path Annotation Folder hierarchies are barely utilised by the search algorithms, in spite of the often sophisticated classification hierarchies that users construct. For example, pictures taken in Hannover could be stored in a directory entitled “Germany”, so it would be useful if we could use this information for search. The path annotation component annotates files with each token in their file path, as well as additional semantic information provided by the WordNet system[9], such as synonyms, hyponyms, hypernyms, meronyms and holonyms [CGG^{+}05].

[6]http://beagle-project.org
[7]http://citeseer.ist.psu.edu/
[8]http://dblp.uni-trier.de/
[9]http://wordnet.princeton.edu/

WebCache This component facilitates the user's search for web pages by starting from a familiar or prominent web site. The notion of "visited link" is broadened by defining it as a web page that was previously visited by the user (the link's target page is present in the browser cache). Such metadata is created for every web page in the cache, containing the links that have been visited from that page, as well as the in-going links from which the user could have arrived (inverse of a visited link).

Finally, some additional annotation algorithms are used in order to enhance this automatically extracted available information, to facilitate the finding of information.

The *entity identification* algorithm, included in Beagle^{++}, aims at enriching the metadata by identifying the entities along with their associated objects (such as duplicate objects with different spelling etc.). The algorithm uses the context of the stored metadata to construct a Bayesian network representing the status of the entities and objects of the metadata and suggests similarities between attributes which prove entity matches, providing information about the objects that refer the same entities.

While Beagle^{++} is not strictly an annotation tool, we include it here because it does contain an automatic semantic annotation component, and it provides a useful tool for those looking to do a little more than basic metadata annotation on text.

1.2 Multimedia annotation tools

The continually increasing volumes of non-textual digital content made available, demand for means to describe such content in order to render it accessible, and thus utilizable. The extraction of such descriptions (annotations) has always been the *holy grail* for the multimedia community, resulting in a vast range of diverse approaches. The *semantic gap* [SGJ01] however, between representations that can be automatically extracted from such content (i.e. features such as colour and shape attributes) and the semantics being conveyed, constitutes still a tremendous challenge that computer vision is still far from confronting, despite the significant achievements reported the last decades. In the majority of the previous decade's approaches, such descriptions focused mostly on features, whose implicit associations with conceptual notions were exploited for the identification of relevant content during search and retrieval. Such representations usually follow the XML-based MPEG-7 standard [Mar02], the greatest effort towards the standardisation of multimedia content description tools.

The advent of the Semantic Web however, paved a new direction, establishing ontologies as the means for capturing and representing domain conceptualisations, and thus providing multimedia content annotation with precise vocabularies, explicit semantics and automated inference services. The multimedia community influenced by the new technologies and the potential brought, began to adopt and assess SW tools utility in the different stages of multimedia content processing and annotation: ontologies started to be explored for representing knowledge required for analysis (descriptions extraction) and for defining the produced annotation vocabulary and semantics. Ontology-based automatic approaches to multimedia annotation fall out of the scope of this deliverable, since their study would translate to the examination of the rationales underlying the methodologies that realise the extraction process, i.e. an analysis rather than annotation oriented perspective. Therefore, we address solely tools for manual annotation.

Before proceeding with the presentation of the examined tools, we discuss some further characteristics, prominent of multimedia content, so as to provide the reader with the context of the follow-up comparison. Generally speaking, multimedia annotation tools may serve two purposes:

1. they allow for attaching descriptive metadata to multimedia content so as to enable its further semantic processing (e.g. summarisation, filtering, search), and

2. they support the automatic extraction of such metadata, serving as sources for eliciting different types of knowledge necessary for analysis.

The difference in the aforementioned usages reflects on the functionalities required. For example, a tool developed to support analysis needs to allow the representation of signal feature information. Another characteristic, prominent of multimedia content, is that multimedia comes in two intertwined layers: the content layer (referring to the meaning conveyed) and the media layer (referring to structural and decomposition aspects). As a result, tools for multimedia annotation have in general to account for both aspects. Practically, this translates into being able to provide representations of structural aspects as well, including localisation information. For example, when annotating a region in an image, one needs to be able to specify and identify this region.

In the following, we attempt to evaluate the various multimedia annotation tools with respect to a number of features that emerge within the aforementioned context of usages. In the undertaken study we considered the following ontology-based, manual, annotation tools: M-OntoMat-Annotizer, PhotoStuff, AKTive Media and Ontolog. The early stage witnessed in ontology-based multimedia annotation tools leads us to adopt a slightly different methodology for their comparison and evaluation than that used for the textual annotation tools. This differentiation was further dictated by the fact that the text annotation tools have as main objective the recognition of (fractions of) the analysed text semantics, while the multimedia ones assume the manual definition of the respective content semantics.

1.2.1 M-OntoMat-Annotizer

M-OntoMat-Annotizer [10] [SPA$^+$06], is a tool intended to contribute towards knowledge acquisition for automatic annotation of multimedia content. It allows the user to extract

[10] $http://www.acemedia.org/aceMedia/results/software/m-ontomat-annotizer.html$

MPEG-7 visual descriptors from both images and videos and to store these descriptors as so-called *visual prototypes* of ontology classes. The prototypes are stored as RDF instances using a RDF version of the MPEG-7 visual descriptors. By using this prototype approach, direct linking of domain concepts to visual instances is avoided, and thus the ontologies' semantics are kept within OWL-DL (i.e. no reification is necessary).

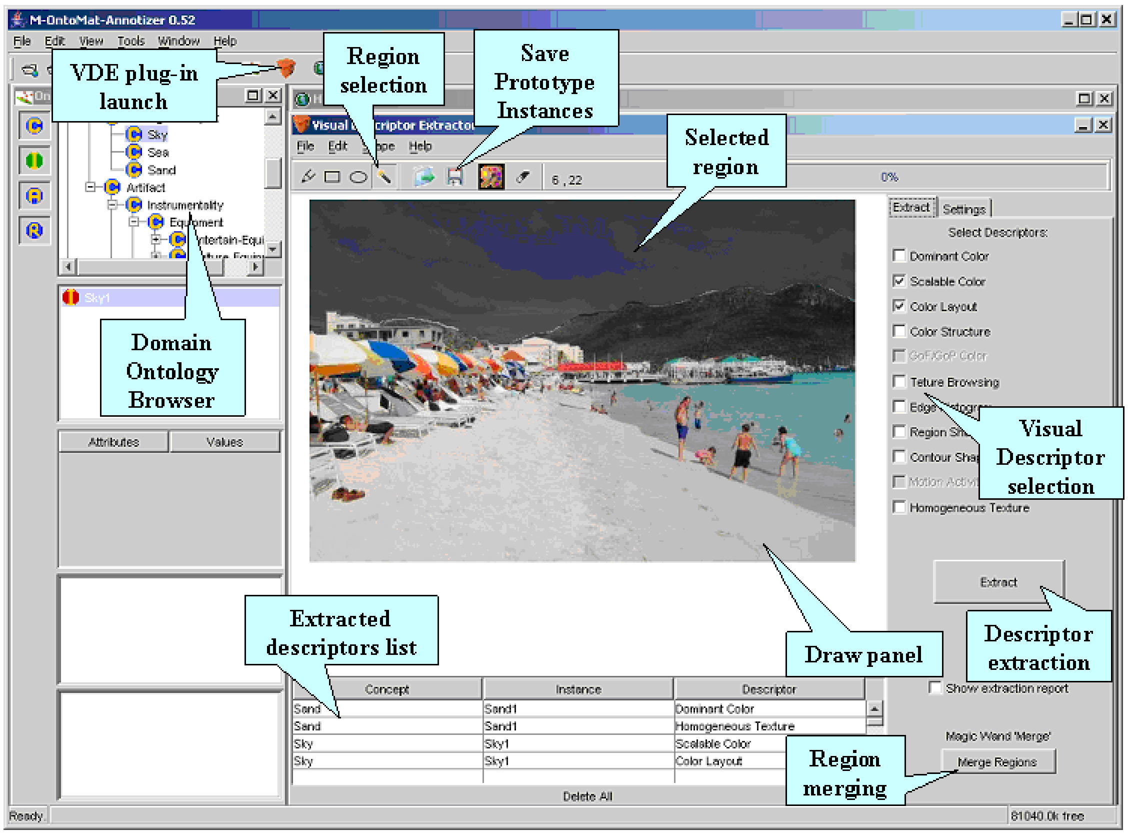
In terms of implementation, the M-OntoMat-Annotizer is an extension of the CREAM (CREAting Metadata for the Semantic Web) framework and its reference implementation, OntoMat-Annotizer [HSC02]. The Visual Descriptor Extraction Tool (VDE), developed as a plugin to OntoMat-Annotizer, is the core component for extending its capabilities and supporting the initialisation and linking of domain ontologies with low-level MPEG-7 visual descriptors.

More specifically, the VDE plugin supports the transformation of the extracted XML multimedia resources into instances of the visual descriptors defined in the Visual Descriptor Ontology (VDO) [BPS$^+$05], by means of an XSL transformation specification that creates a corresponding descriptor instance for each extraction, which is then handed to the knowledge base. Then, the VDE automatically links the newly created visual descriptor instance (e.g. the $vdo : HomogeneousTextureDescriptor$ of the example) with the selected domain concept prototype instance (e.g. Sand_Prototype_1 instance of concept Sand). All concept prototype instances, together with the corresponding extracted descriptors, are eventually saved in an RDF file.

As illustrated in Figure 1.8, the provided interface seamlessly integrates with the OntoMat-Annotizer one. As it is common to extract visual features not only for the entire image, but with respect to specific objects included in the image (video frame) as well, M-OntoMat-Annotizer allows the user to draw a region of interest in the image (video frame) and apply the multimedia descriptor extraction only to this selected region. Alternatively, to ease some of the annotator effort, M-OntoMat-Annotizer also supports automatic segmentation of images (video frames). Whenever a new image (video frame) is loaded it is automatically segmented into regions, leaving to the user only the selection (by clicking) of the desired region(s). To account for under-segmentation phenomena (i.e. having more than one regions corresponding to a single semantic concept), the tool allows the user to merge two or more regions, before proceeding with the extraction of visual descriptors. It is noted again that the purpose of M-OntoMat-Annotizer is to annotate images and videos for assisting subsequent multimedia analysis, not to produce end-user annotations.

The supported languages for the domain ontology include RDFS, DAML, and OIL. The produced annotations are at region level, and region selection can be performed via scribble, rectangle and ellipse shape, or magic wand. Video annotation does not take into account the temporal dimension, i.e. annotation is performed in a frame by frame manner, following the same procedure as for still images. To allow for localisation representation, M-OntoMat-Annotizer uses masks that are linked with the respective instances through naming conventions.

A main limitation that is, however, shared among all the multimedia annotation tools,



Select Descriptors:

- [ ] Dominant Color
- [x] Scalable Color
- [x] Color Layout
- [ ] Color Structure
- [ ] GoF/GoP Color
- [ ] Teture Browsing
- [ ] Edge Histogram
- [ ] Region Shape
- [ ] Contour Shape
- [ ] Motion Activity
- [ ] Homogeneous Texture

Concept	Instance	Descriptor
Sand	Sand1	Dominant Color
Sand	Sand1	Homogeneous Texture
Sky	Sky1	Scalable Color
Sky	Sky1	Color Layout

Figure 1.8: Annotation with M-OntoMat-Annotizer

is the inability to open existing annotations and modify them (add or remove). This can be attributed to the following reasons:

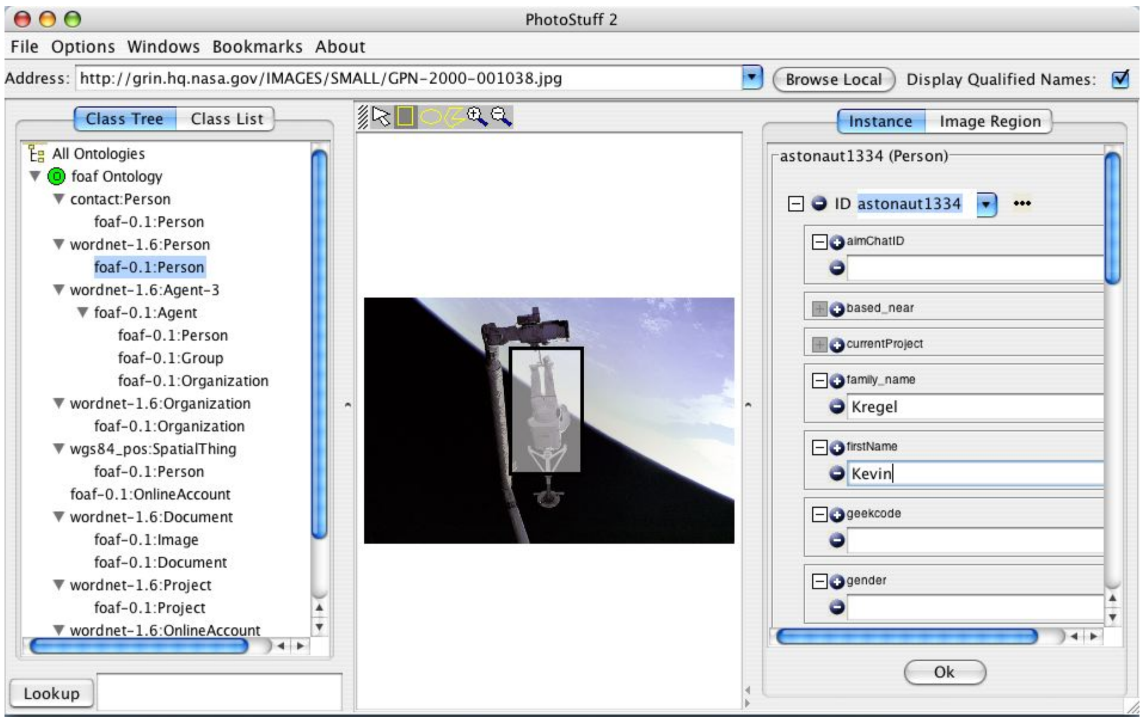
- due to the inherent media layer, the different tools should follow consensual media representations, a requirement that constitutes still a great challenge, as the reported initiatives follow different conceptualisations;
- the recency of the efforts and research related to multimedia and Semantic Web;
- the different applications that the tools target.

1.2.2 Photostuff

PhotoStuff[11] [HWGS+05] is a toolkit that provides users with the means to annotate regions of still images with respect to an RDF or OWL ontology, and publish the generated metadata to the Web, as shown in Figure 1.9. It is a platform-independent image anno-

[11] *http://www.mindswap.org/2003/PhotoStuff/*

tation tool which uses an ontology to provide the expressiveness required to describe the contents of an image, as well as information about the image, such as creation date etc.



Address: http://grin.hq.nasa.gov/IMAGES/SMALL/GPN-2000-001038.jpg
Display Qualified Names: ✓

All Ontologies
foaf Ontology
contact:Person
foaf-0.1:Person
wordnet-1.6:Person
foaf-0.1:Person
wordnet-1.6:Agent-3
foaf-0.1:Agent
foaf-0.1:Person
foaf-0.1:Group
foaf-0.1:Organization
wordnet-1.6:Organization
foaf-0.1:Organization
wgs84_pos:SpatialThing
foaf-0.1:Person
foaf-0.1:OnlineAccount
wordnet-1.6:Document
foaf-0.1:Image
foaf-0.1:Document
wordnet-1.6:Project
foaf-0.1:Project
wordnet-1.6:OnlineAccount

astonaut1334 (Person)
ID astonaut1334
aimChatID
based_near
currentProject
family_name: Kregel
firstName: Kevin
geekcode
gender

Figure 1.9: Annotation with Photostuff

PhotoStuff supports loading multiple ontologies at once, enabling the user to mark up images with concepts distributed across any of the input ontologies. It supports loading ontologies and images from both the Web and locally, while the produced annotations can be either stored to disk or published to a Web portal. Using a variety of region drawing tools, users are able to highlight regions around portions of images (from Web and/or local disk) loaded in PhotoStuff. Classes from loaded ontologies can be dragged into any region, or into the image itself, creating a new instance of the selected class. Alternatively, the user can first specify regions in the image, and then drag a class from the class tree to one of these regions, so as to create region-associated annotations. On the other hand, one can create annotations using pre-existing instances by dragging an entry from the instance list to the examined image or its regions. Support for batch annotation is also provided. Furthermore, it takes advantage of existing metadata embedded in image files (Dublin core and EXIF file) by extracting and encoding such information in RDF/XML. Another feature of PhotoStuff is the use of bookmarks, which serve as an easy way to keep named links to ontologies and media that are of general interest and which one would like to access quickly.

PhotoStuff provides a simple plugin-based architecture for extending the functionality of PhotoStuff in any of a variety of ways, for example, adding support for a new type of data store or media (see Figure 1.10). Currently there are three plugins for PhotoStuff

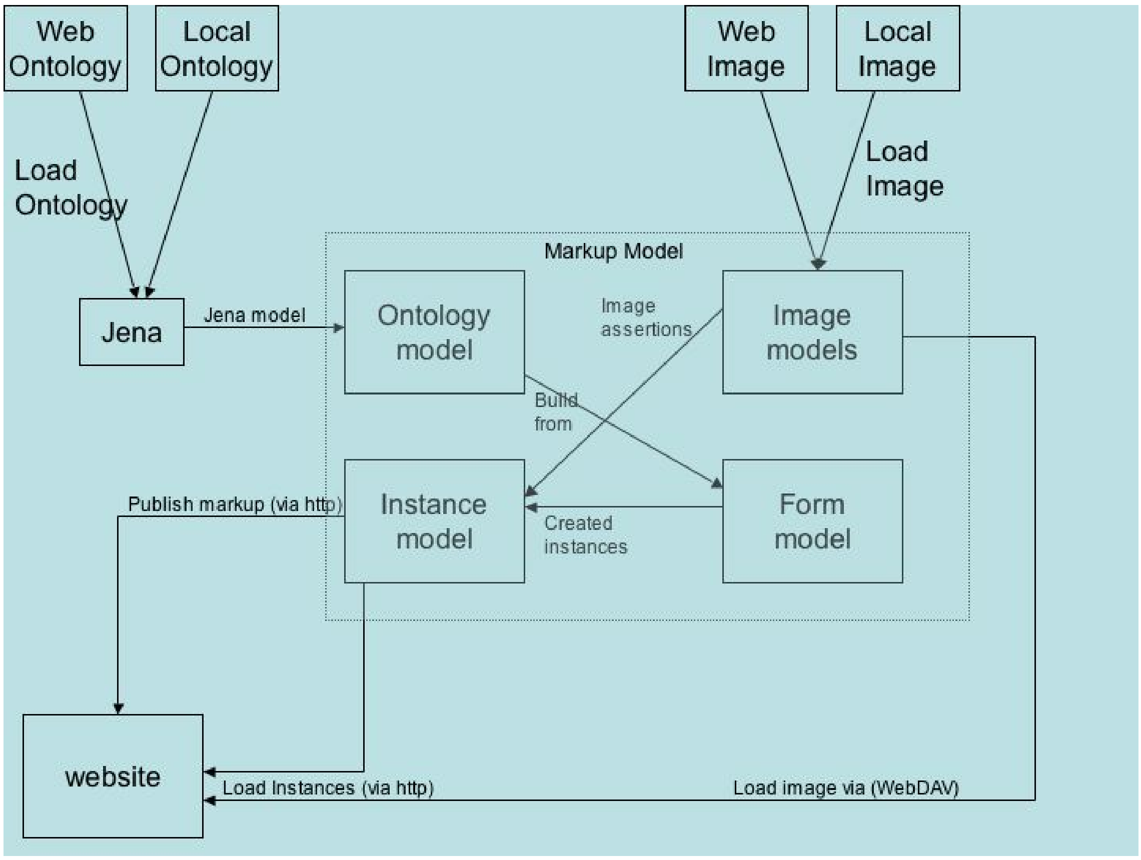


Figure 1.10: Architecture of Photostuff

available. Two of these extend PhotoStuff's basic data store support beyond file/URI-based and Mindswap portal-backed technologies. There is also a plugin for using Sesame and Kowari data sources. These plugins allow users to seamlessly create, add and use data stores that are backed by either Sesame or Kowari. There is also a plugin for adding Natural Language support to the main PhotoStuff user interface. This plugin offers NL descriptions of media and their depictions, and also for classes loaded into the tool. These plugins can all be downloaded from the PhotoStuff download page.

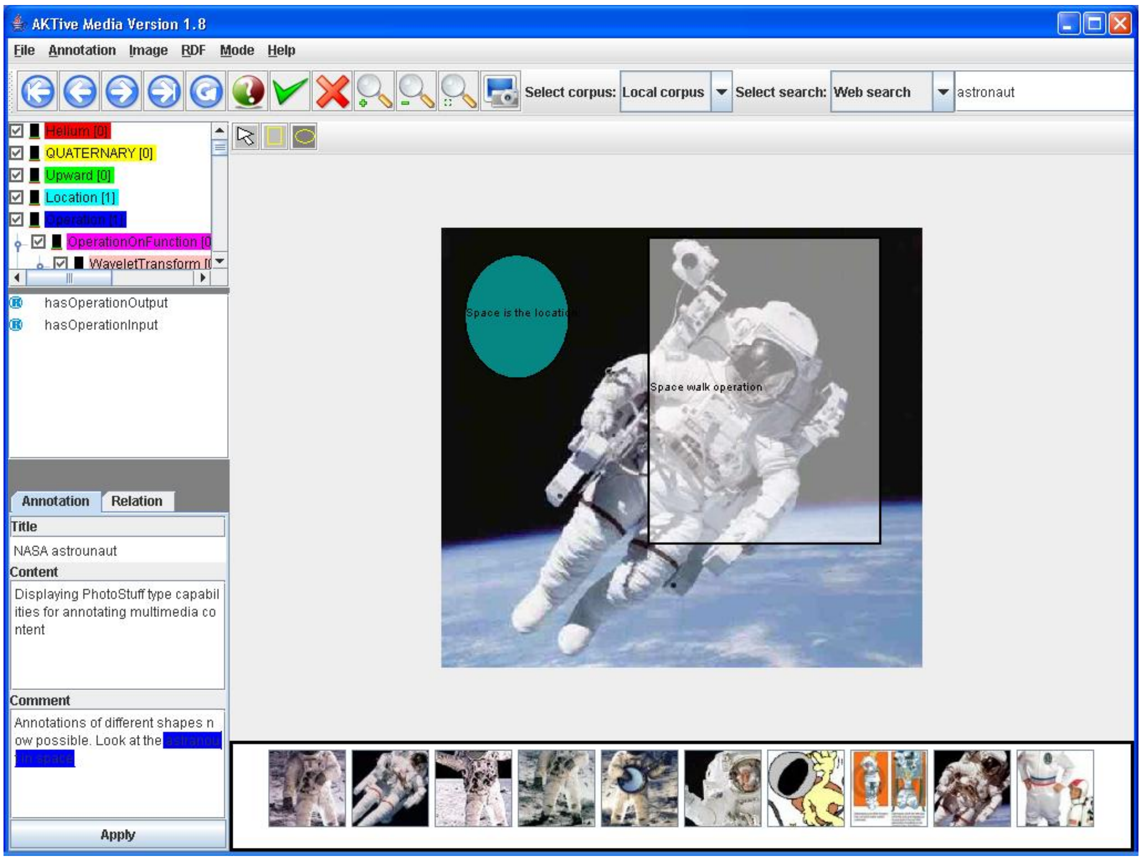
1.2.3 AKtive Media

AKtive Media[12] [CLC06] is an ontology-based cross-media annotation system, supporting both images and text. Two screenshots illustrating image and text annotation respectively are shown in Figures 1.11 and 1.12. The goal is to automate the process of annotation by suggesting knowledge to the user in an interactive way while the user is annotating, hence minimising user effort. The system actively works in the background,

[12] *http : //www.dcs.shef.ac.uk/ ajay/html/cresearch.html*

interacting with web services and querying a central annotation store to look for context-specific knowledge in order to make suggestions to the user.

It provides the means to import multiple ontologies in RDFS, DAML, and OWL, and support for various image formats (JPG, GIF, BMP, PNG, TIFF), and for EXIF metadata extraction. Image annotation is performed at region level, i.e. the user selects and annotates specific regions of an image. Additionally, batch annotation is enabled, where a user can annotate an entire collection of images at the same time. Apart from concept-based annotation, relational annotations are also provided, e.g. to represent that the instance depicted in a region is related to another through a partonomic relation.



Select corpus: Local corpus
Select search: Web search
astronaut

Title
NASA astrounaut

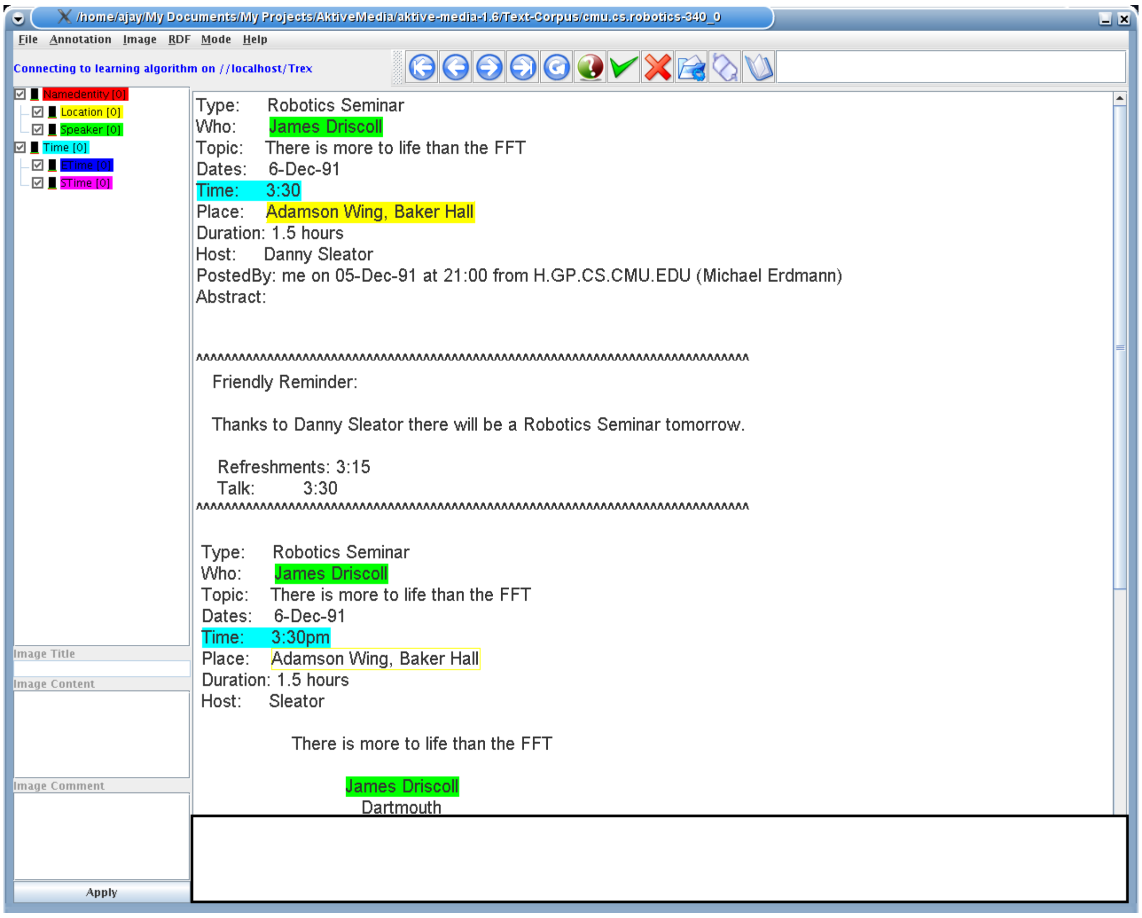
Content
Displaying PhotoStuff type capabilities for annotating multimedia content

Comment
Annotations of different shapes now possible. Look at the astronaut in space

Figure 1.11: Image annotation with AKtive Media

Region selection in images is performed via draw shapes, which the user drags to the preselected domain concept. Additional annotations, addressing the image as a whole, can also be included. Furthermore, the text used itself for region/image annotation can be in turn annotated utilising AKtive Media's text annotation facilities. Having annotated 5 – 10 texts or HTML documents, AKtive Media provides learning facilities for subsequent information extraction that can ease the text annotation process, removing some of the annotator burden through suggestions.

The produced annotation metadata are represented in RDF, while no masks are used


/home/ajay/My Documents/My Projects/AktiveMedia/aktive-media-1.6/Text-Corpus/cmu.cs.robotics-340_0
File Annotation Image RDF Mode Help
Connecting to learning algorithm on //localhost/Trex
Namedentity [0]
Location [0]
Speaker [0]
Time [0]
ETime [0]
STime [0]
Image Title
Image Content
Image Comment
Apply

```
Type:     Robotics Seminar
Who:      James Driscoll
Topic:    There is more to life than the FFT
Dates:    6-Dec-91
Time:     3:30
Place:    Adamson Wing, Baker Hall
Duration: 1.5 hours
Host:     Danny Sleator
PostedBy: me on 05-Dec-91 at 21:00 from H.GP.CS.CMU.EDU (Michael Erdmann)
Abstract:


^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  Friendly Reminder:

  Thanks to Danny Sleator there will be a Robotics Seminar tomorrow.

   Refreshments: 3:15
   Talk:         3:30
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Type:     Robotics Seminar
Who:      James Driscoll
Topic:    There is more to life than the FFT
Dates:    6-Dec-91
Time:     3:30pm
Place:    Adamson Wing, Baker Hall
Duration: 1.5 hours
Host:     Sleator

               There is more to life than the FFT

                       James Driscoll
                         Dartmouth
```

Figure 1.12: Text annotation with AKtive Media

for localisation of annotated regions. Two types of persistent storage are offered for the produced RDF graphs: the Local Store and the Central Triple Store. Furthermore, the RDF import and export facilities allow the user to export the annotated data to RDF for later access or for publishing this information to the semantic web.

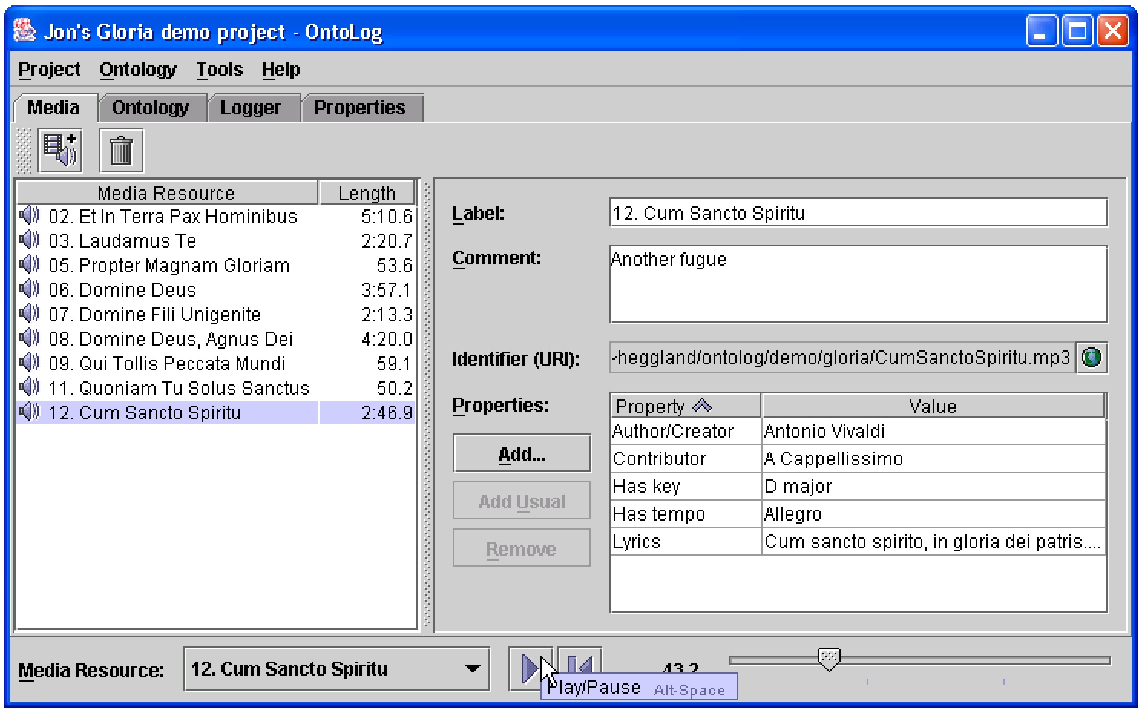
1.2.4 OntoLog

OntoLog[13] [Heg05] is a tool for annotating and indexing video and audio content using ontologies. It is designed to be flexible and adaptable. OntoLog provides two sets of vocabularies as default: the Dublin Core Element Set v1.1 (DCES), and the Dublin Core Element Set Qualifier Library.

As illustrated in Figure 1.13, in the Media panel the user can select the media items they want to annotate with respect to the loaded ontology, and play/pause them. In the Ontology panel, the user can view and edit the ontologies and their properties. Ontolo-

[13] $http://www.idi.ntnu.no/~heggland/ontolog/$

gies are treated as hierarchical collections of concepts relevant to some domain, and are used for two purposes: to annotate the media resources by connecting concepts to media intervals – e.g. associating the Soprano concept with all the intervals where the sopranos in the choir are singing – and to describe the media resources and ontology concepts with properties and values from the ontologies.



Media Resource	Length
02. Et In Terra Pax Hominibus	5:10.6
03. Laudamus Te	2:20.7
05. Propter Magnam Gloriam	53.6
06. Domine Deus	3:57.1
07. Domine Fili Unigenite	2:13.3
08. Domine Deus, Agnus Dei	4:20.0
09. Qui Tollis Peccata Mundi	59.1
11. Quoniam Tu Solus Sanctus	50.2
12. Cum Sancto Spiritu	2:46.9

Label: 12. Cum Sancto Spiritu

Comment: Another fugue

Identifier (URI): -heggland/ontolog/demo/gloria/CumSanctoSpiritu.mp3

Properties:

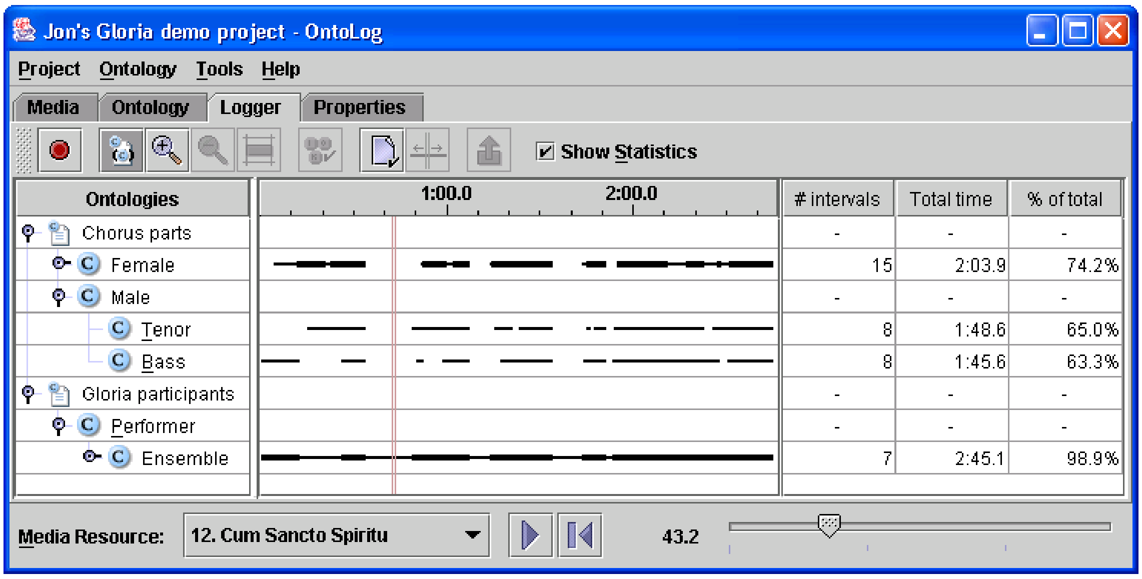
Property	Value
Author/Creator	Antonio Vivaldi
Contributor	A Cappellissimo
Has key	D major
Has tempo	Allegro
Lyrics	Cum sancto spirito, in gloria dei patris....

Media Resource: 12. Cum Sancto Spiritu

Figure 1.13: Media annotation with OntoLog

The Logger panel, illustrated in Figure 1.14, is where the temporal intervals to which annotations are attached are created, viewed and manipulated. On the left the ontology hierarchy is shown, while on the right there is a timeline display, showing the annotation intervals as horizontal black lines. The thickness of lines corresponds to the number of concepts in the ontology that are active at any given moment. The Logger panel can also show some simple statistics on the intervals: i) the number of intervals per concept, ii) their total length, and iii) the percentage of this length with regard to the total length of the media resource.

Ontolog additionally allows the user to create, view and edit the ontologies at hand. It should be stressed that within OntoLog, ontologies are treated mostly as hierarchical vocabularies. Subclass and domain/range notions are included, however no control is provided over the proper use of their semantics. The ontology language supported is RDFS, as is the format of the produced metadata.



Jon's Gloria demo project - OntoLog

☑ Show Statistics

Ontologies	# intervals	Total time	% of total
Chorus parts	-	-	-
Female	15	2:03.9	74.2%
Male	-	-	-
Tenor	8	1:48.6	65.0%
Bass	8	1:45.6	63.3%
Gloria participants	-	-	-
Performer	-	-	-
Ensemble	7	2:45.1	98.9%

Media Resource: 12. Cum Sancto Spiritu 43.2

Figure 1.14: The Logger tab in OntoLog

1.2.5 Multimedia tools categorisation

In this section, we discuss some key characteristics of ontology-based multimedia annotation tools, such as the type of content supported and the granularity of annotations handled, so as to provide some guidelines for their proper use. The criteria follow on the related study conducted within the $W3C$ Image Annotation Taskforce[14]. Using these characteristics as criteria, users can be guided so as as choose the most appropriate tool with respect to the application under consideration.

Type of content

A tool may support the annotation of different kinds of content, such as image, text (including image captions, subtitles, etc.), video, and audio, as well as composite types such as multimedia presentations and web pages.

Type of metadata

Annotation of multimedia content may refer to different types of metadata [vONH04], accounting for the multiplicity of multimedia content semantics. Annotations may include descriptive (e.g. an image depicts Mona Lisa), structural (e.g. a video is composed of three shots), administrative (e.g. creator, location and privacy metadata), low-level feature information (a region color in RGB), etc., based on the application functionality for which they are intended, such as retrieval, navigation, management, and analysis respectively.

[14] $http://www.w3.org/2005/Incubator/mmsem/XGR-image-annotation/$

Metadata format

The metadata format is of significant importance, since it determines to a great extent the interoperability, and thus the value, of the produced annotations. MPEG-7, reflecting the experience of the multimedia analysis community, is the representation most often used for exchanging automatic analysis results, whereas RDF and OWL are favoured in the Semantic Web context. Efforts to couple these two include the initiatives towards building multimedia ontologies and alignment frameworks under core ontologies [TPC07, DHL03, Hun01].

Granularity level

Granularity characterises whether an annotation refers to a multimedia content item as a whole or if it is segment-based. This is an important characteristic, since different applications pose different requirements with respect to content structure significance. For example, from a retrieval perspective, it is useful to provide annotations for individual video segments (e.g. a goal shot) or individual image regions (e.g. a person), especially when delivery to limited resource devices is considered or for applications like summarisation.

Annotation Editing

Annotations are not necessarily static, but may need to be modified during their lifecycle. For example, a video that was initially annotated at shot level, might need at subsequent time to be annotated at frame level as well to allow for minimal bandwidth when broadcasting highlights in mobile devices (i.e transmit the few directly relevant frames instead of the complete set of frames that comprise the shot). Furthermore, a user may be forced to use different tools, e.g. due to different functionalities provided, due to change in license conditions, etc.

Collaborative vs individual

Annotation can be performed independently by individual users, or within a collaborative framework. This is of particular importance for applications addressing large quantities of shared content or for communities where reaching consensus through such an collaborative procedure is important.

Client-side requirement

This dimension considers whether a Web browser can be used for performing the annotation or if it is implemented as a stand-alone application that needs to be locally installed.

Feature	*M-Ontomat-Ann.*	*Photostuff*	*Aktive-Media*	*OntoLog*
Content Type	Image Video	Image RDF	Image Text,HTML	Video Audio
Ontology Language	RDFS DAML,OIL	RDFS OWL	RDFS DAML,OWL	RDFS
Metadata Type	Low-level features	Descriptive Administrative	Descriptive	Descriptive
Metadata Format	RDF	RDF	RDF	RDF
Granularity Level	Region	Region Image	Region	Video/Audio Segment
Collaborative vs Individual	Individual	Individual	Individual	Individual
Client-side vs stand-alone	Stand alone	Web based	Stand alone	Stand alone
License	free	free	free	free
Access control	no	no	no	no

Table 1.1: Multimedia annotation tools categorisation

License conditions

Depending on the intended application context, whether a tool is open source, distributed under educational license, or commercial can be of importance, comprising another factor in the selection process.

Access control

Different applications have different requirements with respect to the levels of access granted to different groups of users.

Table 1.1 summarises the characteristics introduced previously which are rather specific to multimedia tools.

Chapter 2

Usability, Accessibility and Interoperability

2.1 Introduction

In this chapter, we describe and discuss the evaluation of the annotation tools previously described in Chapter 1, in terms of functionality such as usability, interoperability and accessibility issues. In Chapter 5, we shall discuss some evaluations performed with respect to actual performance levels. We leave the discussion of scalability to Chapter 6. In this first evaluation, it would have been ideal to perform a user-based experiment whereby a group of testers could evaluate all the tools and form a comparison between them. However, this was not really feasible due to the fact that it would have been a very time-consuming task for people to undertake, and due to the difficulties in installing so many tools and training people to use them. Instead, we describe our own experiences with the tools, augmented by comments from other users (for example, we asked a group of GATE users how easy they thought it was to install GATE, and took this into consideration when attributing a score). Although this was not an extensive survey, it seemed to be sufficient to provide a reliable set of answers, since the users were taken from a range of skill levels and there were few discrepancies of opinion. Table 2.1 depicts the results of this assessment for textual tools,, while Table 2.2 shows the results for multimedia annotation tools..

The version tested was the version available at the time when the testing was carried out. Some of the annotation tools are in constant development, such as GATE, KIM, MAGPIE and Beagle++, while others (MnM, Ontomat) are – to our knowledge – not being further developed. It is possible that at the time of release of this report, further versions of the annotation tools are available which address issues raised in the experiments. For example, GATE is currently in version 4.0, which addresses some bugs and offers greater functionality such as new processing resources and faster performance.

Tool	MAGPIE	MnM	KIM	OntoMat	GATE	Beagle++
Version tested	v1.0	v1	v1.05	0.8a	v3.0	v1.0
Interoperability						
Platform	W,L,M	W,L	W	W,L,M	W,L,M	L
Browser	IE,MF	own	IE	own	n/a	MF,KON
Browser variation	yes	n/a	n/a	n/a	n/a	no
Data format	HTML,TXT	HTML,TXT	All	HTML,TXT	All	All
Ontology format	own	DAML+OIL, RDF	OWL Lite Lite	OWL Lite	OWL, RDF	RDF/S
Converters	no	no	no	no	yes	no
Source available	no	no	no	yes	yes	yes
Usability						
Installation ease	+2	+1(Win)	+2	+2	+2	+2
Installation doc	-1	0	+1	0	+1	+1
Doc quality	+2	-1	+2	-1	+1	+1
Doc format	SI,I,M	TI	TI,I,M	TI,I	TI,I,M	SI,TI
Linked help	no	no	yes	no	yes	no
Configuration	+2	-1	+1	+2	-1	+2
Aesthetics	+2	-1	+2	-1	+2	+2
Range of tasks	+1	-1	+1	+1	+2	+1
Accessibility						
Graphics	+2	+1	+1	0	0	+2
Tooltips	yes	no	some	some	yes	no
Mouse alternative	no	no	no	some	no	no
Changeable colours	yes	yes	n/a	yes	yes	yes*
Changeable menu fonts	no	no	no	no	yes	yes*
Changeable text fonts	n/a	no	annots only	no	yes	yes*

Table 2.1: Assessment of functionality of textual annotation tools

Tool	M-Ontomat Annotizer	PhotoStuff	AKtive Media	OntoLog
Version tested	v0.60	v3.33 beta	-	v1.9
Interoperability				
Platform	W	W	W	W,L,M
Browser	n/a	n/a	n/a	n/a
Browser variation	n/a	n/a	n/a	n/a
Ontology format	RDF,DAML+OIL	RDF,OWL	RDF,DAML,OWL	own
Converters	no	no	no	no
Source available	no	yes	yes	no
Usability				
Installation ease	+2	+2	+2	+2
Installation doc	+2	+2	+2	+1
Doc quality	+1	+2	0	0
Doc format	SI,I,TI	SI,TI	SI,I,M,TI	SI,I,TI
Linked help	TI	TI	TI	TI
Configuration	n/a	n/a	n/a	n/a
Aesthetics	+2	+2	1	0
Range of tasks	+1	+1	+1	+1
Accessibility				
Graphics	+2	+2	+1	+1
Tooltips	no	no	no	no
Mouse alternative	no	no	no	no
Changeable colours	yes	no	no	no
Changeable menu fonts	no	no	no	no
Changeable text fonts	no	no	no	no

Table 2.2: Assessment of functionality of multimedia annotation tools

2.2 Interoperability issues

Interoperability is concerned with how well the tool interacts with other tools and systems. Annotation is a task that is often combined with other applications, such as browsing, search and retrieval, indexing, etc., so it is important that annotation tools can easily interact with other systems. This is best achieved by conformance to existing standards.

Textual annotation tools add semantic metadata to text. If these annotations are to be reused by other tools, they need to be represented in some standard format that can be interpreted by other tools. There are essentially two ways in which metadata can be represented: as inline or standoff annotation. Inline annotation means that the original document is augmented with the metadata information, i.e. the text is actually modified. Standoff annotation, on the other hand, means that the metadata is stored separately from the original document, with pointers to the location of the corresponding text in the document. This can be either in the form of a database or as e.g. an XML file. For ontology creation or enhancement, standoff annotation method is generally much better, because the text itself is unimportant, rather it is the information gleaned from the text that is interesting. Also because the original text is left unmodified, it can be reused.

Both methods are acceptable from the point of view of interoperability; however, standoff annotation is generally preferable, for the reasons mentioned above, as long as a standard form is used, such as TIPSTER format [Gri95], or provided that a means of export to a recognised format is provided. This is the problem with inline annotation, because it is difficult to manipulate the annotations once created.

Interoperability evaluation not only covers annotation format, but also issues such as:

- data format: what kinds of text format can be processed, e.g. xml, html, sgml, txt, etc.;
- annotation schemes: whether annotation schemes can be imported/exported from other tools;
- plugins: if it is possible to plug in other tools and applications;
- converters: if converters to/from other formats are provided if non-standard formats are used;
- API: whether the tool provides some API to programmatically access it.

Under the topic of interoperability we investigate factors such as which platforms and browsers the tool runs on, ontology formats possible and how easily the tool can be modified. Responses are either in yes/no format or specify the answer exactly (e.g. the tool runs on Internet Explorer).

It must be noted, however, that for multimedia annotation tools, the annotation format and the related dimensions are not sufficient alone for determining the interoperability of the produced annotations, i.e. the interoperability among the different tools. As was briefly described in Chapter 1, two main factors are: i) the media aspects involved and ii) the user vs analysis orientation of the annotation. The first requires common underlying models for the representation of such information, or for appropriate mappings between the different implementations. This requirement poses significant challenges, as on the one hand some tools use masks for the representation following naming conventions, while on the other hand, the existing multimedia ontologies build upon varying rationales, introducing semantic interoperability issues. The second aspect, refers to the different kinds of functionalities required with respect to the corresponding annotation application context. Annotations intended to assist in analysis usually include a number of low-level features that are of no use when targeting end-user annotation applications. Thus a tool meant to support descriptive annotations only, such as PhotoStuff, cannot by definition process/respond to annotations produced by a tool such as M-Ontomat-Annotizer without information loss.

2.2.1 Platform

This factor considers which platforms the tool can be run on, according to the documentation and/or discussion with the providers. We consider only Windows (W), Linux (L) and Mac (M), as these are the 3 main platforms most likely to be used. All tools worked with both Windows and Linux apart from KIM which only worked with Windows, and Beagle++ which only works with Linux using the Gnome or KDE desktop environments. GATE, OntoMat and MAGPIE also worked with Macs. With respect to multimedia tools, all work with Windows only, apart from OntoLog which is platform independent.

2.2.2 Browser

Here we consider which browsers the tool works with: Internet Explorer (IE), Mozilla or Mozilla Firefox (MF) or its own proprietary browser (own). Here MAGPIE was the only tool which worked with both IE and Firefox. KIM worked only with IE, while MnM and OntoMat have their own proprietary browser and GATE does not use a browser as such but imports the documents into its own interface (which is not browser-like). GATE can, however, be run via its API as a web service on any browser. Because Beagle++ only runs on Linux, it works with Mozilla Firefox (MF) and Konqueror (KON), since IE does not operate on Linux. The browser, and subsequently the browser variation, dimensions are not applicable for the case of multimedia tools, as they are implemented as standalone applications.

2.2.3 Browser variation

This question looked at if and how there are any differences when the tool is run with different browsers. This only therefore applied to MAGPIE as the other tools only work with a single browser. It was found that there are quite a few differences with different browsers: when running on Firefox the performance was slightly substandard and there were some unexpected happenings; there were also conflicts between browser and plugin commands on Firefox. It is expected that later versions of MAGPIE will not have this problem, however.

2.2.4 Ontology format

This looks at which ontology formats are compatible with the tool. Note that this again is constantly changing as OWL becomes more and more popular, and that some of these tools were first developed some years ago when DAML+OIL and RDF were more de rigueur. All the tools are compatible with OWL except for MnM (which is an older tool and no longer supported) which uses DAML+OIL and RDF, GATE is also compatible with RDF. At the time of testing, MAGPIE used its own proprietary ontology format but plans are to use OWL and other formats in future versions.

The benchmarking of OWL interoperability of semantic web tools is covered in a separate deliverable [GCDPG07], so we only summarise the results in this deliverable. Only GATE and KIM out of the textual annotation tools could be evaluated under this framework since the other annotation tools do not use OWL ontologies or have other quirks which make them impossible to be evaluated in this way. Furthermore, both GATE and KIM share the same ontology API so their results on the interoperability benchmarking would be identical (only GATE was actually evaluated). GATE itself scored quite highly on the interoperability benchmarking, interacting well with tools such as Protege, KAON, JENA and SWI-Prolog, although it fell down in a couple of places, such as not creating all the instances correctly.

Similarly, all multimedia tools are compatible with OWL, apart from OntoLog which supports only RDF(S). The reader should note however that the input ontology format in the multimedia case is not necessarily the same as the produced metadata format. This is a direct consequence of the multimedia annotation process, at least as implemented currently. The input ontology serves mainly as controlled vocabulary for instantiating concepts (and relations as for example in Aktive Media) present in the annotated content. In practical terms, this means that the tools by definition cannot exceed the expressivity of the loaded ontology, as they can produce only simple concept and relations assertions (e.g. no reification, no property restrictions, etc.). This is why most tools specify RDF as their output metadata format, and separate this from the ontology language support provided for the loaded ontologies.

2.2.5 Converters

Here we investigate whether any converter is provided for the ontology formats supported into other formats. None of the tools provided converters except for GATE which allows conversion between the two formats it supports.

2.2.6 Data format

Here we look at what format the text needs to be in for the tool to be able ot process it. GATE scores highly here by being able to process many kinds of textual format, including plain text, SGML, XML, HTML, RTF, Word, and PDF (though some of the document rendering may be imperfect on the latter three types). KIM also can process the same formats as GATE as it uses the functionalities from GATE3.1 for document format analysis. The other tools only really process texts that can be displayed in a browser, i.e. HTML, plain text, XML etc. With respect to multimedia tools, one could say that in principle all of the most frequent image, video and audio formats are usually supported. This is due to the relative straightforwardness of converting between formats and the variety of existing libraries for reading and processing the different formats.

2.2.7 Source available

This question looks at whether the source code is freely available so that developers can extend or modify the tool as required, for example adding new annotation sets, new visualisation capabilities, new processing resources, etc. This is a very important part of the flexibility and extensibility of the tool, since as discussed in Section 7.1, if the type of user and tasks for which the tool is to be used remain unknown or unpredictable at the time of design, then the tool needs to be able to cater for such flexibility if it is to fulfil interoperability requirements and be widespread in its use. The only tools of those tested that, to our knowledge, have this capability are OntoMat, GATE and Beagle++; all the others are released as black boxes. Although all examined multimedia tools are available under public license conditions, only PhotoStuff and Aktive Media make the source code publicly available.

2.3 Usability issues

Under the topic of usability, we categorise the quite broad issues such as documentation, ease of setup and installation, aesthetics of design, and range of tasks possible. Some more specific issues concerning accessibility are categorised separately in the following section. Responses in this section are mostly allocated a score ranging from -2 to +2. The lowest score (-2) generally reflects a "very poor" answer, -1 reflects "poor", 0 reflects

"OK", +1 reflects "good" and +2 reflects "very good". In the case of questions such as "setup" the scores could be interpreted as ranging from "very difficult" through to "very easy".

2.3.1 Installation ease

First we look at how easy the tool is to install. While this is clearly related to the following question about the installation documentation, the two may be orthogonal as the installation documentation could be very poor but installation may still be very easy. Indeed we can see this from the results, for example MAGPIE was deemed very easy to install (+2) but the installation documentation was deemed poor (-1). In fact, all tools were deemed very easy to install except for MnM, which was deemed as easy to install for Windows but not for Linux. Installation was straightforward for all multimedia tools.

2.3.2 Installation documentation

As mentioned above, the quality of the installation documentation did not necessarily correlate with the ease of installation of the tool. Note also that the installation documentation was evaluated separately from the general tool documentation or user guide. Sometimes the two were part of the same document and sometimes they were separate documents. Although these two did have a slightly stronger correlation, there were still some differences. MAGPIE was the only tool to have a negative score for installation documentation, which correlates with the difficulty of installation. MnM and OntoMat were both deemed satisfactory, and GATE and KIM were deemed good though still with room for improvement. In the case of KIM, GATE and OntoMat, the quality of the installation documentation turned out not to be that important since the tools were easy to install, though had the quality of the documentation been lower this might have been a different case.

2.3.3 Documentation quality

We then turned our attention to the quality of the main documentation (excluding installation instructions). Here we look first of all at the quality in general, and then we look at some aspects more specifically (the format of the document and how easy it is to find help). MAGPIE and KIM had very good documentation, which was clear, comprehensive and easy to follow. MnM's documentation was clear but very basic and not detailed enough, giving it a score of -1, while KIM's documentation was not very clear and also given a score of -1. GATE's documentation was very comprehensive but quite confusing and the users found it hard to locate the relevant information because it was so detailed. This may be linked with the fact that GATE is a much more extensive and powerful tool than the others, with many more functionalities available, and because it is also possible for programmers to edit the source code and add new components, this makes it a much more complex tool requiring more detailed documentation. All multimedia tools have quite detailed accompanying documentations. This is justified to a great extent by the general unfamiliarity of the multimedia community with SW technologies, and of course the complexity this entails at first encounter.

2.3.4 Documentation format

Here we look at the format of the documentation. This links very closely to the quality of the documentation, showing very clearly that the more modalities in the documentation, the higher the quality (though this need not be the case). We investigated 4 aspects of the document format: whether it had step-by-step instructions (SI), images/screenshots (I), movies/demos (M) or just simple textual instructions (TI). MAGPIE was the only tool which had clear step-by-step instructions, and also had both images and movies. OntoMat and GATE both had textual instructions, images and movies, while KIM had textual instructions and images, and MnM just had textual instructions. Clearly the combination of all text, movies and images was the clearest for users, although step-by-step instructions were not necessarily an improvement on simple basic instructions (both MAGPIE and Ontomat scored +2 on documentation quality although only MAGPIE had the step-by-step instructions). As with the installation instructions, however, a tool might be easy to use even though the instructions are poor, so we cannot draw too many conclusions about the instruction quality. All multimedia tools provide step-by-step guidance, including screen shots that further enhance readability. AKtive Media provides additional tutorial videos. Compared with the other tools, we observe that Aktive Media tutorial videos compensate for the less detailed screenshots and stepwise description. Thus, the documentation format of all the multimedia tools can be considered of equal quality.

2.3.5 Linked help

A final part of the instruction examination looked at whether there was a help facility available directly from the tool. This could be either just a link to the documentation (as in the case of GATE) or a specific help function similar to most proprietary programs. While we would not expect a full widget facility (such as the Windows paperclip), it is very useful to have direct access to help without having to revert back to the website or downloaded instructions somewhere in the user's file system. MAGPIE had a (not very obvious) button linking back to the website, from which the user guide could be found, but no direct link to a help facility. KIM and Beagle++ had no help facility, while OntoMat and GATE both have linked help available: GATE links directly to the user guide via a button on the menu.

2.3.6 Configuration

The configuration criterion looked at how easy the tool was to set up in the way that the user wanted. This does not include installation, but rather things like changing the appearance of the GUI to suit the user's needs, changing options such as whether the tool should save session on exit, layout of menus, different skins or "look and feel", altering the fonts, etc. It does not consider the presence or absence of such options as such, but is based on how easy it is to set up the tool according to the options given, i.e. how easy these options are to use. MAGPIE and OntoMat were deemed very easy to set up (possibly because not so many options were available), while KIM was deemed fairly easy. MnM was considered quite hard to set up because of unnecessary and confusing dialogues, while GATE was deemed quite hard simply because many options were available and it was not always clear what they did, for example "Add space on markup unpack if needed" is not immediately clear to a new user unless they have used GATE and are familiar with some of the problems with respect to offsets. While the more complicated options are in the "Advanced" section, some more common and straightforward options (such as "Save session on exit") are also included in this section. Some options are really not obvious (such as changing the colour of annotations) without reading the user guide, both in terms of whether or not it is possible, and in terms of how to do it.

The configuration options provided for the examined multimedia tools are very poor. With the exception of M-Ontomat-Annotizer, which allows some very basic configuration with respect to choosing the colour of the drawing tools for specifying a region, the tools do require or allow for any set up, after the installation.

2.3.7 Aesthetics

This question looked at how pleasing the tool was generally for the user, in terms of appearance, attractiveness etc. MAGPIE, KIM and GATE all scored highly here, and were considered to be colourful and interesting. MnM and OntoMat were rated quite poorly, and considered to be dull and unappealing. The aesthetics of a tool seems to be quite highly linked with the overall goals of the tool: MnM was designed as a very simple tool for a fairly limited range of tasks, and therefore it seems that not much consideration was given to its attractiveness. GATE on the other hand is designed for a very broad range of users and applications, and since one of its objectives is to be used as widely as possible, much consideration has gone into the look and feel of the tool and how much people will enjoy using it. Generally, the aesthetics of the examined tools are considered quite high, and seem to match user intuition. OntoLog is the only one that falls down a bit, due to the more simplistic implementation it follows.

2.3.8 Range of tasks

This question looked at how complex the tool was and at how wide a range of tasks could be achieved. All tools were classified as having a fairly wide range of tasks except for MnM, while GATE was considered to have a very wide range (particularly since it can easily be extended to incorporate new Processing Resources doing all manner of new things). The range of tasks along the examined multimedia tools is quite evenly distributed. The main difference is observed between M-Ontomat-Annotizer that allows for low-level annotations, and the rest that target user ones. This differentiation however is enforced by the intended application usage, and as the reader can observe, it is not reflected in parts of functionalities related to media management such as region selection.

2.4 Accessibility

Software accessibility is essentially about making tools that are usable, easy to use and attractive to use for everyone (not just for people with disabilities) [May05a]. Generally, however, designing websites and software with certain disabilities in mind covers the majority of cases for people with and without disabilities. Particular care should be taken to design sites and systems usable by the following categories: blind and colour-blind people, people with other visual impairments (e.g. partially sighted); deaf people, people with motor problems (e.g. those who cannot use a mouse or normal keyboard), dyslexic people, people with learning difficulties, people with epilepsy (who may not be able to tolerate flashing colours, for example).

Obviously not all categories need to be considered, depending on the tool and intended user, but care should be taken not to exclude potential unknown users. For example, one might not imagine that a blind person would want to use an annotation tool, but one cannot be sure of this in advance. It is also important not to stereotype certain categories of disability. For example, making sure that tools work with a screen reader will not necessarily benefit all blind and partially sighted people – they may also require easy navigation, clear and simple layouts without clutter, consistency, good use of colour, changeable font sizes, etc.

Some of the most important examples of accessibility problems stem from *inflexibility*. A well designed tool will have options to change the user's preferences regarding colours, layout, font sizes and styles, and so on, and the ability to save and restore latest sessions, etc.

Even though a user should be able to choose such options, the default options should also be well designed. For example, text should be in a mixture of upper and lower case where possible (as this is the most easily readable, hence the reason it is used for road signs), and colour schemes should incorporate dark writing on a light background or vice versa. Icons should be clearly understandable, not just with alternative text on mouseover, but should also use clear symbols and be large enough to click on easily (for those with motor or sight problems). Mouse alternatives should also be widely available, again for people with motor and sight problems, RSI etc.). In the remainder of this section, we look at some specific aspects of accessibility and investigate to what extent they are addressed in the tools.

2.4.1 Graphics and tooltips

This question looks at how graphics are displayed in the tool: for example whether icons are clearly displayed and also intuitive, and whether tooltips are provided for all images such as icons and symbols (for those users who have difficulty identifying or understanding the graphics). MAGPIE scored the best on this question, having clear images and tooltips, wile MnM and KIM both scored +1: MnM having clear icons but no tooltips, and KIM having only some clear icons and tooltips. OntoMat and GATE both scored O here as neither had very clear icons and while GATE does have tootips, OntoMat only has some tooltips. For example, GATE uses the ? symbol normally associated with a Help function to represent Plugins, which is not at all intuitive and can be quite misleading, unless the user waits for the (quite slow) tooltip to appear before clicking on it. As expected, all multimedia tools score particularly well on this aspect.

2.4.2 Mouse alternative

In this question we look at whether alternatives to a mouse can be used to input data (e.g. create and edit annotations, etc.). With the increasing incidence of RSI (Repetitive Strain Injury) as well as for users with a disability such as motor or sight problems, it is important that users are not restricted to a particular kind of HID (Human Input Device) such as a mouse and can, for example, use a keyboard for all functions instead. Most of the tools fall down in this category, which is understandable given that the easiest way to annotate a piece of text is to select it with the mouse and then click something. However, it would still be possible to have an alternative keyboard mechanism for selecting and annotating. OntoMat partially has this facility, while the other tools do not have any mechanism for using mouse alternatives. Unfortunately, the same observations hold for multimedia tools as well, since the most intuitive way to select a given image region or temporal interval is by doing some mouse drawing and clicking. It is worth noticing that moving to mouse alternatives for multimedia presents some additional challenges related to the limitations of automatic segmentation that results in over or under segmented regions. In cases where strict boundaries are required, manually defined regions are necessary.

2.4.3 Colours

The last 3 questions in this section look at the options that the user can set in order to customise the GUI to his/her preference. First we look at whether the user can select their own colour preferences. Magpie, MnM, OntoMat and GATE all allow the user to change the colours, although some of the mechanisms for changing the annotation colours are not very obvious. For KIM, the option is not applicable. Beagle++ relies on the settings from the desktop environment for colours and fonts, such as KDE or Gnome, so these can be changed, but not independently from the general desktop settings. With respect to customisation, not many options are provided in the case of multimedia tools. M-Ontomat-Annotizer is the only one that allows the user to change the colour of highlighted image regions.

2.4.4 Menu fonts

This looks at whether the user can change the size and style of the menu fonts in the tool. This is not possible with any of the tools except GATE, which has a comprehensive range of styles available. The same holds for multimedia tools.

2.4.5 Text fonts

This looks at whether the user can change the size and style of the text fonts, e.g. the fonts of the loaded document. This is not really applicable in MAGPIE as it uses a regular browser. MnM and OntoMat do not permit changing the text fonts, whereas with KIM the user can change the font size of the annotations only. GATE allows the user to select a wide range of font sizes. Again, such customisation is not possible with any of the examined multimedia tools.

2.5 Summary

In this section, we have examined some features of the tools relating to issues other than performance, such as usability, accessibility, and so on. It is clear that there are issues still to be resolved with most tools before one could claim that they are completely user friendly for a mass market; however, these are all primarily designed as research tools. In order to scale to industrial use, it is quite important for the developers to look more closely at some of the issues highlighted here. For example, Beagle++ only works with Linux, which restricts its potential user base quite considerably. Of the textual annotation tools, GATE is used the most widely, in various incarnations, and this is supported by the results of the experiments, since it has good interoperability, good usability and supports a wide range of tasks, compared with the other tools which are somewhat lacking in one or more of these areas.

Chapter 3

Metrics for Evaluating the Performance of Annotation Tools

3.1 Introduction

In this chapter, we discuss methods of measuring annotation performance. This includes both traditional metrics such as Precision and Recall, and some newer metrics developed especially for ontology-based information extraction in order to provide a more informed analysis of ontology-based tools. We describe also some experiments we have designed to test the validity of these new metrics and to compare them with traditional metrics, and show that for ontology-based annotation, such metrics are more useful. However, such metrics are not always easily applicable because they rely on having an appropriate hand-crafted gold standard which can be time-consuming to create, and inter-annotator agreement may be much lower than for non-ontology-based gold standards. As will be discussed in Chapter 5, it was not always possible to use the ontology-based metrics in our evaluation of annotation tools: in this case we fall back on the traditional Precision and Recall metrics. In Chapter 4, we continue discussion of evaluating performance by describing an investigation into the suitability of a thesaurus for automatically annotating a given document set. This includes some further analysis of similarity metrics, including the one described in the current chapter for comparing the system results with the gold standard, used in the BDM metric.

3.2 Balanced Distance Metric

The Balanced Distance Metric has been developed by the University of Sheffield as part of work carried out in D2.3.6 (Prototypes of language dependent tools for evaluation) [MPSC06]. In this deliverable, we discussed the need for a new evaluation metric for dealing with ontology-based information extraction techniques for semantic annotation and ontology population, and described the metric, along with a discussion of other existing metrics. Here we therefore only present the metric itself briefly, but describe in more detail its comparison with other metrics and discuss its suitability for evaluating annotation tools.

Traditionally, information extraction systems have been evaluated using Precision and Recall, which classifies each entity returned by the system as either correct or incorrect. However, this is not sufficient for ontology-based information extraction, because the distinction between correct and incorrect is more fuzzy: if an answer is closely related to the correct answer, then some credit should be given for an "almost correct" answer, rather than simply classifying it as wrong. So a metric which classifies the correctness of an answer based on its semantic proximity to the real answer should give us a fairer indication of the performance of the system. Other existing cost-based or distance-based metrics, such as Learning Accuracy (LA) [HS98a], have some flaws such as not taking into account the density of the hierarchy, and in the case of LA, being asymmetrical (see [MPSC06] for a more complete discussion of this).

The BDM [May05a, MPL06a] computes semantic similarity between two semantic annotations of the same token in a document. The metric has been designed to replace the traditional "exact match or fail" metrics with a method which yields a graded correctness score by taking into account the semantic distance in the ontological hierarchy between the compared nodes. These nodes are called Key and Response. The final version of the BDM is a slightly improved version of the original BDM described in the deliverable D2.3.6, which did not take the branching factor into account (as described below).

The BDM is computed on the basis of the following measurements:

- CP = the shortest length from root to the most specific common parent, i.e. the most specific ontological node subsuming both Key and Response)
- DPK = shortest length from the most specific common parent to the Key concept
- DPR = shortest length from the most specific common parent to the Response concept
- n1: average chain length of all ontological chains containing Key and Response.
- n2: average chain length of all ontological chains containing Key.
- n3: average chain length of all ontological chains containing Response.
- BR: the branching factor of each relevant concept, divided by the average branching factor of all the nodes from the ontology, excluding leaf nodes.

The complete BDM formula is as follows:

$$BDM = \frac{BR(CP/n1)}{BR(CP/n1) + (DPK/n2) + (DPR/n3)} \tag{3.1}$$

The BDM itself is not sufficient to evaluate our populated ontology, because we need to preserve the useful properties of the standard Precision and Recall scoring metric. Our APR metric (Augmented Precision and Recall) combines the traditional Precision and Recall with a cost-based component (namely the BDM). We thus combine the BDM scores for each instance in the corpus, to produce Augmented Precision, Recall and F-measure scores for the annotated corpus, calculated as follows:

$$AP = \frac{BDM}{n + Spurious} \; and \; AR = \frac{BDM}{n + Missing} \tag{3.2}$$

while F-measure is calculated from Augmented Precision and Recall as:

$$F - measure = \frac{AP * AR}{0.5 * (AP + AR)} \tag{3.3}$$

3.3 Comparison with other metrics

In order to validate the effectiveness of the BDM, we carried out some experiments to compare it with 2 other metrics, Learning Accuracy and the flat traditional measure. We experimentally compared the Hieron algorithm [LBC06] with the SVM learning algorithm [CST00, LBC05a] for OBIE. The SVM is a state of the art algorithm for classification. In the application of the SVM to OBIE, we learned one SVM classifier for each concept in the ontology separately and did not take into account the structure of the ontology. In other words, the SVM-based IE learning algorithm was a flat classification in which the structure of concepts in the ontology was ignored. In contrast, the Hieron algorithm for IE was based on hierarchical classification that exploits the structure of concepts.

For each experiment we computed three F_1 values to measure the overall performance of the learning algorithm. One was the conventional micro-averaged F_1 in which a binary reward was assigned to each prediction of instance — the reward was 1 if the prediction was correct, and 0 otherwise. We call this flat_F_1 since it does not consider the structure of concepts in the ontology. The other two measures were based on the BDM and LA values, respectively, which both take into account the structure of the ontology.

	flat_F_1	BDM_F_1	LA_F_1
SVM	73.5	74.5	74.5
Hieron	74.7	79.2	80.0

Table 3.1: Comparison of Hieron and SVM for OBIE

Table 3.1 presents the experimental results for comparing the two learning algorithms SVM and Hieron. We used three measures: conventional micro-averaged flat_F_1 (%), and the two ontology-sensitive augmented F_1 (%) based respectively on the BDM and LA, BDM_F_1 and LA_F_1. In this experiment, the International-Politics part of the OntoNews corpus was used as the test set, and the other two parts as the training set.

Both the BDM_F_1 and LA_F_1 are higher than the flat_F_1 for the two algorithms, reflecting the fact that the latter only counts the correct classifications, while the former two not only count the correct classifications but also the incorrect ones. However, the difference for the Hieron is more significant than that for the SVM, demonstrating an important difference between the two methods — the SVM based method just tried to learn a classifier for one concept as well as possible, while the Hieron based method not only learned a good classifier for each individual concept but also took into account the relations between the concepts in the ontology during the learning.

In terms of the conventional flat_F_1, the Hieron was slightly better than the SVM. However, if the results are measured by using the ontology-sensitive measure BDM_F_1 or LA_F_1, we can see that the Hieron performed significantly better than the SVM. Clearly, the ontology-sensitive measures such as the BDM_F_1 and LA_F_1 are more suitable than the conventional flat_F_1 to measure the performance of an ontology-dependent learning algorithm such as Hieron.

In order to analyse the difference between the three measures, Table 3.2 presents some examples of entities predicted incorrectly by the Hieron based learning system, their key labels, and the similarity between the key label and predicted label measured respectively by the BDM and the LA. Note that in all cases, the flat measure produces a score of 0, since it is not an exact match.

No.	Entity	Predicted label	Key label	BDM	LA
1	Sochi	Location	City	0.724	1.000
2	Federal Bureau of Investigation	Organization	GovernmentOrganization	0.959	1.000
3	al-Jazeera	Organization	TVCompany	0.783	1.000
4	Islamic Jihad	Company	ReligiousOrganization	0.816	0.556
5	Brazil	Object	Country	0.587	1.000
6	Senate	Company	PoliticalEntity	0.826	0.556
7	Kelly Ripa	Man	Person	0.690	0.667

Table 3.2: Examples of entities misclassified by the Hieron based system

All the concepts and their relations involved in Table 3.2 are illustrated in Figure 3.1, which presents a part of the PROTON ontology. This ontology section starts with the root node *Thing*, and has 10 levels of concepts with *TVCompany* as the lowest level concept. Note that the graph does not show all the child concepts for most of the nodes presented.

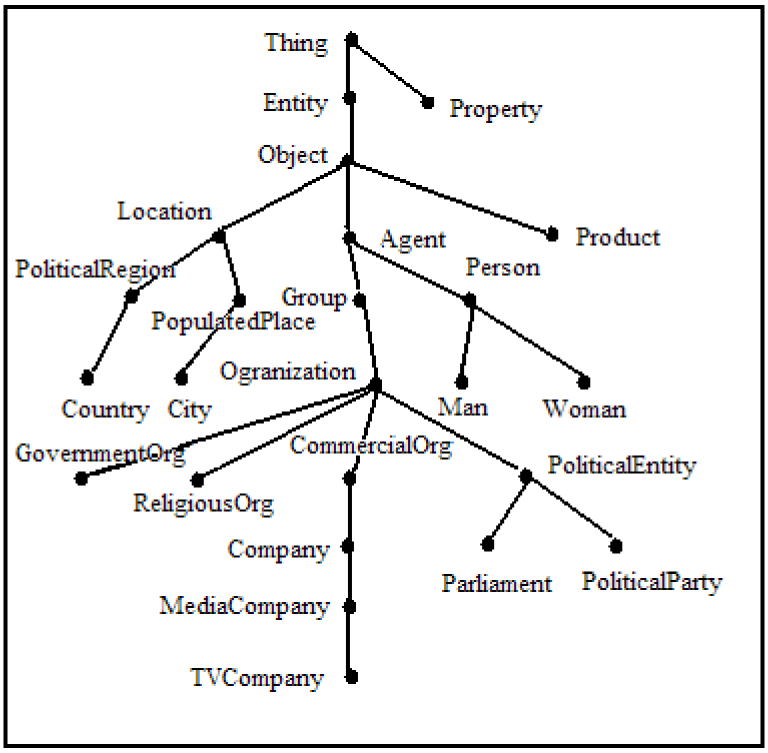


Figure 3.1: Subset of the PROTON ontology

The conventional flat measure assigned each case a zero similarity because the examples were misclassified and the measure does not consider the structure of labels. On the other hand, both the LA and BDM take into account the structure of labels and measure the degree of a misclassification based on its position in the ontology. Hence they assign a non-zero value to a misclassification in most cases. Note that zero would be assigned in the case where the MSCA is the root node. In our experiments, all the concepts used were below the node "Entity" and so we used its immediate upper node "Thing" as root[1]. This meant that CP (the depth of the MSCA) was always at least 1, and hence there is no zero value for BDM or LA in our experiments. This is because we consider that if an entity's instance is recognised but with the wrong type, the system should have a non-zero reward because it at least recognised the instance in the first place. However, this could be changed according to the user's preference.

However, BDM and LA adopt different mechanisms in consideration of the ontology structure. In particular, the LA assigns the maximal value 1 if the predicted label is an ancestor concept of the key label, regardless of how far apart the two labels are within the ontological chain. In contrast, the BDM takes into account the similarity of two concepts in the ontology and assigns a distance-dependent value. The difference is demonstrated

[1]"Thing" subsumes both "Entity" and "Property"

by the examples in the table. For example, in the Proton ontology, the predicted label *Organization* is the parent concept of the key label *GovernmentOrganization* in the second example, and in the third example the same predicted label *Organization* is 4 concepts away from the key label *TVCompany*. Hence, the BDM value of the second example is higher than the BDM value of the third example. In the first example, the predicted label *Location* is 3 concepts away from the key label *City* but its BDM value is lower than the corresponding value in the third example, mainly because the concept *Location* occupies a higher position in the Proton ontology than the concept *Organization*. Similarity is thus lower because higher concepts are semantically more general, and therefore less informative.

Another difference between the BDM and LA is that the BDM considers the concept densities around the key concept and the response concept, but the LA does not. The difference can be shown by comparing the fourth and the sixth examples. They have the same predicted label *Company*, and their key labels *ReligiousOrganization* and *PoliticalEntity* are two sub-concepts of *Organization*. Therefore, the positions of the predicted and key labels in the two examples are very similar and hence their LA values are the same. However, their BDM values are different — the BDM value of the fourth example is a bit lower than the BDM value of the sixth example. This is because the concept *PoliticalEntity* in the sixth example has two child nodes but the concept *ReligiousOrganization* in the fourth example has no child node, resulting in different averaged lengths of chains coming through the two concepts.

The BDM value in the fifth example is the lowest among the examples, mainly because the concept *Object* is in the highest position in the ontology among the examples. These differences in BDM scores show the effects of the adoption of chain density and branching factor as penalty weights in the computation of the score. These reflect the level of difficulty associated with the selection of a particular ontological class relative to the size of the set of candidates.

3.4 Scalability of evaluation metrics

In addition to examining the scalability of the annotation tools, we also take a look at the scalability of the evaluation metrics described in Section 3.2. Specifically, we would like to know how the new metric we have proposed (the BDM) measures up to other metrics when the ontology is collapsed or expanded in various ways, and what happens with smaller or larger ontologies. We therefore performed some experiments to measure this.

For the experiment, we needed an initial ontology in various incarnations, a corpus, a set of gold standard annotations, and a semantic annotation system. We started with the Proton ontology and created two shallower versions of it, converting also the gold standard annotations to match these ontology versions. We then performed annotation experiments with these new versions, and compared the result with the original ontology version. We also ran the annotation system with 2 different learning algorithms: SVN and Hieron, and compared 3 different metrics: BDM, LA, a standard distance metric, and a flat (traditional) metric.

3.4.1 Creating the ontology versions

The ontology used in the generation of the ontological annotation process for our experiments was the PROTON ontology[2], which has been created and used in the scope of the KIM platform[3] for semantic annotation, indexing, and retrieval [KPO$^+$04]. The PROTON ontology forms part of an annotation tool for automatic ontology population and open-domain dynamic semantic annotation of unstructured and semi-structured content for Semantic Web knowledge management applications. The ontology consists of around 250 classes and 100 relations. PROTON has a number of important properties, e.g. it is domain-independent, and therefore suitable for the news domain, and it is modular (comprising both a top ontology and a more specific ontology).

We deduced two shallower versions of the Proton ontology from the upper module of the Proton Ontology (Protonu): PTop and Link-1 (a small section of which is shown respectively in Figures 3.2 and 3.3). Note that these figures show the same section of Proton as was depicted in Figure 3.1 showing the original Proton ontology.

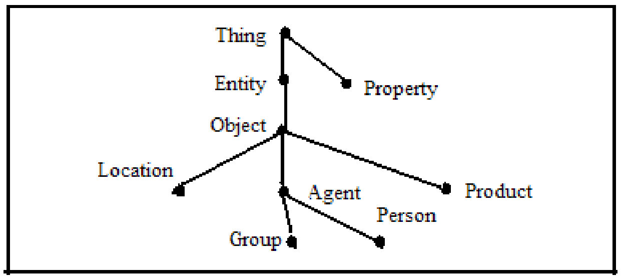


Figure 3.2: Subset of the PTop version of Proton

PTop was based on the concept levels of the ontology, and was created by just keeping the concepts with the "ptop" tag in the original Protonu ontology, i.e. the uppermost concepts. Other concepts in Protonu were mapped to the nearest ancestor concept, i.e. "ptop". This reduced the number of concepts from 272 to 25.

Link-1 was based on the link characteristics. For each node in the Protonu, if it was the only child concept of its parent, then the node was collapsed with its nearest ancestor

[2]http://proton.semanticweb.org
[3]http://www.ontotext.com/kim

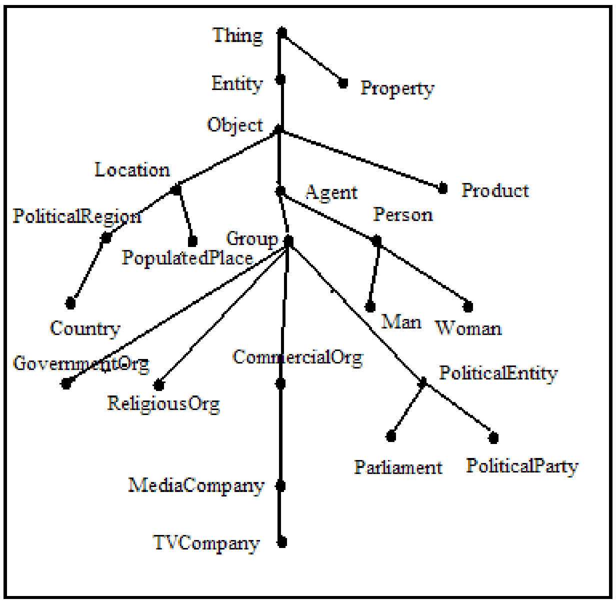


Figure 3.3: Subset of the Link1 version of Proton

concept with more than one child node. This reduced the ontology size from 272 to 244 concepts.

3.4.2 Experiments

We then performed the annotation experiments to compare the shallower versions with the original one. We used an SVM-based IE system and a Hieron-based IE system, respectively.

The SVM [CST00]) is a state of the art algorithm for classification. [LBC05a] applied the SVM with uneven margins, a variant of the SVM, to the traditional information extraction problem and achieved state of the art results on several benchmarking corpora. In the application of the SVM to annotation, we learned one SVM classifier for each concept in the ontology separately and did not take into account the structure of the ontology. In other words, the SVM-based IE learning algorithm was a flat classification in which the structure of concepts in the ontology was ignored.

In contrast, the Hieron algorithm for IE was based on hierarchical classification that exploits the structure of concepts. This algorithm learns a Perceptron classifier for each concept in the ontology; meanwhile it tries to keep the difference between two classifiers proportional to the cost of their corresponding concepts in the ontology. In other words, the learning algorithm tries to classify an instance as correctly as it can. If it cannot classify the instance correctly, it then tries to classify it with another concept with the least cost associated with it relative to the correct concept. The algorithm is based on the Hieron, a large margin algorithm for hierarchical classification proposed in [DKS04]. See [LBC06] for details about the learning algorithm and experiments.

For the experiments we used the OntoNews corpus [PABC05]. This semantically annotated corpus consists of 292 news articles from three news agencies: The Guardian, The Independent and The Financial Times, and cover the period of August to October, 2001. The articles belong to three general topics or domains of news gathering: international politics, UK politics and business. For each learning algorithm, two parts (business and UK politics) were used as training data and the third part (international politics) as test data. Note that although the tripartition of the corpus indicates three distinct and topically homogeneous parts of the corpus, these parts are used as training and testing data for the comparison of different algorithms, and not their performance. For this purpose, semantic homogeneity does not play a role.

Since the corpus used for the experiments has already been annotated with the Protonu Ontology, we therefore needed just to convert it to make versions compatible with the shallow versions of the Proton ontology, by mapping from the removed concepts to the concepts in the new versions.

3.4.3 Experimental measures

We compared 4 different evaluation metrics in our experiments.

The **flat** metric only considers as correct an exact match between the span and type of the response and that of the gold standard (key) entity. This is the standard Precision and Recall metric used in traditional IR and IE evaluations.

The **distance** metric not only considers as correct an exact match, but also allows for a very basic semantic distance between key and response concepts in a hierarchy. If an entity's type is misclassified as another type, then it gives the match a score between 0 and 1, which is the distance between the two concepts divided by the maximal distance between any two concepts in ontology.

The **BDM** metric uses the BDM score for each match (whether correctly classified or misclassified), as described in Section 3.2.

The **LA** metric uses the LA score for each match.

The most important differences between these metrics are as follows:

- The flat measure does not consider any kind of misclassification (whereas all the other measures do).
- The distance and BDM measures are normalised with the size of the ontology, but the LA is not.

	Flat	Distance	LA	BDM
Protonu	73.7	83.7	82.5	83.3
Link-1	73.8	83.4	82.2	83.0
Ptop	81.8	88.5	88.3	88.4

Table 3.3: Results for SVM-based annotation

	Flat	Distance	LA	BDM
Protonu	70.1	88.2	86.4	87.7
Link-1	69.5	88.5	86.6	88.0
Ptop	79.1	88.4	88.1	88.3

Table 3.4: Results for Hieron-based annotation

- The distance measure is normalised directly according to the ontology size, but the BDM is only normalised with respect to this in an indirect way.
- The BDM considers the local density of the ontology but the distance measure does not.
- The LA is the only non-symmetrical measure, i.e. it gives a different score if the Key and Response are inverted.

For each metric, the Augmented F-measure described in Section 3.2 is calculated.

3.4.4 Results and discussion

The results for the SVM-based annotation with the four measures are shown in Table 3.3, in terms of Augmented F-measure (%).

The results for the Hieron-based annotation with the four measures are shown in Table 3.4, in terms of Augmented F-measure (%).

As expected, the measures which take into consideration the misclassifications have higher values than the flat measure. This is particularly true for the Hieron system. These measures also tend to increase in score as the ontology becomes shallower. This is unsurprising, because there is therefore greater similarity between key and response values with a shallower ontology.

In the SVM experiment, there is little difference in score between the Protonu and Link-1 ontologies. The flat measure increases a fraction in value while the other measures all decrease slightly. There is a much greater difference (for all measures, but especially for the flat measure) between the Protonu and Ptop ontologies. Clearly, the flat measure increases because more entities are considered correct, the more the ontology is flattened.

The distance and BDM metrics only show slight variation between Protonu and Ptop because they take into account the depth of the ontology. The LA shows a greater variation because it does not take this into account, and therefore shows a higher value because key and response elements are closer in the ontology. Another interesting point is that the shallower the ontology, the more similar the 3 distance-based metrics are to each other.

In the Hieron experiment, we see a much greater distinction between the flat and hierarchical measures. Here, we also see that the LA shows a much greater variance between the different versions of the ontology than the BDM and Distance measures. It seems quite strange here that the Distance measure actually shows a lower value for the shallowest ontology (Ptop) than for the next shallowest (Link-1), though both are higher than for the full ontology (Protonu). However, the variance between all 3 scores for the Distance measure is quite small. On the other hand, the LA and BDM measures show more of a difference between the different ontology versions, both increasing in score as the ontology gets smaller. Again, the increase is greater for the LA than for the BDM.

We can conclude from these experiments that all three hierarchical measures are better than conventional measures for evaluating ontology-based annotation. The BDM is less sensitive to ontology size than LA, because it considers normalisation with respect to ontology size, which is important as the ontology gets bigger. BDM is also the only one which reflects ontology density; the others only reflect size. We propose to do some further experiments to highlight this. The results show us that with a very shallow ontology, it makes little difference which of these three measures is used. The larger (and deeper) the ontology, the more difference it makes which method we use. We would expect that as the ontology increases in size and complexity, the more important the choice of metric is. Of course, shallowness of the ontology is not the only factor related to its size: properties, labels, or other ontology components also play a part. However, since they are not used for the ontology-based information extraction component, we are not interested here in evaluating these factors separately.

3.5 Evaluating the performance of multimedia annotation tools

As discussed in Chapter 1, the common feature shared among all of the examined multimedia annotation tools, is the fact that they address manual annotation solely. Semi- and fully automatic multimedia annotation has been the holy grail in the content-based multimedia community for the past decade, leading to a plethora of initiatives and research efforts, including both machine learning and knowledge-assisted methodologies. However, bridging the so called *semantic gap* presents major challenges still, especially when domain and application independence is considered. As a result, most existing multimedia annotation tools provide manual annotation facilities only.

This renders problematic both the definition of metrics and the evaluation of such tools. As described in the tools presentation in Chapter 1, the user is responsible for manually selecting the region/temporal interval they want to annotate, and the corresponding concept from the domain ontology. Consequently, the correctness of the annotation cannot be subject to question. The same applies in the case of M-Ontomat-Annotizer, where low-level feature knowledge is addressed. The extraction is implemented following the MPEG-7 XM specifications, i.e., it complies to the standard, and is decoupled by the actual annotation functionalities provided. A dimension that could be of potential interest in manual annotation could be the study of subjectivity introduced by the different annotators. However, such evaluation does not really reflect the provided functionalities, but rather examines epistemic and perceptional aspects of the annotators.

There are, however, a couple of dimensions along which tools for manual multimedia annotation could be characterised. These extend the features discussed in Chapter 1 and include among others:

- support for automatic segmentation that would remove some of the annotator burden;
- built-in ontology for representing media aspects, or, alternatively, support to load one (in addition to the domain specific ontologies);
- automatic extraction of spatial relations that would enable the annotator to speed up annotations where spatial information is needed;
- recommendation services based on audivisual similarity that would automate annotation to a considerable degree, etc.

Although such functionalities are not currently supported, they are part of the planned extensions to most tools.

Chapter 4

Metrics for Evaluating Automatic Document Indexing

4.1 Introduction

Precision and Recall are widely used metrics for evaluations in information retrieval. With automatic document indexing, they can be used to describe the quality of the automatically assigned concepts with respect to a gold standard of manually selected keywords.

In this chapter, we present experiments on assessing the quality of automatic document indexing. We show the weakness of the classical definition of Precision and Recall and extend the definition to a generalized Precision and Recall (also referred to as Augmented Precision and Recall).

The focus lies on the impact of different similarity measures on the evaluation results.

In particular, we take the following steps:

- We automatically create annotations for different document sets and thesauri by means of the Collexis Engine described in Section 4.1.3.
- Next, we compare them with existing manual annotations using different similarity measures.
- We show the differences between these measures and how these differences influence the result of the precision and recall.

4.1.1 Thesauri

In the experiments, we use two thesauri from different domains and with quite different characteristics, in order to be able to generalize our observations.

The first is MeSH[1], a well established thesaurus from the medical domain, which is extensively used to annotate large collections of medical documents.

The second is the German standard thesaurus for business and economics, STW[2], which has been created recently to provide the basis for indexing literature in the area of business and economics. In contrast to MeSH, the use of this thesaurus is limited to a number of specialized libraries, and there is no experience with its use for automatic indexing.

In our experiments, we used the MeSH 2006 thesaurus in English, with 31956 concepts and about 170.000 terms, and the STW in German, which consists of 6294 concepts and 27204 terms. While the hierarchy of the subparts is only available in German, every concept contains an English term as a synonym.

In our first experiments, we used an implementation of the STW thesaurus that was not built by ourselves. This implementation is broken, as it lacks the complete hierarchy on top of the different subthesauri. Instead, the subthesauri simply are put directly under an artificial root concept. Only the broader terms within one subthesaurus are used to build the hierarchy. The result is a very flat hierarchy with a huge number of single concepts directly located under the root node.

We came across the broken implementation when we achieved astounding good results in our evaluation with different similarity measures. So we use this implementation to demonstrate some weaknesses and pitfalls of these measures.

4.1.2 Document sets

For this evaluation, we use two different textual corpora. The first is a randomly selected collection of 706 Medline abstracts[3]. These articles were annotated with the MeSH thesaurus. The Medline database contains keywords from the MeSH thesaurus selected by human experts.

The second corpus is a document base of 391 economic abstracts, provided by Elsevier B.V.[4], indexed with the German-English Standard Thesaurus Wirtschaft (STW). We acquired manually selected STW keywords from the Econis Database of the German central economic library (Deutsche Zentralbibliothek für Wirtschaftswissenschaften)[5]. The abstracts are from three different journals:

- Journal of Health Economics (JHE)
- Journal of Accounting and Economics (JAE)

[1]http://www.nlm.nih.gov/mesh/
[2]http://www.gbi.de/thesaurus/
[3]http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?DB=pubmed
[4]http://www.elsevier.com
[5]http://zbwopc4.zbw.ifw-kiel.de:8080/DB=1/LNG=DU/

Concept	Keyword
Price	Pricing behaviour of firms
Cigarette	Cigarette industry
Oligopoly	Oligopoly
Tax increase	Effects of taxation
State tax	Tobacco tax
Regulation	
Retail price	
State	
Panel	

Table 4.1: Concepts found by the indexer and by manual selection

- Journal of Financial Economic (FINEC)

The following randomly selected article is used as an example:

Title Do cigarette producers price-discriminate by state? An empirical analysis of local cigarette pricing and taxation.

Authors Theodore E. Keeler, Teh-wei Hu, Paul G. Barnett, Willard G. Manning, Hai-Yen Sung

Abstract This study analyzes the interactive effects of oligopoly pricing, state taxation, and anti-smoking regulations on retail cigarette prices by state, using panel data for the 50 US states between 1960 and 1990. The results indicate that cigarette producers do price-discriminate by state, though the effect is not large relative to the final retail price. There are two further results: (1) state taxes are more than passed on - a 1-cent state tax increase results in a price increase of 1.11 cents, and (2) sellers offset state and local anti-smoking laws with lower prices, thereby blunting effects of the regulations.

Journal Journal of Health Economics

Table 4.1 shows the concepts found by the indexer and the manually selected keywords (gold standard).

4.1.3 Collexis engine

The Collexis Engine is a thesaurus-based search engine built by Collexis B.V., Geldermalsen, The Netherlands. As such, it is not meant to be a stand-alone textual annotation

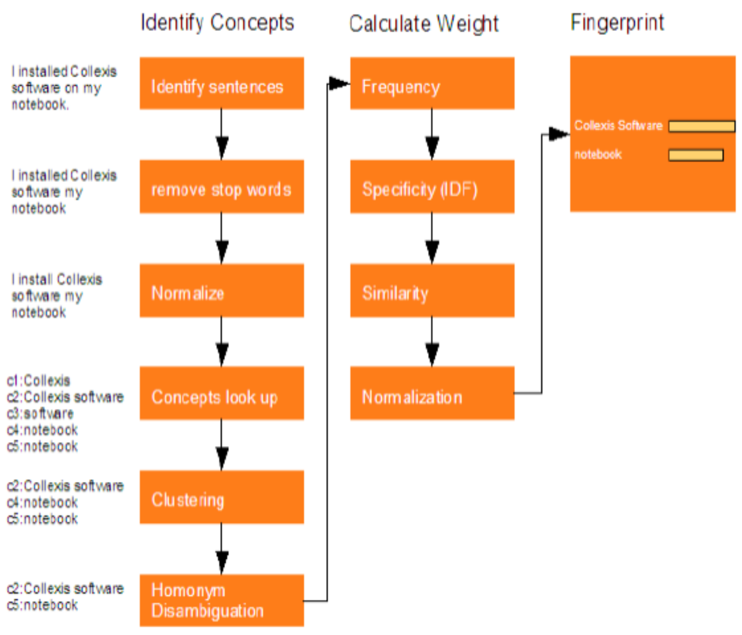


Figure 4.1: Collexis Workflow *(Source: Collexis B.V.)*

tool, but it performs a semantic analysis of given text documents and annotates the texts with concepts from a thesaurus to support a later document retrieval. The Collexis Engine is used in Chapter 4 to generate annotations for documents. These annotations are then compared to manually selected keywords by means of different semantic similarity measures.

Generating Fingerprints Collexis uses the vector space model for document retrieval. The vectors representing the records and queries are called fingerprints, the generation of them is called fingerprinting. Such a fingerprint vector contains relevance values for concepts taken from a thesaurus and found in a given text. The Collexis engine is used to create fingerprints of text based information like documents, papers, sheets or web pages.

Figure 4.1 shows the workflow to generate a fingerprint, including normalization, clustering and disambiguation of the words in the text. Every concept in the fingerprint is ranked by its relevance for the given text (Figure 4.2).

The process of fingerprint creation can involve multiple thesauri and languages. This makes it possible to handle documents from different languages and find relevant documents, even if they are written in a language not used for the query.

Document Retrieval Once the fingerprints for all documents in the document base are generated and saved in a so called “Collexion”, one can use the Collexis Engine to search for relevant documents with respect to a given query. In this case, a fingerprint of the

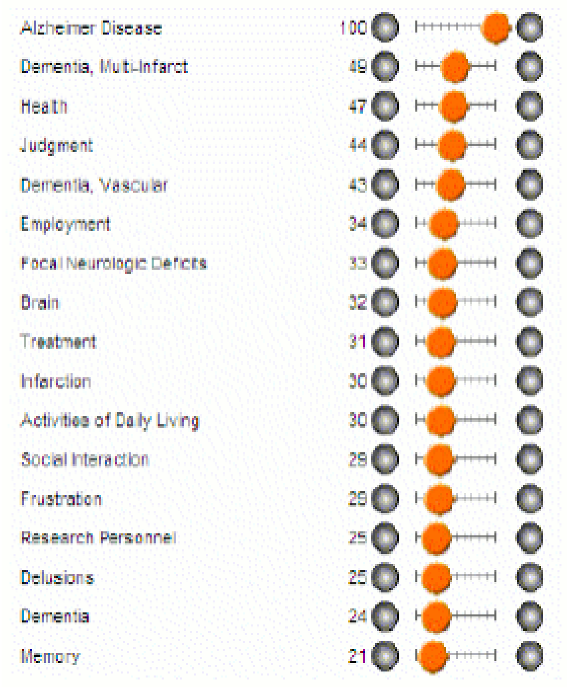


Figure 4.2: Collexis Fingerprint *(Source: Collexis B.V.)*

query string is generated and a distance measure in vector space is used to find the nearest documents and return them as result.

Architecture The Collexis architecture is developed as a 3-tier environment, as illustrated by Figure 4.3. The core components are the thesaurus component and the matching component. The thesaurus component is used to generate the fingerprints of documents and queries, the matching component executes the document retrieval. The underlying data (thesauri and collexions with fingerprints) is stored in a proprietary database and can only be accessed via the Collexis Engine.

The clients and tools communicate with the Collexis Engine via the TCP/IP Collexis Gateway, which provides the full API for the Engine. There are different implementations of this API for different languages, like Java, .NET and Python.

4.2 Generalised Precision and Recall

Table 4.2 shows the average Precision and Recall for the two document bases used. STW (broken) refers to the broken implementation, as described in Section 4.1.1. With our Medline reference set, only 25% of the gold standard keywords are found by the indexer. The result of the Elsevier base is even worse at only 18%.

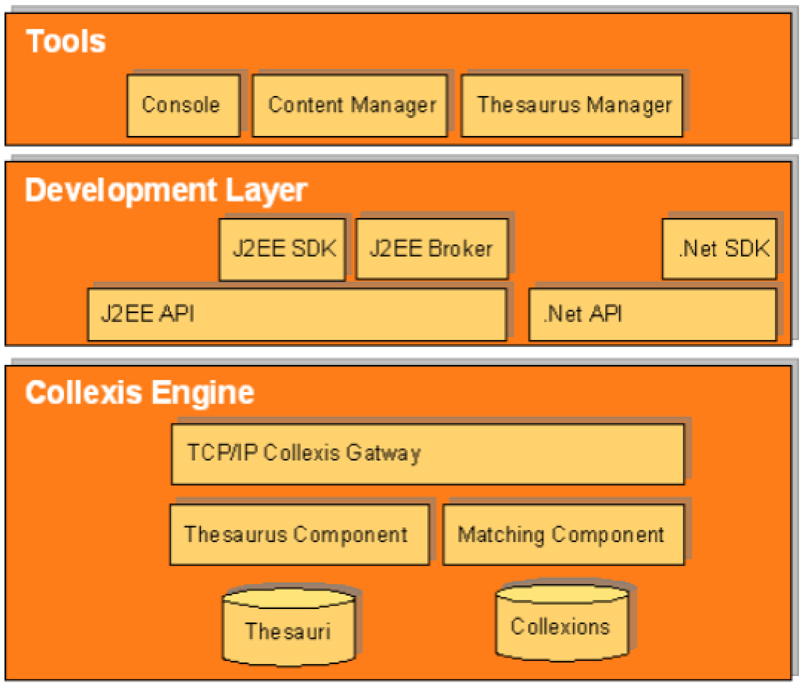
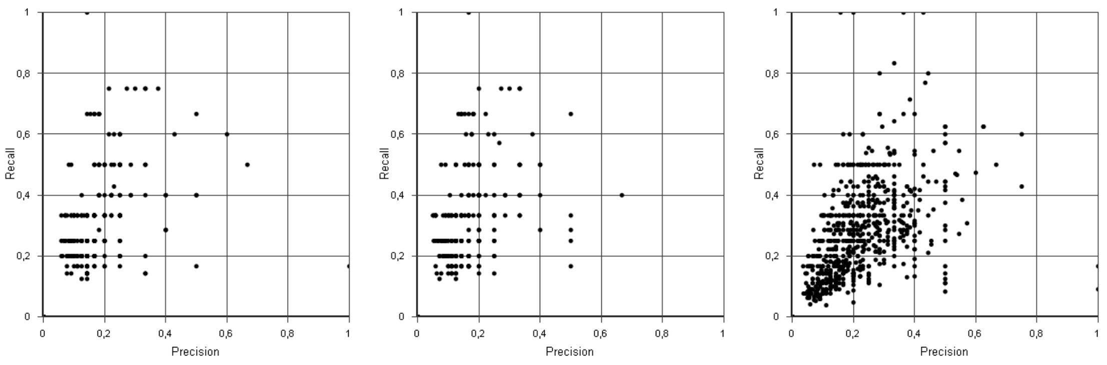


Figure 4.3: Collexis Architecture *(Source: Collexis B.V.)*

Document Base	# Documents	# Keywords	# Correct	Recall
STW (broken)/Elsevier	391	1658	293	0.18
STW/Elsevier	391	1646	309	0.19
MeSH/Medline	706	8143	2030	0.25
		# Concepts	# Correct	Precision
STW (broken)/Elsevier	391	2980	293	0.1
STW/Elsevier	391	3377	309	0.09
MeSH/Medline	706	10041	2030	0.20

Table 4.2: Binary Precision and Recall results



(a) STW/Elsevier Binary (b) STW (broken)/Elsevier Binary (c) MeSH/Medline Binary

Figure 4.4: Binary Precision and Recall results

Looking at the graph showing the Precision and Recall of all documents (Figure 4.4), we see that the results are mostly located in the bottom left quarter, thus indicating a rather bad result. Only a few samples have either a good Recall or a good Precision, and no samples show good results for both of them. The Precision is generally lower than the Recall, as there are more found concepts than human selected keywords.

An examination of the indexing results shows that the binary approach judging a found concept as correct or incorrect with respect to the given keywords is not appropriate: a human can use more abstract keywords not used in the text. For example, in some contexts the keyword *Asia* might get assigned by a human, whereas *China* and *Japan* are used in the text. In the abstract, the common concepts *Price* and *Retail price* are found, whereas the human selected keywords contain the more specific *Pricing behaviour of firms.*

A generalised approach To reflect these circumstances, we need a metric to decide just how right or wrong a found concept is, with respect to the given keywords.

Different researchers have proposed alternative measures, often referred to as generalised precision and recall that do not only take the overlap between concepts into account but also consider the semantic distance between between concepts that are not in the common term set [HS98b, MPL06b]. The most recent proposal for generalising precision and recall has been made by Euzenat [Euz07], who defines generalised precision and recall in the following way:

$$Prec_\omega(A,R) = \frac{\omega(A,R)}{|A|} \qquad Rec_\omega(A,R) = \frac{\omega(A,R)}{|R|}. \tag{4.1}$$

Here A is the automatically created annotation, R is a reference annotation, in our case the manually created one and ω is a function that measures the overlap between A and R.

There are many options for choosing ω. In the context of comparing annotations from a thesaurus it makes sense to base the definition on notions of semantic similarity between concepts. There are a number of proposals for semantic similarity measures including purely structural measures, as well as measures that are based on information theoretical concepts.

This leads us to the following definition of precision and recall with respect to a single document:

$$Recall = \frac{\sum_{r \in R_d} \mathtt{max}_{a \in A_d}\ Similarity_L(r,a)}{|A_d|} \tag{4.2}$$

$$Precision = \frac{\sum_{a \in A_d} \mathtt{max}_{r \in R_d}\ Similarity_L(a,r)}{|R_d|}. \tag{4.3}$$

where A_d refers to the set of automatically created annotations for document d, R_d and to the set of manually assigned keywords of document d. Note that we can calculate the overall precision and recall for a complete document set by summing over all documents d.

This generalisation is compatible with the Augmented Precision and Recall presented by Maynard et al. [May05b] [MPL06b] and described in Section 3.2. To summarise briefly, they use a balanced distance metric BDM_i, which corresponds to our $weight(c,k)$ function. The sum of all distances in a given set $BDM = \sum_{i=1..n} BDM_i$ matches the overlap function $\omega(A,R)$ of Euzenat and is used in our implementation as well.

4.3 Semantic similarity measures

We tested with several widely known and well examined similarity measures to determine the degree of correctness for a given concept and its nearest matching counterpart in the reference set.

Measuring the semantic similarity between words or documents is an important task for information retrieval and natural language processing. The idea behind semantic similarity is to define a metric, that says how similar two words or documents are.

One general approach for calculating semantic similarity is to use a thesaurus and then find some distance measure in this thesaurus. The structure of the thesaurus and the relationship used to build the thesaurus provide the focus of the resulting similarity. Similarity measures using this approach are referred to as network-based, thesaurus-based or ontology-based semantic similarity measures.

4.3.1 Thesaurus based measures

Thesaurus-based measures generally use a distance measure within the thesaurus tree to determine the degree of similarity. The simplest approach would be using the distance between two nodes. The distance is defined by the number of nodes or edges on the shortest path between two nodes. This is called node-counting or edge-counting.

Leacock and Chodorow. Leacock and Chodorow use node-counting for their similarity measure presented in [LC98]:

$$Similarity_{LC}(c_1, c_2) = -\log \frac{distance}{2 \cdot maxdepth} \tag{4.4}$$

where $distance$ is the node-counting distance between the two concepts and $maxdepth$ is the maximum depth of the thesaurus. As $2 \cdot maxdepth$ is the longest possible distance, the values of this measure range from 0 to $\log(2 \cdot maxdepth)$.

We normalised this measure by dividing by the maximum possible value:

$$Similarity_{LCNORM}(c_1, c_2) = \frac{Similarity_{LC}(c_1, c_2)}{\log(2 \cdot maxdepth)} \tag{4.5}$$

After normalisation, this measure has the value 1, if and only if both input concepts are the same. This holds for the following measures as well.

The best results are achieved using the broken STW implementation (see Section 4.1.1). This implementation lacks the additional top-level hierarchy of the single sub-thesauri and is thus a rather flat hierarchy. As the Leacock Chodorow measure does not punish the involvement of the root node as least common subsumer, all the concepts near the root of a subthesaurus are considered very similar.

Wu and Palmer A normalised measure with values between 0 and 1 was presented by Wu and Palmer [WP94]:

$$Similarity_{WP}(c_1, c_2) = \frac{2 * depth(LCS)}{depth(c_1) + depth(c_2)} \tag{4.6}$$

They used the least common subsumer (LCS) of two concepts, which is the most specific concept (with the highest depth) in the tree that has both input concepts as child nodes. As the root node has a depth of 0 in our implementation, this measure always assigns 0 to concepts from different subtrees of the root node. So the results indicate clearly the weakness of the broken STW implementation.

4.3.2 Information based measures

One problem with the measures described above is that they assume that all the nodes in the thesaurus have the same distance according to similarity. In other words, with these measures, two concepts have the same degree of similarity if they have the same distance in the thesaurus. In real-world thesauri, there is a wide variability in the semantic similarity between adjacent nodes, especially if the thesaurus is combined from different source thesauri. Generally there are areas with a high density of differentiating concepts and other areas where only some common concepts exist.

To overcome this problem, Resnik introduced a new way to measure the semantic similarity of words, based on the notion of information content [Res95]:

$$IC(c) = -\log P(c) \tag{4.7}$$

The information content of a given concept is derived from its probability to encounter an instance of this concept or one of its child concepts in a document base. So, the probability increases monotonically as one moves up in the thesaurus hierarchy. If c_2 is a child concept of c_1 then $P(c_2) < P(c_1)$. With a single root thesaurus we have $P(root) = 1$.

In line with information theory, a concept has a higher information content if its probability of being encountered is lower. So the information content of the root concept is 0.

Resnik The similarity measure of Resnik reads as

$$Similarity_R(c_1, c_2) = \max_{c \in S(c_1,c_2)} (IC(c)) \tag{4.8}$$

with $S(c_1, c_2)$ denoting the set of subsumers from the LCS to the root concept.

As the information content decreases with the level of the concept in the thesaurus hierarchy, we can use the least common subsumer:

$$Similarity_R(c_1, c_2) = IC(LCS) \tag{4.9}$$

Note, that this simplification only holds for monohierarchical thesauri. In a polyhierarchical thesaurus, two concepts can have more than one LCS with different information content. In this case, again the maximum information content has to be chosen.

Normalised Information Content A drawback of Resnik's measure is that the similarity values range from 0 (the root concept is LCS) to $\lim_{P(c)\to 0} IC(c) = \infty$. For our experiment, we normalize 4.7 to

$$IC_{norm}(c) = \begin{cases} \frac{-\log P(c)}{-\log(\frac{1}{\alpha N})} = -\frac{\log P(c)}{\log \alpha N} & freq*(c) > 0 \\ 1 & freq*(c) = 0 \end{cases} \tag{4.10}$$

with α as a weighting parameter for the special case of $freq*(c) = 0$. With $\alpha = 1$, a concept with a $freq*(c)$ of 1 has also an Information Content of 1. With increasing α, the Information Content of these concepts decreases.

In our experiments, we see that as the information content of the least common subsumer is used as similarity value, most and even identical concepts have an assigned similarity value far below 1. As the root concept has an information content of 0, at least the flat structure of the broken STW implementation is punished.

Lin Lin introduced his measure [Lin98] to build on Resnik's. It leads to a normalised value between 0 and 1:

$$Similarity_L(c_1, c_2) = \frac{2 \cdot IC(LCS)}{IC(c_1) + IC(c_2)} \tag{4.11}$$

Our results with this measure also show the punishment of the broken STW due to the Information Content of 0 of the root node.

Jiang and Conrath A very similar approach is used by Jiang and Conrath [JC97]. They introduce not a similarity, but a distance measure. Instead of counting the nodes between two concepts, they sum the link strengths between these nodes. This link strength is defined as the difference of the Information Content of a node and its parent node:

$$Distance_{JC}(c_1, c_2) = IC(c_1) + IC(c_2) - 2 \cdot IC(LCS) \tag{4.12}$$

This distance measure can also be used as similarity measure [CM05]:

$$Similarity_{JC}(c_1, c_2) = \frac{1}{IC(c_1) + IC(c_2) - 2 \cdot IC(LCS)} \tag{4.13}$$

A linear transformed and normalised version of this measure can be found at [SVH04]:

$$Similarity_{JC}(c_1, c_2) = 1 - \frac{IC(c_1) + IC(c_2) - 2 \cdot IC(LCS)}{2} \tag{4.14}$$

With this measure, the root node as LCS does not lead to a 0 result, but the influence is significant. Generally, this measure produces rather high values of Precision and Recall compared with Lin.

4.3.3 Intrinsic Information Content

Seco et al. [SVH04] presented an approach to determine the information content of a given concept without statistics from an underlying document base. Instead, they only used the thesaurus structure to define a measure for the information content. This so-called Intrinsic Information Content is defined as

$$IIC(c) = -\log\left(\frac{hypo(c)+1}{max}\right) \tag{4.15}$$

and can be normalized to

$$IIC_{norm}(c) = \frac{\log\left(\frac{hypo(c)+1}{max}\right)}{\log\left(\frac{1}{max}\right)} = 1 - \frac{\log(hypo(c)+1)}{\log(max)} \tag{4.16}$$

with $hypo(c)$ as the number of hyponyms (i.e. child nodes) of a given concept c and max as the number of concepts in the whole thesaurus.

The Intrinsic Information Content can be used as a replacement for Information Content in the above mentioned measures.

The authors used Wordnet[6] with very good results, and conclude that further experiments have to be done to see, if the intrinsic metric generalizes to other hierarchical knowledge bases.

In our experiments with the MeSH and STW Thesauri, the results with IIC were indeed comparable to the results with true Information Content. The Intrinsic Jiang Conrath values for the broken STW implementation are worse as with true IC Jiang Conrath. There are a lot of concepts in the broken STW with few or no child nodes. As this leads to a high value for the Intrinsic Information Content, the similarity values are lower than with true Information Content. In this regard, the measures with Intrinsic Information Content are more sensitive to an improper thesaurus structure.

We compared the results of Information Content with Intrinsic Information Content for the MeSH Thesaurus. We found that there is a high correlation in the two graphs and we can thus confirm the evaluation of Seco et al.

4.3.4 Conclusion

We evaluated various different similarity measures and their impact on the results of our Generalized Precision and Recall for the indexing process. The results depend strongly on the similarity measure used, so the preferred measure should be chosen carefully. A measure that does not punish the involvement of the root concept as LCS is not appropriate for our purpose. A similarity value of 0 for all concepts with the root node as LCS

[6]http://wordnet.princeton.edu/

Algorithm	γ
Leacock Chodorow	0.82
Wu and Palmer	0.74
Resnik	0.77
Lin	0.80
Jiang and Conrath[a]	-0.81
Resnik*[b]	0.77
Lin*	0.81
Jiang and Conrath*	0.84

[a]distance measure
[b]the * denotes the use of the intrinsic information content (Section 4.3.3)

Table 4.3: Correlation between human and machine similarity judgements

corresponds well with the traditional Precision and Recall, because these concepts can be seen as missing, respectively spurious.

In our opinion, a similarity measure is most convenient, if equality is denoted with 1, i.e. the values range between 0 and 1 with the value 1, if and only if a concept is compared to itself. The binary judgement of correct and incorrect matchings fits seamlessly to this kind of measure.

For IC based measures, we recommend the use of intrinsic information content on small document sets, but with large sets, the original measure with information content should be used to reduce the dependency on the quality of the thesaurus structure.

Table 4.3 shows the correlation of the different approaches to human judgement. These correlation values are taken from [SVH04]. It has to be noted that the correlation between several humans judging the same contents by far is nowhere near 100%: for example, Resnik performed a study with human subjects and found a correlation of 0.88 [Res99]. So this can be seen as an upper bound of what can be reached with a calculated similarity measure.

There are many more approaches for measuring semantic similarity. A complete overview goes beyond the scope of this work, but for example there is a measure using neural networks, proposed by Li, Bandar and Mclean [LBM02]. An extension to the similarity of concept sets or full texts can be found in [BKKB05] and [CM05]. Finally, Bernstein et al. suggest that the choice of a similarity measure depends on the underlying thesaurus, proposing a personalized measure adapted to the thesaurus [BKBK05].

Chapter 5

Performance of Annotation Tools

5.1 Introduction

This chapter investigates the performance of the various annotation tools discussed previously. Ideally, we would be able to run a single experiment comparing all the tools on a single dataset using the same ontology. There are several reasons, however, why this was not possible. First, as we have discussed earlier, the tools are not very interoperable and do not necessarily make use of the same ontology or annotation format. This makes it difficult, for example, to use an OWL ontology on a tool that is designed only to work with an ontology in its own proprietary format. Secondly, the tools are all designed for slightly different purposes, and so in some cases a direct comparison on the same data and task would not make sense and could provide misleading results. Thirdly, some of the tools had technical problems which made it impossible to carry out a new evaluation on them, so in such cases we can only report previous evaluation results.

In this chapter we describe and discuss the evaluation of each tool, followed by some details of previous experiments involving some or all of these tools. Where tthe experiments were not performed directly in the context of KnowledgeWeb, we sstate this explicitly. We stress that it is important to take account of performance results, but to consider them in perspective. For example, some tools are designed more to be used in combination with manual annotation than as a replacement for it, so optimal precision or recall may not be as important as other factors such as the usability issues described in Chapter 2 or the scalability issues described in Chapter 6.

5.2 MnM

One of the main problems with using MnM for semantic annotation is that in its current implementation (integrated with Amilcare), it requires the user to create a separate training corpus for each ontology class and to train each class individually. Clearly with anything other than a very small toy ontology, this is not really feasible as it is too time-consuming. There are also several other problems related to scalability issues which impede the evaluation of MnM in this way: see Section 6.5 for more details. For this reason, we were unable to conduct any experiments on MnM. We can report, however, about some previous small-scale experiments that have been carried out with it in the context of the Dot.Kom and other projects (see for example [SP05]). More details of these experiments can be found in Section 5.8.

5.3 Magpie

As has been mentioned earlier, Magpie in its annotation capacity relies on an input from existing ontologies, thesauri and other vocabularies. The content of these is then matched against the web pages in question – in the case of described experiment against the content of the experimental corpus. Thus, for the purpose of creating a Magpie lexicon for experimenting with, several existing ontologies and data stores have been used in the following manner:

- TAP and OpenCyc ontologies: large, general-purpose ontologies from which we reused such entities as product categories (e.g. beverage, shaving cream, etc.)
- Countries and geographic entities: large KB comprising countries and significant cities complemented with prior work done on screen-scraping instances for this ontology (using a generic Google corpus rather than a specialised one, as in this experiment)
- Political and News entities: largely focusing on named entities appearing commonly in the news stories, screen-scraped from a range of news sites (BBC, CNN)
- KIMO: general purpose knowledge base built by the KIM project – this has been used as a validation for the previous category, not as a source of entities per-se
- Agrovoc: a large thesaurus of terms related to the domain of agriculture – this has been used to partially supply knowledge for the category of (agriculture-related) products and environment entities

As in the case of other tools, Magpie can be evaluated with several different methods. For the purposes of this annotation, we decided to restrict the content classification within the Magpie lexicon to that acquired from the existing ontologies and knowledge bases. In other words, we tried to avoid manual re-classification or correction of items; the only modification that was necessary to do was related to Magpie's feature of 'top-level categories'. These were created by referring to upper-level classes within several ontologies, and manually deciding on one label. Hence, the lexicon for Magpie is structured and includes the following categories: (i) Products, (ii) Political entities, (iii) Geographic entities, and (iv) Environmental, agricultural, etc. entities.

This removal of specific classes enables us to compare Magpie's performance with that of other tools. In addition, it also enables us to re-use non-ontological entities such as thesauri, which would not normally have any association with concepts/classes. We therefore distinguished between applying differently focused lexicons to the three corpora documents. The reason for comparing differently focused lexicons rather than simply applying the mashed-up lexicon is to see the effect of semantic proximity between lexicon and corpus domain – otherwise, in a large lexicon, the behaviours on three corpora would be obscured. In other words, when e.g. the political entities part fails on a business corpus, the product and environment parts may artificially push the performance measures up. However, in general the average performance would not be much different, which is why we considered the use of the large, mashed-up lexicon to be slightly obscuring.

Corpus	Pol	Geo	Env	Prod
Business	94.7	100.0	88.9	91.1
Pol-Int	98.2	100.0	83.1	90.7
Pol-UK	100.0	100.0	83.0	92.0
Total	97.6	100.0	85.0	91.2

Table 5.1: Precision Evaluation for Magpie on Four Lexicons with Different Focus

Corpus	Pol	Geo	Env	Prod
Business	24.7	33.2	28.9	21.7
Pol-Int	28.2	35.0	33.1	45.7
Pol-UK	22.3	28.4	28.0	53.0
Total	25.1	32.2	30.0	40.1

Table 5.2: Recall Evaluation for Magpie on Four Lexicons with Different Focus

As can be seen in the result summary in the above two tables, the outcomes with respect to precision are very positive – indeed, if an item has been recognised by Magpie, it has been usually marked correctly. One reason why certain product- and environment-related terms were recognised with lower precision is partly due to overlaps between the two categories. For example, terms like 'crops' have appeared as generic, environment-related terms, but also in the context of products (such as 'GM crops' – a product of company Monsanto). Generally, this set of results is well above most other annotation tools – unfortunately, this only holds for the precision measurement.

The values for recall as a criterion are much more varied. The first observation is that the recall (i.e. the capability to identify the terms at all) ranges from mid twenties (in %) for political entities to mid thirties (in %) for geography. This is due to the nature of constructing the Magpie lexicons and also due to the nature of Magpie's term matching engine – while all occurrences of items directly present in the ontologies (e.g. 'Conservative Party' or 'Tony Blair') are spotted perfectly, the variations of these are completely ignored (typical examples include 'Mr Blair' vs. 'Tony Blair' or 'Foreign Secretary Straw' vs. known 'Jack Straw'). Similarly, in the geographic domain, the names of more important and larger entities (e.g. countries) were readily recognised, but not so the cities or regions (e.g. while 'Kabul' or 'Afghanistan' were identified, 'Robben Island' or 'Long Kesh' were not recalled at all).

In case of the business corpus, a vast number of named individuals, but also a substantial number of companies, were overlooked and not recalled – again, this is to a great extent due to the nature and structure of the lexicons derived from ontologies. More frequently occurring items like 'Gerry Adams' have been included as political figures and entities, but, say, 'Alessandra Tripodi' as an airlines spokesperson has not been. This obviously reduces the recall capacity for the politically based lexicon.

Interestingly, the recall rate for products seems to be higher in the politically based corpora. We believe this may be partly due to (i) a limited number of products mentioned in such stories, and (ii) the bias of political stories toward certain common, high-level 'products' (e.g. drugs, explosives, aeroplanes, etc.) However, even in this case, more detailed entities such as 'fire-retardant foam' were not recalled – in line with the observations made above on the narrow scope of ontologies.

Thus, rather unsurprisingly, this experiment shows both the advantage of the ontology-centric annotation in terms of good guaranteed precision, as well as its major shortcoming in terms of being blind toward the items that were not formally included into the ontology. Hence, tools like Magpie may help the user with obtaining a quick (but incomplete and possibly dirty) picture of what are the key items in a given text *from a particular perspective or viewpoint*. In other words, Magpie is not very helpful to tell the user details about the entities identified and annotated in the text, but it can quickly and reliably extract and show those terms that apply to a specific domain (say, politics). This can be useful in rapidly scanning large textual outputs or reports, where the user wants to efficiently ascertain whether the document deserves more attention or not.

5.4 OntoMat

The OntoMat framework uses a combination of Amilcare and C-PANKOW for its automatic annotation and ontology population. Unfortunately, the tools were broken at the time of testing, and undergoing a major overhaul, and so an evaluation on our corpus and ontology could not be carried out as we had hoped. We can report therefore only some details from previous experiments – these can be found in Section 5.8.

5.5 GATE

GATE itself can be used with a variety of different IE plugins. The most effective OBIE algorithm is probably the Hieron [LBC07]. We summarise briefly here the evaluation experiments performed with this algorithm in GATE, which were carried out jointly in the SEKT and KnowledgeWeb projects.

The experiments were performed on the Ontonews corpus [PABC05]. The articles in OntoNews were divided into three subsets according to the article's theme, namely business, international politics and UK politics, which has 91, 99 and 100 articles, respectively. The corpus was annotated manually according to the Proton ontology[1]. Our experiments described below used the *psys:Entity* branch of the Proton ontology that has 272 unique concepts. The news corpus was annotated with 167 concepts, of which 146 concepts are in the Proton ontology and are used in our experiments[2]. The concepts span from the 3rd to the 10th level of the hierarchical structure of Proton.

As there are no previously reported results on this corpus, we compare Hieron against two state-of-the-art "traditional" learning algorithms for IE: SVM and Perceptron. In these experiments, we used the uneven margins SVM and Perceptron with uneven margins (PAUM), instead of the standard algorithms, because the uneven margins algorithms have better performance than the respective standard models for IE (see [LBC05b]).

	Traditional F			Augmented F		
	PAUM	SVM	Hieron	PAUM	SVM	Hieron
Business	74.1	75.3	82.7	78.8	79.3	91.2
Pol-Int	77.1	80.1	83.3	83.0	85.9	91.3
Pol-UK	82.0	82.9	82.5	83.6	84.4	90.1

Table 5.3: Micro-averaged F_1 scores for Hieron algorithm in GATE

Table 5.3 presents the results of the three learning algorithms on the Ontonews corpus, measured by conventional micro-averaged F_1 (traditional F) and by augmented F-Measure which uses Augmented Precision and Recall calculated with the BDM (see Section 3.2). We can see that not only are the figures high in general, but also that Hieron outperforms the non-ontology-based IE methods measured by conventional F_1. The 5–10% improvement in results demonstrates that the IE algorithm benefits from taking into account the relationships between classes in the ontology and using this information during the learning process to optimise performance. Unsurprisingly, the Augmented F results are higher than the traditional F results, because they still give some credit for a partial error. They also give us more precise information by showing a greater difference between the various methods than the traditional F results, due to the finer granularity of the

[1]See http://proton.semanticweb.org.

[2]The other 21 concepts were proposals as extensions to the ontology, but have not been adopted yet and therefore are not used in these experiments.

measure. The comparison between F and Augmented F measure for GATE's results is shown in Figure 5.1.

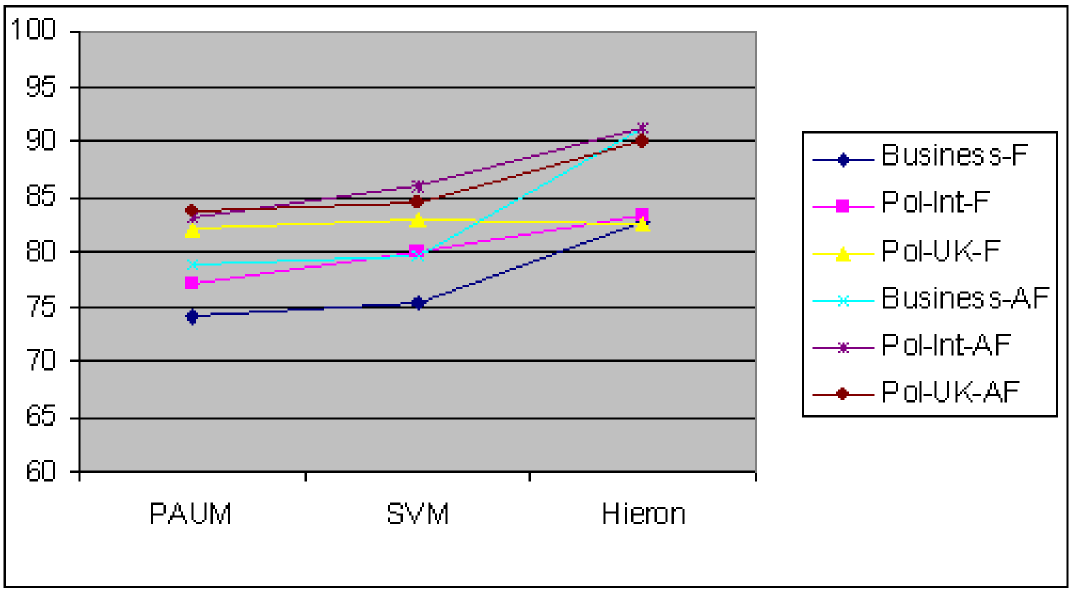


Figure 5.1: GATE results

5.6 KIM

We performed experiments with KIM on the same corpus and ontology as we used for GATE. In fact, KIM uses a wider range of concepts than GATE, referring to some common names as well as proper names. For example, "e-commerce" is classified as a "Business Abstraction", while "telecoms analyst" is classified as a "Profession" in the complete version, while in the narrower version such things would not be annotated. We could not compare KIM's results on this wider set of concepts with those of GATE, because current implementations of OBIE in GATE do not cater for such concepts. However, we do have a set of manual annotations for this wider range of concepts. So we were able to compare the wider range of concepts (which we shall refer to as "common-names") with the gold standard. In order to compare KIM's results with GATE, we also removed these extra common names from KIM's annotations, in order to produce a set of results matching the wider annotation set (which we shall call "proper-names"). We did this automatically using a set of JAPE grammars in GATE: it is important to note that for this reason there may be a few mistakes in this reduction. We were then able to compare the narrower set with the manually annotated proper-names set, and also compare these results with GATE's results. Tables 5.4 and 5.5 show the results for standard Precision, Recall and F-Measure, and for Augmented Precision, Recall and F-Measure (using the BDM), for the proper-names and common-names respectively.

Corpus	P	R	F	AP	AR	AF
Business	55.7	48.5	51.9	71.0	66.3	68.5
Pol-Int	42.1	44.0	43.1	63.1	66.3	64.7
Pol-UK	45.9	40.5	43.0	71.9	62.4	66.8
Total	47.1	44.3	45.7	70.9	66.3	68.5

Table 5.4: Evaluation for KIM on Proper Names

Corpus	P	R	F	AP	AR	AF
Business	56.9	31.9	40.9	88.0	46.3	60.7
Pol-Int	42.8	30.7	35.8	72.8	49.6	59.0
Pol-UK	24.2	18.4	20.9	39.8	29.3	33.8
Total	37.8	40.5	48.8	61.4	40.5	48.8

Table 5.5: Evaluation for KIM on Common Names

Unsurprisingly, the results for common names are a bit lower than those for proper names, because they are generally much harder to identify. The results for the political-uk corpus are much lower than for the other two corpora, for the common names. We assume this is because there is a much higher proportion of common names to proper names in the political-uk set: there are 5901 common names of which only 85 are matched, whereas there are 4518 proper names of which 1834 are matched.

5.7 Beagle++

In order to evaluate the performance of Beagle^{++}, the natural baseline that was considered is Beagle, since we need to prove that what we bring new to this system is worthy, and with what costs. The first category of experiments considered the performance in terms of time to index collections of data, the amount of extra data (metadata) generated and the response time for queries. The second type of experiments was done with human judges who rated the results that our system provided to personalised queries. Both sets of experiments were conducted on 11 data sets which consist of the data willingly provided

	Number of hits	
User	Beagle	Beagle^{++}
1	0	7
2	0	10
3	0	27
4	0	38
5	0	36
6	0	22
7	0	0
8	0	26
9	0	36
10	0	41
11	0	43

Table 5.6: Number of hits provided for a query with Beagle and Beagle^{++}

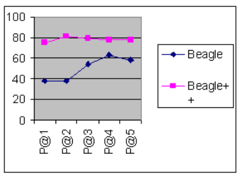
by our researcher colleagues: emails, documents, publications, address books, calendar appointments and other resources found on the users' desktops(.txt, .doc, .ppt, .html, etc.).

First, the data collections were indexed with both Beagle and Beagle^{++} and we observed the time to index for various dimensions of the data sets. On average, we had 1,421 resources per user, with an average size of 250,140 bytes per resource, which means approximatively 3.6 GB of data. The range of sizes varied from as little as 92 resources, to almost 4,000. The average time of indexing with Beagle was 19.8 minutes, while with Beagle^{++} this increased to 113.6 minutes. Even though this increase seems rather large, it is misleading, since this indexing is done only once, when Beagle^{++} is first run on a machine. Afterwards, when a new action occurs (creation, deletion, moving, renaming of a file), only the particular affected file is handled, which is transparent to the user and makes use of very few resources. The obvious gain in this indexing is the additional metadata generated by the annotation tools — on average, Beagle^{++} generated 62,119 triples per user, useful data which is used for a better retrieval.

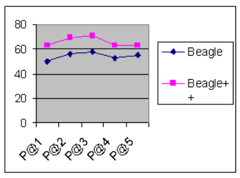
To measure the time of response for a query, each user proposed a personal query and a total of 11 queries were run against each data collection. 3 of the queries provided no result when Beagle was used as the search engine, and when Beagle^{++} was used for these queries, it provided more results in almost all cases. In Table 5.6 the number of results for the query "peer to peer" for all users is shown. In this case, no results were provided by Beagle. The response time for these queries increased from an average of 0.348 seconds for the query with Beagle, to 2.192 seconds with Beagle^{++}, which is still quite a good response time. For the rest of the queries, Beagle^{++} outperforms Beagle with a significant increase from an average of 5.714 results per query, to 18.156 results, which means a bigger range of possible positive responses for the user to choose from. Obviously, the response time also increased, from 0.372 seconds to 2.200.

Type of query	P@1		P@2		P@3		P@4		P@5	
	B	B^{++}	B	B^{++}	B	B^{++}	B	B^{++}	B	B^{++}
clear	0.38	**0.75**	0.38	**0.81**	0.54	**0.79**	0.63	**0.78**	0.58	**0.78**
ambiguous	0.50	**0.63**	0.56	**0.69**	0.58	**0.71**	0.53	**0.63**	0.55	**0.63**

Table 5.7: P@1-5 for querying with Beagle and Beagle^{++}



(a) clear queries



(b) ambiguous queries

Figure 5.2: Comparison of Beagle and Beagle^{++}

For the second round of experiments, each user was asked to propose 4 queries related to the content of their data collection: 2 which had a clear meaning (one sense only), and 2 which were considered ambigous (with at least 2 senses, e.g. "Java"). For the top-5 results, the user was asked to rate a query result with 0 (not relevant) or 1 (relevant). The user needed to consider a result only as relevant or not, disregarding the extent of the relevance.

We measured the quality of the produced annotations using *precision*, a standard IR (Information Retrieval) evaluation measure. As the results had a confidence score, we computed precision at different levels, namely P@5, P@4, P@3, P@2, P@1. The precision at level K (P@K) is the precision score when only considering the Top-K output. It represents the number of relevant query results within the Top-K results divided by K, the total number of results considered. First, the P@K scores were computed for each user and query, then we averaged these values over the 2 queries of each type (clear and ambiguous), obtaining the user's opinion on some type of query. We further averaged over all subjects and the resulting values are listed in table 5.7, where B represents the results obtained using Beagle and B^{++} using Beagle^{++}. These results are also shown graphically in Figures 5.2(a) and 5.2(b) for the clear and ambiguous query types respectively.

In all cases, we observe that Beagle^{++} outperforms Beagle (we highlighted the best results in the previous table), but has weaker results for the ambiguous queries, which means that the power of our current system to disambiguate between senses is not so de-

Tool	Precision	Recall	F-measure
MnM	93.9	77.5	84.9
KIM	100	82.5	90.4

Table 5.8: Evaluation on Structured Corpus

veloped, and this suggests a future direction of research, to be able to detect the ambiguous queries at run time and treat them appropriately.

5.8 Other experiments

Various other experiments have been performed on annotation tools in the past: we report here on some of the main findings which concern us.

5.8.1 Sazedj and Pinto

[SP05] report on a set of experiments carried out with 5 ontology-based textual annotation tools: KIM, Melita, MnM, OntoMat and C-Pankow. However, the evaluation was flawed in some ways. First, due to technical reasons, they were only able to evaluate OntoMat in its manual mode, so we can discount this tool from the performance evaluation. Second, the creation of the manually annotated corpus used as a gold standard had quite low inter-annotator agreement (IAA), apparently because clear guidelines were not issued to the human annotators and there was much confusion over what exactly should be annotated and how (even down to very basic issues such as whether multiple occurrences of the same entity should be annotated in a document). This is not uncommon for manual annotation, but it remains a problem for the purposes of the experiment. Third, it was found that the semi-automatic tools Melita and MnM failed to create annotations on unstructured corpora such as news texts, so they could only be evaluated on structured corpora such as the tabular one used in their experiment. However, one of the advantages of the experiment is that it reflects well the richness of metadata produced by the tools, something that is often overlooked in evaluations.

The tabular corpus was created especially for the experiment and consisted of a 2-row table, the left column containing the concepts and the right column containing the instances. The annotation task consisted of matching the instance with the appropriate concept. A simple `Conference` ontology was also designed for this experiment. Table 5.8 shows the results of comparing KIM and MnM on this corpus.

On the unstructured corpora, KIM was compared with C-Pankow, which is the IE engine typically coupled with OntoMat. Table 5.9 shows the results on two corpora - Planet Visits and Baha'i. The Planet Visit corpus is a well known corpus from KMI

Tool	Precision	Recall	F-measure
Planet Visit			
KIM	63.8	54.8	59
C-Pankow	21.2	13.4	16.4
Baha'i			
KIM	70.6	60	64.9
C-Pankow	36.4	30	32.9

Table 5.9: Evaluation on Unstructured Corpora

(Open University) about visits of other people to and from the university, while Baha'i is a set of international news articles from the Baha'i International Community world news service.

From these small-scale experiments, we can nevertheless see that KIM stands out as the best performing tool out of the three, although as we have mentioned, the evaluation was somewhat flawed. Clearly the task on unstructured corpora is much harder than that on structured corpora, as reflected by the results. However, as the authors point out, KIM may not work so well on a more domain-specific text where its ontology (at the time KIMO, now PROTON) is not so useful. The authors also state that when comparing KIM with the semi-automatic tools such as MnM, "the future of the latter tools is clearly under question taking into account the simplicity of the former".

5.8.2 Dot.Kom experiments

The EU project Dot.Kom evaluated various information extraction and knowledge management tools on a set of testbeds. Among these were the annotation tools Magpie and OntoMat, for which we summarise below the results obtained. It is important to note that in this project, they did not directly compare most of the tools, but performed different experiments according to the capabilities of the tool.

Magpie was evaluated on its basic Named Entity recognition capabilities, given a set of entities stored in an ontology. The ontology used was the AKT reference ontology[3] which describes the structure of academic institutions and represents things like organisations, people, contact details, projects and publications. Such entities are generally quite easy to recognise as they are not that ambiguous and many of them consist of quite regular patterns. The evaluation consisted of recognising and classifying entities of three types – Person, Organisation and Research Area – in a dataset of 22 web pages. However, some of these web pages had actually been used to develop Magpie originally, so there was a small overlap between the data used for training and testing. The results for Magpie are shown in Table 5.10. We can see clearly that Precision is high while Recall is low. This is

[3]http://www.aktors.org/publications/ontology

Entity type	Precision	Recall	F-measure
Person	91.3	11.2	19.95
Organisation	95.2	12.2	21.63
Research Area	99.6	39.8	60.34
Average	95.37	21.07	34.52

Table 5.10: Evaluation of Magpie with AKT ontology

System	No. of concepts	Accuracy
MUC-7 systems	4	>90%
Fleischman	8	70.4%
PANKOW	59	24.9%
Hahn	196	21%
Alfonseca	1200	28.26%

Table 5.11: Evaluation of PANKOW

because Magpie can largely only find entities that it already knows about, i.e. which are contained in its ontology, and cannot match against unknown people or organisations. So it is quite limited as soon as it is used in a domain outside that in which it has been trained and for which a known set of entities already exists.

Dot.Kom also evaluated the Pankow method implemented in OntoMat, which is essentially the "Self Annotating Web" paradigm described in [CHS04]. The main idea of this paradigm is to disambiguate instances of an ontology (with respect to the concepts they belong to) using a large number of linguistically motivated patterns. They compare the results with various other approaches, in terms of accuracy. A summary is shown in Table 5.11. However, it is quite difficult to draw any conclusions, because each method reported has been tested on a different number of concepts. Clearly the fewer concepts among which to disambiguate, the easier the task and the higher the expected results, so while MUC-7 systems achieved more than 90% accuracy, we would expect this figure to drop dramatically when tested on the 59 concepts used by PANKOW or the 1200 used by Alfonseca's system. It is important to note also that PANKOW itself uses no preprocessing, while other systems use various methods of pre-processing such as N-gram extraction or syntactic and semantic analysis.

5.9 Multimedia annotation tools

As discussed in Chapter 3, the multimedia annotation tools only perform manual annotation and therefore cannot be evaluated for performance as such, although, as mentioned previously, there are some dimensions which could be targeted for future research in this area.

5.10 Discussion

It is clear from these evaluations that automatic annotation (using ontology-based information extraction techniques) is a much harder task than traditional information extraction, where systems can typically reach F scores in the 90s. This is unsurprising, given that traditional information extraction tasks consist of assigning one of only a handful of possible categories to an instance in the text, and that such categories are very coarse-grained (e.g. Location as opposed to City). This also requires that different techniques need to be used: for example, while hand-coded rules can be written for IE over a set of 10 concepts, it would be far too time-consuming to develop such rules to deal with more than 100 concepts.

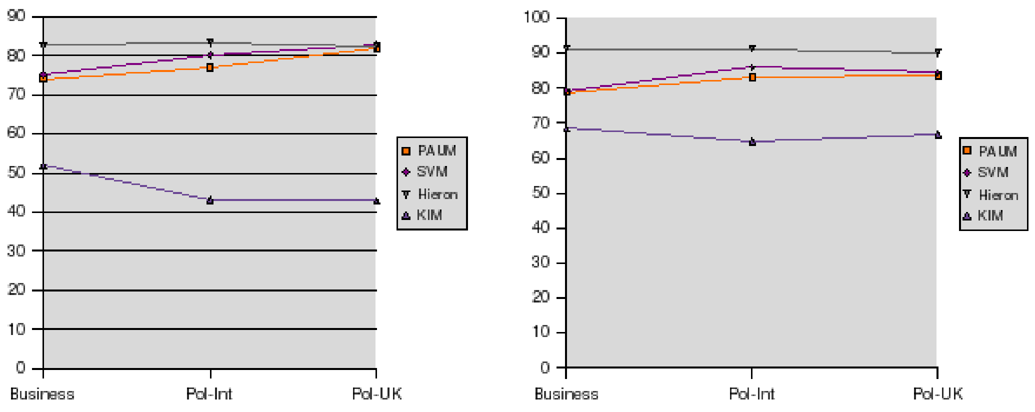
We can also see that the quality of the system is largely dependent on several factors:

- the number of concepts in the ontology and the level of granularity at which the system is required to perform;
- the domain of the text (in general, the more specific the domain, the easier the task)
- how closely related the task and domain are to that expected by the system. By this we mean that if a system was developed to find instances of general concepts in a news domain, and we test it on a very specific corpus of texts about meetings, we may get lower results than if we tested it on a news corpus.

Traditional information extraction systems can be compared very easily using standard annotated test sets such as the MUC series, CONLL and so on, due partly to the series of initiatives aimed precisely at this evaluation task, and partly due to the fact that there are no real interoperability issues for the systems apart from the ability to import and export annotations in certain formats (and converters are generally available if not).

It is much harder to compare the performance of textual annotation tools in this way due to various factors:

- no specific large-scale initiatives comparable to MUC etc.;
- the lack of existing annotated data for testing;
- the widely differing aims of annotation tools (most are not aimed at recognising instances of Proton concepts from news texts, for example);
- the previous lack of a common metric for evaluation that takes into account a more flexible approach than traditional Precision and Recall;



(a) F-measures

(b) AF-measures

Figure 5.3: Comparison of GATE and KIM

- the quite low scores generally achieved by systems (people are less inclined to publicise their results when scores are low);
- interoperability issues (not all systems can input and output OWL files; many use their own ontologies and are quirky in design)

Thus the only real performance **comparison** that can be made is between the two most similar tools, GATE and KIM. These are the only two which can be directly compared: for reasons discussed earlier, we had problems when trying to compare the other tools on our test data. GATE and KIM, on the other hand, use the same API, share some components, were both designed with roughly the same goals in mind, were trained on the same kinds of data, and were developed in collaboration although by different institutions. Even so, we found that the results are not directly comparable, in that we had to perform post-processing on some of the annotations produced by KIM, in order to perform an identical evaluation to that run on GATE.

On the Ontonews corpus, our results show that all three of the ML algorithms tested in GATE outperformed KIM, on the proper-names corpus, achieving results in the 80s as opposed to in the 40s. Figures 5.3(a) and 5.3(b) show the comparison of results for GATE (using all three algorithms) and KIM, for the F-measure (using traditional Precision and Recall) and Augmented F-measure (using BDM). Interestingly, the difference between the two systems is less when using the BDM: it is possible that KIM makes many small errors in determining the ontological class (as opposed to simply not recognising something at all or finding false positives), which therefore get some credit and thus increase its score over the traditional metric which gives no credit for such mistakes.

Previous evaluations of Ontomat / C-PANKOW indicate quite low results on this kind of task, indicating that it is best used for other kinds of task rather than finding instances of Proton-type concepts. According to previous evaluations, MAGPIE achieved very high Precision but low Recall on tasks and using ontologies close to those for which it was designed. It is not very flexible in its application, however, like most of the annotation tools tested. Both MnM and KIM performed very highly on the simple task of finding instances in structured data, but MnM performs dismally on any kind of task where the relevant concepts are not explicitly mentioned, i.e. on unstructured data.

Chapter 6

Scalability of Annotation Tools

6.1 Introduction

Gartner recently reported that for at least the next decade more than 95% of human-to-computer information input will involve textual language. They also report that by 2012 taxonomic and hierachical knowledge mapping and indexing will be prevalent in almost all information-rich applications. The web revolution has been based largely on human language materials, and in making the shift to the next generation knowledge-based web, human language will remain key.

Clearly, manual semantic annotation is infeasible on a very large scale, as it is too time-consuming for humans to hand annotate large volumes of text, or even small amounts of text with huge ontologies. So to scale to real-world applications, there must be some form of automation.

As discussed in KnowledgeWeb deliverable D1.4.2 version 2 [PLM$^+$07], HLT is probably one of the hardest domains in which to make the transition between a research prototype and an application that is suitable for real use in industry, largely because of the nature of language itself. The inherent difficulties with natural language processing tasks (such as incompleteness, language change etc.) make it difficult for HLT applications to scale up to the real world. One solution here is to think specific. In the field of machine translation, it was originally thought that automatic translation systems would be general enough to use in any situation, for any kind of domain and task. However, it was soon found that this was going to be near impossible, and once research focused on explicit tasks in very specific domains, real world applications became possible. So it is with semantic web applications involving human language technology. Language processing tasks become really accurate and usable when they are tightly focused and restricted to particular applications and domains.

Thus, one aspect of scaling up to the real world is to combat the problems of accuracy when systems are deployed in industrial applications, such as the enabling of public metadata-on-demand services. However, even in a closed domain with a tightly focused application, there are more mundane issues concerning the sheer volume of data, with respect to storage, processing speed and power. Scalability has been identified as a critical issue for the processing of large volumes of data, so that statistical IE algorithms can be designed and trained (since these require large amounts of annotated training data in order to be accurate).

In this chapter, we discuss some research designed to investigate the feasibility of scaling annotation tools up to the demands of the real world, i.e. performing annotation on an industrial level with massive volumes of data and/or huge ontologies.

6.2 SWAN

SWAN was a recent experiment in scaling up automated metadata extraction for industrial strength Semantic Web application development. The SWAN project leverages language technology to advance a number of existing elements covering both basic research and applications demonstrators. The SWAN experiment has enabled its developers to determine the issues and problems in large-scale deployment of the technology in applications such as HLT for the Semantic Web, EAI and e-commerce applications, and to gain experience in scaling up their systems and in their application in next-generation knowledge access and management.

SWAN was based on a combination of three existing systems: GATE, KIM and SECO (a focused crawler and integration site for Semantic Web Metadata, hosted at DERI Galway). It had the following subtasks:

- Ontology-aware IE: performs IE against a rich taxonomy/ontology of over 300 classes in open-domain text.
- Entity IR: instance disambiguation and indexing.
- Instance base: gathering and evolving world knowledge
- Mixed-initiative IE for user-adapted domain-specific extensions.

6.3 Massive semantic annotation – the KIM Cluster Architecture

Following on from the SWAN experimental research, the **KIM Cluster Architecture** has been developed, partly within the context of the SEKT project, with the specific aim of

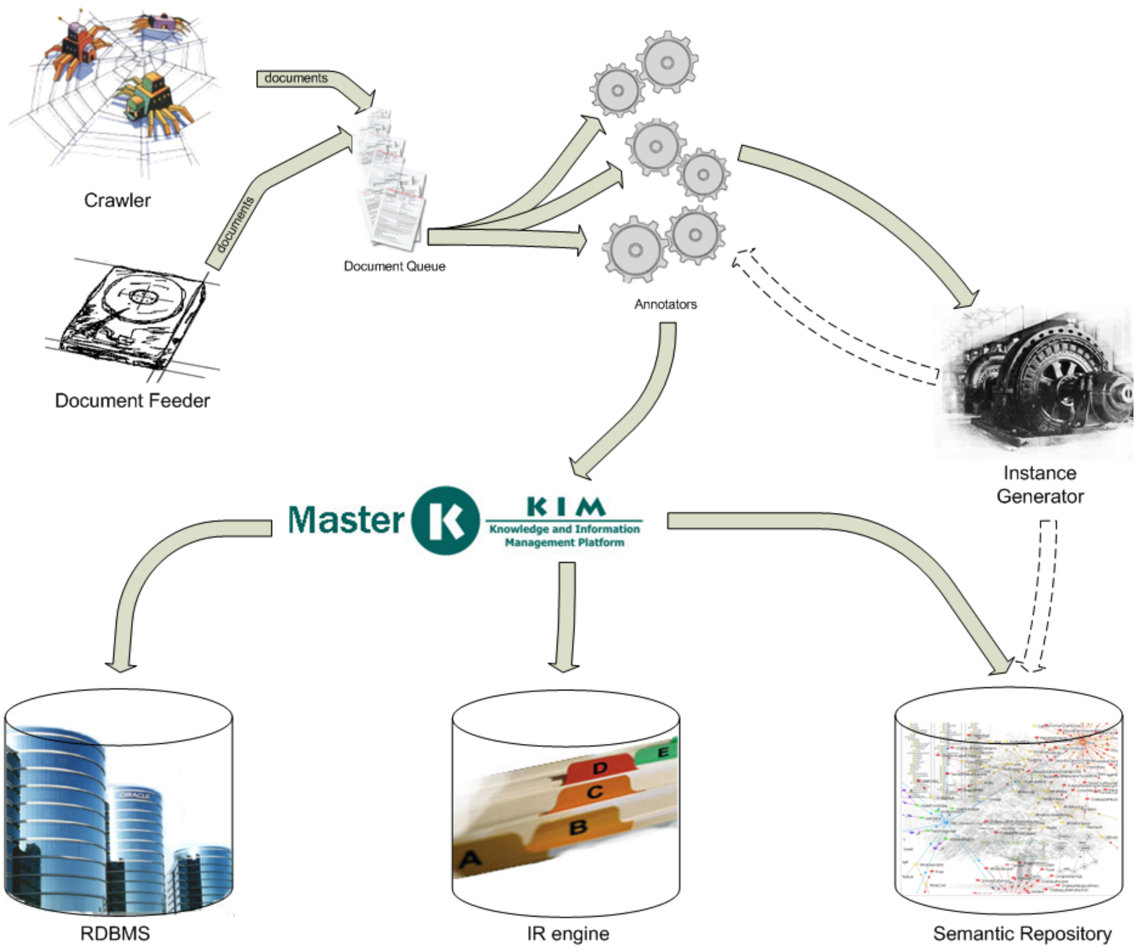


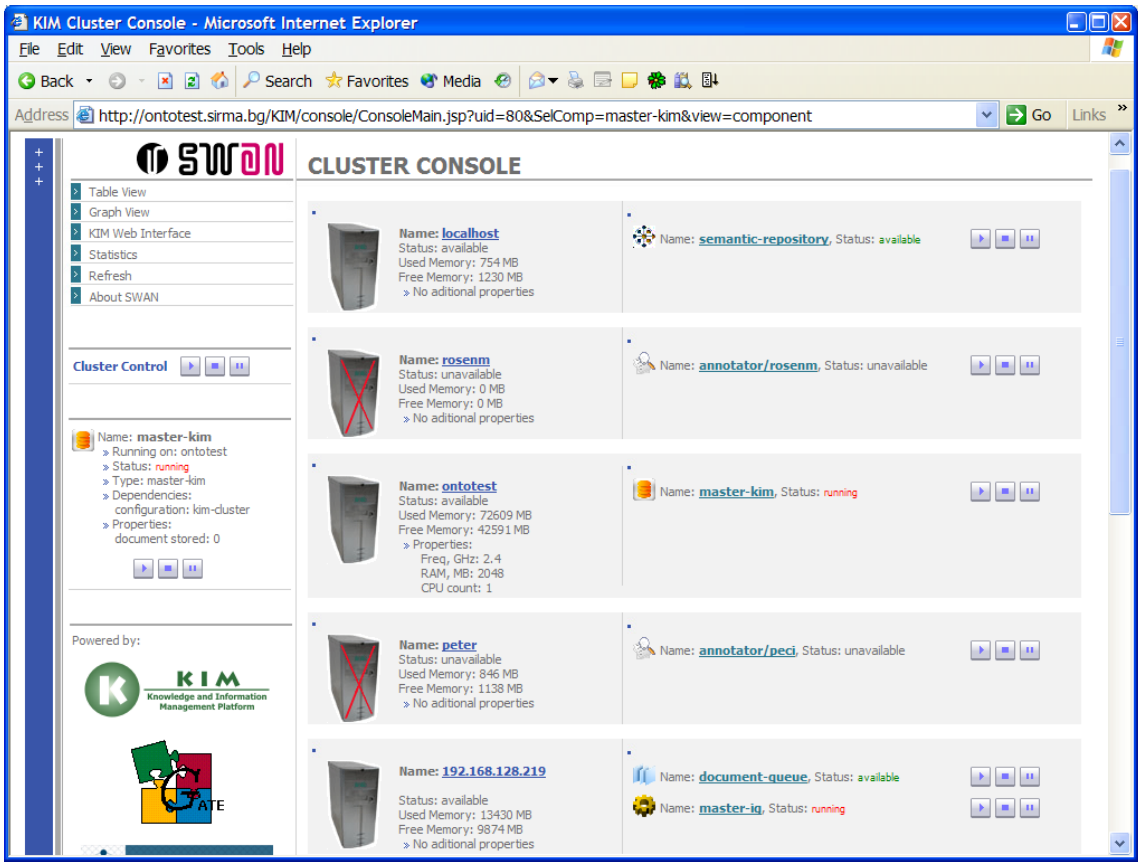
Figure 6.1: Architecture of KIM Cluster

extensive scaling of the existing KIM model. More information about both the architecture and its performance can be found in [MP05]. An illustration of the architecture of the KIM cluster is shown in Figure 6.1.

Amongst other features, the KIM Cluster Architecture provides:

- support for a virtually unlimited number of annotators (the components which perform the computationally most expensive processing);
- centralised ontology storage and querying;
- centralised metadata (annotations) and document storage;
- support for multiple crawlers or other data sources;
- dynamic reconfiguration of the cluster (e.g. starting new crawlers or annotators on demand)

Figure 6.2 shows a screenshot of the cluster management console in use.


KIM Cluster Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://ontotest.sirma.bg/KIM/console/ConsoleMain.jsp?uid=80&SelComp=master-kim&view=component Go Links
SWAN
Table View
Graph View
KIM Web Interface
Statistics
Refresh
About SWAN
Cluster Control
Powered by:
KIM Knowledge and Information Management Platform
GATE

CLUSTER CONSOLE

Name: master-kim
» Running on: ontotest
» Status: running
» Type: master-kim
» Dependencies:
configuration: kim-cluster
» Properties:
document stored: 0

Name: localhost
Status: available
Used Memory: 754 MB
Free Memory: 1230 MB
» No aditional properties

Name: semantic-repository, Status: available

Name: rosenm
Status: unavailable
Used Memory: 0 MB
Free Memory: 0 MB
» No aditional properties

Name: annotator/rosenm, Status: unavailable

Name: ontotest
Status: available
Used Memory: 72609 MB
Free Memory: 42591 MB
» Properties:
Freq, GHz: 2.4
RAM, MB: 2048
CPU count: 1

Name: master-kim, Status: running

Name: peter
Status: unavailable
Used Memory: 846 MB
Free Memory: 1138 MB
» No aditional properties

Name: annotator/peci, Status: unavailable

Name: 192.168.128.219
Status: available
Used Memory: 13430 MB
Free Memory: 9874 MB
» No aditional properties

Name: document-queue, Status: available

Name: master-iq, Status: running

Figure 6.2: Screenshot of KIM Cluster Architecture

The architecture relies on the KIM World Knowledge Base, which contains a quasi-exhaustive coverage of the most popular named entities in the world. This aims to include entities of general importance, such as those typically found in the news, and to cover what an average person would be expected to know about beyond the scope of their own domain of interest, country, job, hobbies etc. It involves names of people, locations and organisations, collected from various sources such as geographical and business intelligence gazetteers, and is dynamic since it "learns" from texts. Table 6.1 shows the scale of the entries, in both the smaller version and the full scale version.

In order to manage the ontologies and Knowledge Bases, KIM makes use of OWLIM, which is a high-performance Sesame Storage And Inference Layer (SAIL) with OWL inference. There are two versions of OWLIM:

- SwiftOWLIM performs reasoning and query evaluation in-memory, while a reliable persistence strategy assures data preservation, consistency, and integrity. SwiftOWLIM is the fastest RDF(S) and OWL engine to date. Even on a desktop PC, it can load

	Small KB	Full KB
RDF Statements		
explicit	444,086	2,248,576
after inference	1,014,409	5,200,017
Instances		
Entity	40,804	205,287
Location	12,528	35,590
Country	261	261
Province	4,262	4,262
City	4,400	4,417
Organization	8,339	146,969
Company	7,848	146,262
Person	6,022	6,354
Alias	64,589	429,035

Table 6.1: Size of KIM World Knowledge Base

and infer over 7 million statements and has a processing speed of 40,000 statements per second.

- BigOWLIM operates directly with binary persistence files, which allows it to scale to billions of statements. BigOWLIM is the only engine which offers OWL inference over 1 billion triples. On average, each entity is described by 10 RDF statements, which means that BigOWLIM can handle 100 million entities. KIM itself can index and manage a million documents on a server worth approximately 3500 euros.

The system has been demonstrated on over 1.2 million news articles, which have been processed with the KIM Cluster architecture. This resulted in the recognition of over 1 million named entities (50,000 of which were pre-populated). These were all stored in the semantic repository based on OWLIM, along with their semantic descriptions (attributes and properties). The average speed of annotation in this experiment was 10KB per annotator node/component.

6.4 GATE

GATE is an infrastructure for performing language engineering tasks and as such it can be used internally by applications that generate semantic annotation through the use of HLT. In fact both SWAN and KIM mentioned previously employ GATE as the supporting technology for their OBIE work.

In order to support scalability of applications built on top of GATE, the HLT infrastructure provides very efficient implementations for a set of basic text processing algorithms and supports generic text processing tasks through the JAPE engine [CMT00]. As these implementations are provided directly with the GATE platform, they are optimised to make best use of the GATE API, thus reducing the storage requirements and execution times.

Another element that is relevant for scalability is the support for workflow control implemented in the GATE controllers. These are designed for optimising the memory use while processing large textual collections and provide support for dynamically modifying the execution flow in the case of heterogeneous data streams.

6.4.1 Serial Analyser Controllers

GATE uses serial controllers to provide a method of executing a set of processing resources (PRs) in sequence over a document. This is fine for very small scale applications, but in most cases, we want to process large amounts of data and we therefore need to run processes over a set of documents contained in an entire corpus, rather than just over a single document. For this, GATE uses Language Analysers, which are corpus-aware PRs. In order to take advantage of their capabilities, GATE also has a Serial Analyser Controller that extends the functionality of the Serial Controller. Given that GATE corpora are lists of documents, and any pipeline-style controller holds a list of PRs, the task of the Serial Analyser Controller is to run all the member PRs (or, more specifically, Language Analysers) over all the documents in a corpus. The strategy applied is to run the whole set of PRs over each of the documents, moving on to the next document as soon as the full processing of the current one is finished, rather than running each PR over all the documents and then continuing with the next PR. The reasoning for this is as follows.

GATE corpora are not limited in size - they can contain an indefinite number of documents. In practice, this number is limited by the number of documents that can be loaded into memory at the same time, which is not very large (e.g. a few hundred typical web pages). To circumvent this restriction, GATE provides the concept of DataStore, which is a mechanism for storing in a persistent manner all sorts of Language Resources, mainly documents and corpora. This allows users to create corpora that are as large as they can fit onto their computer's hard drive which, while still limited, usually exceeds normal requirements. For corpora stored in a DataStore, GATE provides a persistent corpus implementation that can transparently load documents from the DataStore when they are needed, and has API calls for unloading documents from memory when not required any more.

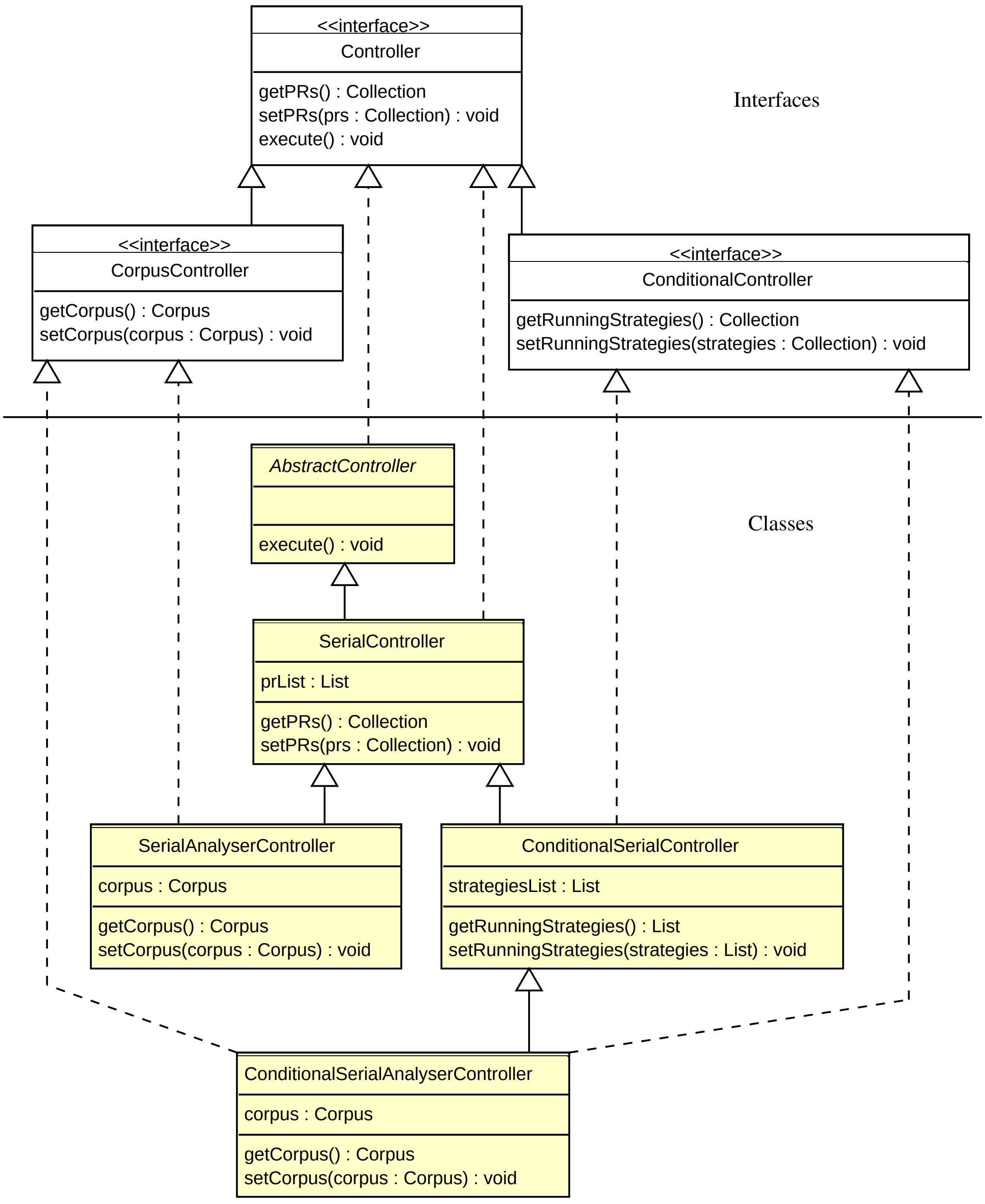
During the design of the Serial Analyser Controller, we needed to cater for the use of serial corpora and datastores. The operation of loading documents from the datastore is expensive in terms of time complexity, so it needs to be used sparingly to preserve efficiency. When comparing the two execution strategies described above, it becomes apparent that in the first case each document will need to be loaded and unloaded only once during the whole process, while the second strategy would require each document to be loaded and unloaded once for each processing resource contained in the controller.

6.4.2 Conditional Controllers

Serial Analyser Controllers are capable of processing all the documents from a corpus in a uniform manner. This can improve productivity during the development phase of an IE system, but can also cause problems when the documents in an input corpus are not homogeneous in nature. For instance let us consider a corpus obtained from crawling a set of web sites: in this case the large majority of the documents will be web pages written in some version of HTML. However, other types of documents might be included occasionally, such as PDFs. If we also assume that HTML and PDF documents need to be treaded differently (e.g. a different type of analyser needs to be run to extract data regarding the page formatting), then the system developer is faced with a problem. A similar problem occurs when the input corpus contains documents in different languages or even documents about different topics. To address this, GATE also provides another kind of controller that is able to dynamically change the execution flow according to the features of the current document, called the Conditional Controller. Implementations of this are provided for both Serial Controllers and Serial Analyser Controllers, as shown in Figure 6.3.

A Conditional Controller associates with each of the member PRs a running strategy that is used to decide whether a particular PR should be run or not. In the case of the Conditional Serial Controller, the decision is manually set by the user before executing the controller, and can be changed between runs using the graphical interface. This can be useful during the development phase for temporarily disabling the execution of some components in the IE system in order to perform experiments. However, the real benefit is obtained in the case of Conditional Serial Analyser Controllers, where the decision is based on the value for a particular feature on the current document and is dynamically calculated at execution time. For example, one can envisage a system where the input comprises documents in different languages. The first analyser in the pipeline is used to identify the document language which is then marked by means of a feature on the document. All the subsequent PRs can be run or not depending on whether they are capable of processing documents in that particular language.

This hierarchy of controller types as well as their associated user interfaces enables rapid development for IE systems. They allow the definition of various pipeline architectures, the processing of large corpora through the use of datastores, the flexibility of dynamically modifying the system configuration depending on the input by using conditional controllers. All these facilities are aimed at helping the system engineer achieve the desired results as quickly as possible.



<<interface>>
Controller
getPRs() : Collection
setPRs(prs : Collection) : void
execute() : void
Interfaces
<<interface>>
CorpusController
getCorpus() : Corpus
setCorpus(corpus : Corpus) : void
<<interface>>
ConditionalController
getRunningStrategies() : Collection
setRunningStrategies(strategies : Collection) : void
AbstractController
execute() : void
Classes
SerialController
prList : List
getPRs() : Collection
setPRs(prs : Collection) : void
SerialAnalyserController
corpus : Corpus
getCorpus() : Corpus
setCorpus(corpus : Corpus) : void
ConditionalSerialController
strategiesList : List
getRunningStrategies() : List
setRunningStrategies(strategies : List) : void
ConditionalSerialAnalyserController
corpus : Corpus
getCorpus() : Corpus
setCorpus(corpus : Corpus) : void

Figure 6.3: Execution Controllers Hierarchy

6.4.3 Experiments

The scalability of the GATE platform has been demonstrated through a set of previous experiments that ran various language processing tasks over large textual collections. Probably the most relevant in the context of the Semantic Web is the processing of the whole British National Corpus (BNC) with the ANNIE named entity recognition application. While ANNIE itself is not focused on a particular domain, and so it should not be used directly for OBIE, it is based on the type of technologies that would be used for an actual application aimed at producing semantic annotation.

The BNC, on which ANNIE was run, contains around 100 million words in 4124 documents, and is around 11GB in size (as XML). It took approximately 14 days for ANNIE to run over the entire corpus. Similarly it was run on a set of around 113,000 news articles containing a total of 62 million words (around 440MB) and took approximately 10 days. It is important to note that these experiments were performed several years ago, with an older version of GATE which was not optimised for performance and which had some memory leaks, and also on slower machines than currently in use; we would expect performance to have roughly doubled in speed since then.

As for ontology-based IE, the machine learning API in GATE is capable of running on huge amounts of data, e.g. very large ontologies (size dependent on the hard disk size) and thousands of documents for training (depending on memory size and – for some learning algorithms – hard disk size). This is mainly due to the datastore mechanism in GATE, which, as described above, stores immediate data and results on the hard disk rather than in memory, loading and unloading documents as they are processed.

We applied the machine learning OBIE application in GATE to the BNC corpus in order to assess the scalability of GATE, and in particular the machine learning facility. Our experiment was done on a Linux server with 3.00 GHz Intel CPU and 4G memory. We first learned the SVM models from the OntoNews corpus containing 290 news articles. The documents in the corpus were annotated with the entities corresponding to the 146 concepts in the Proton ontology. We used the binary linear SVM algorithm and learned two SVM models for the starting token and last token of one type of entity, respectively. So basically the learned models contained 292 SVM classifiers. The training took 14.8 hours and the peak memory was about 25%.

Then we applied the learned SVM models to the BNC corpus, as described above. Many documents are quite large. All the documents were stored in one GATE datastore. In the application, the system first loaded one document into the memory, processed it, saved the processing results and finally unloaded it from the memory. Hence the memory used in the application did not change much, and used around 20% of total memory depending on the actual size of the document. When the SVM models were to the BNC corpus, it took 6 days for the first 400 documents. So it probably would have taken about 60 days for the whole corpus. We noticed that most time was spent on obtaining the NLP features from the ANNIE pre-processed documents, which involved a procedure of sorting all tokens in one document by their positions in the document. The token sorting procedure may take quite a long time for a large document. So we stopped the application directly on the original documents and split the documents into smaller ones, each of which contained at most 400 lines of the original document. After splitting we obtained 85,552 documents. It then took 217.8 hours (about 9 days) to process all the documents. Finally we easily added the split documents into the original ones with the machine learning annotations.

6.5 MnM

As we have already mentioned, MnM was not really designed for large-scale operations, but rather as a proof-of-concept tool. It demonstrates very clearly some of the problems involved in trying to scale such research prototypes into practical tools usable by the community.

For realistic cases, MnM (coupled with Amilcare) requires that a single concept at a time is used to mark up training documents, and that each concept is trained separately. It also requires between 20 and 30 webpages to be marked up for each learning phase. Generally this means that if there are 100 concepts in an ontology, there would need to be up to 3000 manual markups in order to obtain good results. Clearly this is infeasible.

There are further usability issues with MnM when scaling up. For example, if a document contains more than one instance of a concept, then the tool may not be able to allocate the correct properties to the correct instance, because it is unable to differentiate between them. Second, for each ontology class, the file has to be annotated and saved separately. Clearly this is not very user friendly or even feasible once the ontology becomes more than a very simple toy example. Also the learning has to be repeated for each ontology class used in the annotations.

Finally, the XML annotation tags are generated from the class/instance names. Documents containing the annotations can be saved locally as HTML, XML and text. However, tagged annotations are built by concatenating the first letter of each word in the concept and the full attribute name, withouth any other reference to the ontology. So for example, an annotation for the class "Conference" with the attribute "has-location" would be $< c\_has - location >$...$< /c\_location >$. If there is any other class beginning with the letter C then this would be ambiguous which concept the attribute was attached to. For any ontology of a significant size, this would create a real problem.

6.6 Magpie

Magpie is a web browser plug-in that was designed to offer the user a glimpse into the Semantic Web and to do it in real time, with no additional time delays experienced by the users. In terms of scalability, it makes sense to assess Magpie in terms of the lexicon size it can load and subsequently use to annotate the web pages visited by the user. So far, the largest lexicons Magpie has been exposed to is the Agrovoc – a multilingual thesaurus from the Food and Agriculture Organization of the United Nations that covers

15 languages, each with around 55,000 unique terms (to which we need to add some simple variations). With these lexicons, Magpie was able to load the lexicon and highlight web pages within one second. Note that the length of a web page may also affect the performance, but even extensive pages (like some in WikiPedia.org) did not throw the tool off the track.

One usability issue that we have encountered in connection with Magpie is related to its depiction of the top-level classes as labelled buttons, and placing those buttons in the web browser's toolbar. Even on larger desktops, we managed to meaningfully use maximum eight to ten top-level categories (button labels). More could be obviously used but they would get hidden in a pull-down menu, which to some extent defeats the point of showing the annotations visually. Nevertheless, this is not a major issue, as the top-level categories used to label the buttons are not necessarily the actual classes from a particular ontology. Thus, even for large ontologies with many classes that the user may potentially be interested in, one can cluster the classes and associate the button labels with some of the more abstract classes in ontology branches or sub-networks, or alternatively, devise the labels to thematically cover a sub-set of classes. An example may be to cluster concepts like 'Political Persons', 'Political Organizations' and 'Governments' into a new label 'Political' – as we did in our studies.

Nonetheless, there is another aspect that needs to be mentioned in connection with scalability of this tool. Magpie's annotations are only the precursor that enables the user to access content and web services semantically related to the annotated entity. With respect to this, Magpie is open as to the location of the services as well as to their number. Hence, the scalability per-se is not the main goal for Magpie; scalability has to be always considered with the computational effort and, even more importantly, with the time burden it places on the user. For a tool like Magpie, it would be futile to increase the scale without maintaining the real time response (i.e. ideally, within a second).

6.7 Ontomat

Ontomat was not designed for large-scale operations. It was designed as a research tool for annotation. It was the first of that kind within the compared tools here, being actively developed between 2000-2004. The development stalled in the meantime and only recently did work on it begin again. The scalability of Ontomat is defined by the underlying technology it uses. For pure manual annotation it is defined by the scalability of the Jena API, i.e. the number of instances and classes it can deal with are restricted by the number of instances and classes that the Jena component can deal with. For the semi-automatic annotation, Ontomat supports two modes: i) machine learning-based annotation using Amilcare, ii) linguistic pattern-based annotation, viz. Pankow.

Learning with Amilcare requires at least 10 positive examples of a named entity, i.e. for an instance of an Concept. Also the learned algorithm only works then within a similar class of web pages, e.g. hotel web pages, homepages, etc. Hence, the Amilcare approach requires extensive training, scales only with a small ontology and with a limited set of web pages. The Pankow approach is domain independent and works with any ontology and on any web page. However it requires a lot of Google queries, which restricts its scalability. Hence, the Pankow approach scales only within a small ontology.

Ontomat scales well with respect to the kind of metadata it produces, i.e. when it has write access to the document it stores the annotation within the document, otherwise it creates an annotation that is stored in an annotation server, that refers to the original document. With this reference technique it allows fine-grained and overlapping annotations.

6.8 Beagle^{++}

Beagle^{++} is a desktop search engine which takes advantage of the various semantic information created by its number of metadata generators. It does not involve the user in any way regarding the annotations that it makes, i.e. it is all fully automated, even taking care of new resources that are added to the system, moved or even deleted. By its very nature, Beagle^{++} deals with a large quantity and range of information, namely the whole collection of data that each user has on their desktop. It is well known that these collections are growing constantly, reinforced also by the advance in hard-disk storage space, which is also growing fast, easily reaching a few hundred gigabytes. This means that Beagle^{++} is capable of indexing the whole plethora of resources found on one's desktop and of generating metadata attached to numerous resources.

Even though the ontology behind the tool is rather small, the data volume it processes is quite large. All the generated annotations are stored in an RDF repository which is able to store and process the metadata received from various metadata generators, making it further available for querying or other operations needed by other modules/metadata generators. Just like KIM, for example, it supports theoretically any number of metadata generators, tailored for specific needs, and it also benefits from the centralised RDF repository for storage of the generated metadata, the Sesame Storage and Inference Layer (SAIL). All these factors make Beagle^{++} a highly scalable tool, dealing with constantly increasing amounts of data and metadata.

6.9 Multimedia annotation tools

Recent reports on multimedia content on the Web show an exponential increase in the content made available: 1.6 billion images and over 50 million audio and video files on Yahoo (YSearchBlog, August 2005); 2,187,212,422 indexed images (Google, August 2005); YouTube reported in July 2006 100 million video viewing and 65000 video uploads per day, while over 1 billion song downloads were recorded from iTunes launch

(April 2003) till February 2006. Clearly, given such numbers, manual annotation becomes strictly prohibitive, and in certain cases (such as annotating the digitised content of a museum) particularly costly. Yet, as previously discussed, the existing multimedia annotation tools target solely manual annotation, mainly due to the lack of robust, sufficiently general, algorithms for providing semi or fully automatic annotation. The relevant multimedia semantic analysis literature which, as noted earlier, has produced a plethora of machince learning and knowledge-assisted approaches, could allow for a pragmatic scalability assessment, once incorporated into such tools. However, one should keep in mind the challenges involved, which have not yet allowed even the definition of common set of metrics or even common ground truth datasets. Such challenges include the interdependencies among the different tasks involved (e.g. a poor segmentation inevitably lowers annotation performance the same way as plain keyword-based retrieval fails to exploit semantics), and the different goals each application targets that impose different requirements in terms of precision, granularity, etc.

One, could claim here as a counter example the popularity of tools like Flickr, Picassa and YouTube. However, the reader should note among other things that:

- these tools support annotation in terms of either tags/keywords or free-text, raising interoperability issues;
- the granularity level of the annotations is user dependent, as well as the intended semantics, as there is no consensus on the terms used (an image depicting cliffs tagged as “rock” and an image of a live concert tagged as “rock”);
- annotations refer to the entire content and are rather coarse, which for personal content organisation might be sufficient, but for applications such as broadcasting/filtering on constraint environments (e.g. mobile phones), summarisation (e.g. of video for educational purposes), etc., this is not acceptable;
- annotations have to be controlled and detailed;
- already initiatives to social, collaborative-based approaches towards content interpretation are being reported, which clearly shows that the need to automate to some degree the extraction of content annotations is necessary.

Consequently, although the popularity of such tools provides significant hints with respect to user practical requirements, perception of content annotation and sharing, and social semantics acquisition and elicitation prove very interesting research challenges, such tools are lacking in aspects that would render them enabling factors of semantic-enabled content applications.

6.10 Summary

In this chapter we have investigated some scalability issues with respect to the various annotation tools. To sum up our findings, Beagle^{++}, GATE and KIM are designed to be scalable and to be applicable in real world scenarios. Extensive experiments - both on the tools themselves and with their use in industrial settings - have shown that they can cope reasonably well with large volumes of data and real-life ontologies without sacrificing performance. There are, however, some limitations: GATE applications may be quite slow depending on factors such as the design of the JAPE rules and combination of processing components, and the texts themselves (some kinds of text are notoriously problematic for GATE's sentence splitter, for example). MnM was never designed to be used in a large-scale environment and therefore suffers from a number of problems (some of which are indeed insurmountable without modifying the actual tool) when applied to large ontologies. Similarly, OntoMat was never designed for scalability and can suffer problems when used in such an environment. In general, any system that relies on learning will have problems when applied to a new ontology, as it will require a large amount of manually annotated training data which is time-consuming to produce. Multimedia annotation tools are perhaps the most problematic with respect to scalability, since they currently do not allow for automation, and yet manual annotation clearly remains infeasible on a large scale: research needs to be directed precisely at these problems.

One aspect of scalability that showed particularly strongly in the case of Magpie was the relationship between the desire to scale up as opposed to the real time responsiveness of the application to the user's interaction. Thus pragmatically, scale is only an added value of a particular annotation tool as far as the user is willing to wait for the outputs: In the case of GATE, this is easily in the extent of several hours or even days, since the output is a marked-up collection of documents. In the case of Magpie, the time limits are in the range of seconds – otherwise, the user is less likely to achieve the main objective of this tool, the navigation on the Web using semantic relationships and links.

Chapter 7

Recommendations and conclusions

In this deliverable, we have introduced a set of different annotation tools and investigated various aspects, such as their aims, usability, scalability and performance. Our aim is not to try to identify which tool is the best, but rather to examine the features a user should look for when deciding on a tool to use, and to highlight some issues which should be factors in that user's choice.

It is clear from our research that the most important thing to consider is the task to which the tool will be put and the situation in which it will be used. For this, we must first examine two main factors: the type of user, and the scope of tasks on which the tool will be put to use. Once these have been identified, we can turn our attention to more details: for example, if the tool is to be used solely by annotators, then a high degree of usability straight from the box may be important; if the tool is to be used only for a single, standalone kind of task, interoperability may be less of an issue than if the tool is to be used within a larger suite of tools, and so on.

7.1 Analysing the requirements for annotation tools

7.1.1 Users

According to [RJH05], there are 3 groups of annotation tool users:

- Annotators: those who just want to annotate data without worrying about representation, design, or how the tool works.
- Annotation Consumers: those who want to use annotated data for various purposes, and who need to be able to visualise, search and query annotated data and annotations.
- Developers: there are two main types:

- Corpus developers: corpus designers who may wish to add new annotation schema, map existing data to a new structure, etc.
- System developers: programmers who may wish to alter or extend the functionality of the tool, and who therefore need to understand how it works, add new components etc.

In reality these groups may not be so well defined, for example, corpus developers are very often the same people as the annotators. However we should be aware of the differing abilities of these groups of users. We would not necessarily expect annotators to have linguistic skills, let alone computer skills, although they will probably have domain knowledge, whereas system developers can be expected to have computer skills but not necessarily domain knowledge or linguistic skills. Corpus developers will generally have linguistic skills and probably domain knowledge, while annotation consumers will probably have domain knowledge but not necessarily linguistic or computing skills.

With respect to the tools analysed, all of them should be usable by both annotators and annotation consumers, while only GATE is really suitable for use by corpus and system developers. All the multimedia tools are suitable to be used by annotators (although M-Ontomat-Annotizer presupposes that the annotator has the relevant skills for multimedia analysis), and are also suitable to be used by developers for enhancing them with further functionalities.

7.1.2 Flexibility and usability

One of the main problems with designing annotation tools (common to many other software tools) is the tradeoff between generality and specificity. This is often less of a problem in industry than academia, because in industry, tools are generally designed with a single specific purpose in mind. Research tools, however, often need to be more flexible if they are to be taken up by other people and used for more than one task or application. Even though annotation may appear on the surface to be a quite straightforward task with a clear objective, both the uses and users of annotation tools may differ widely, and unless the tool is designed for a specific clear purpose with a particular kind of user in mind, there will almost always be dissatisfaction somewhere. One of the most important criteria is therefore that the annotation tool should be flexible and easily adaptable and/or extendable to the user's needs. This could range from something as simple as being able to change the colour of the annotation, through to enabling a whole new kind of visual representation or even modality.

[RJH05] also distinguish their requirements in terms of *functionality* and *usability*, whereby functionality concerns "the presence or absence of functions for a given task", i.e. "the relation tool-task", and usability concerns "the relation tool-user". While this distinction is a useful one, it does not cater at all for requirements such as performance. However, the main focus of their work is to characterise the evaluation of annotation tools in terms of annotation problems, such as portability, interoperability etc.

In Chapter 2, we investigated various aspects of non-performance-related issues such as general usability, accessibility, interoperability and so on. We cannot draw any hard and fast rules from this assessment: clearly it depends quite precisely on the user's requirements as to which tool best fulfils their needs according to this aspect. For example, GATE is probably the most fully-featured tool in terms of accessibility, allowing the user to have control over fonts, colours, text size. However, its graphics are quirky and unclear, and actions are almost entirely mouse-driven, forcing users to perform nearly all functions using a mouse rather than keyboard or alternative input device. Users may also have very specific requirements such as being able to use the tool on Linux, in which case the KIM plugin we tested would not be suitable, nor would the versions of MnM and OntoMat for Mac users. Some problems can of course be overcome if time and effort is available to be invested: for example it might be possible to adapt a tool to use a different ontology format (Magpie and MnM in the versions we tested were not compatible with OWL, for example). Concerning the range of tasks expected of the tool, GATE, KIM and Magpie have a much wider range than Ontomat and MnM. The multimedia tools examined either serve different purposes (M-Ontomat-Annotizer is mostly analysis oriented, while PhotoStuff, Aktive Media and Ontolog address descriptive annotation), or handle different types of content, making it difficult to form any general conclusions on design principles, especially since it is primarily the intended application context that determines them.

7.2 Performance

The influence of domain dependence on the effectiveness of NLP tools such as IE systems is an issue that is all too frequently overlooked. IE systems mostly extract fixed information from documents in a particular language and domain. For the technology to be suitable for real-world applications, IE systems need to be easily customisable to new domains [KSP99]. Due in no small part to the MUC competitions (e.g. [Sun95, Sun98]), work on IE however, has largely focused on narrow subdomains. For example. MUC 3 and MUC 4 focused on newswires about terrorist attacks, while MUC 7 was concerned with reports on air vehicle launches. Some work has been carried out on adapting existing systems to new domains, but there have been few advances in tackling the problem of making a single system robust enough to deal with different domains. The adaptation of existing systems to new domains is hindered by both ontology and rule bottlenecks. A substantial amount of knowledge is needed, and its acquisition and application are non-trivial tasks. For IE systems, the complexity of the domain may be particularly influential [Bag98].

An independent, though related, issue concerns the adaptation of existing systems to different text genres. By this we mean not just changes in domain, but different media (e.g. email, spoken text, written text, web pages), text type (e.g. reports, letters, books), and structure (e.g. layout). The genre of a text may be influenced by a number of factors, such as author, intended audience and degree of formality. For example, less formal texts may not follow standard capitalisation, punctuation or even spelling formats, all of which can be problematic for the intricate mechanisms of IE systems.

The problem of domain dependence is also related to the issues of algorithmic reuse. It seems to be particularly true of IE systems that they are tailor-made for specific domains and applications, with the result that not only are they hard to adapt for new tasks, but that it is difficult to extricate potentially reusable components or sub-components which are buried deep in the architecture. For example, there may not be a distinction between foreground information, which is dependent on the domain and application, and background information, which can be reused as it stands; and consequently, between the tools needed to access and manipulate these two types of information.

The issue of domain dependence is largely tackled by the performance evaluation (Section 5). As discussed there, it was difficult to compare different tools on an equal footing because they were almost all designed with different goals in mind and therefore work best on quite specific kinds of corpora, apart from GATE and – to some extent KIM – which are designed specifically to be general, and which require modification in order to be fine-tuned to any particular domain. Leaving this issue aside, according to the evaluations performed by us and by others previously, we see that in general, GATE has the best performance, narrowly followed by KIM, while MnM, Magpie and OntoMat only really have good performance on a very specific kind of text, and even then still fall some way behind GATE and KIM. In the case of Magpie, the precision on annotating text that is within the domain of the loaded lexicon is far superior to any other tool; however, the recall tends to decline rapidly if the text moves beyond the domain of the lexicon. This is a well-known feature of ontology-driven annotation, often presented as ontology brittleness.

With respect to evaluating performance, it is important to note, as we have seen in Chapters 3 and 4, that there is a variety of ways in which to measure performance, and the choice of measure may be as dependent on the tool itself as on the aim of the evaluation. For example, although MnM annotates texts with instances from an ontology (as do the other textual annotation tools) it is almost impossible to run – and therefore to evaluate it – on an ontology of any significant size (with more than a very small handful of concepts). As the results from the performance experiments show, however, we can see quite different results on the same tools depending on which kind of measure we use. For example, we saw with GATE and KIM that using Augmented Precision and Recall, as defined and discussed in Section 3.2 rather than standard Precision and Recall gives us a much better idea of the strengths and weaknesses of the tools.

7.3 Tasks

While GATE, Magpie, OntoMat, KIM, and to some extent MnM are all primarily designed to perform the same kind of annotation task, Beagle++ is slightly different in that it is designed primarily as a search engine tool, which happens to perform annotation in order to carry out this task. Clearly, one would not use Beagle++ if one were just aiming to mark up some textual data without the ultimate goal of desktop search. Similarly, one would be unlikely to use GATE to annotate data prior to desktop search. The multimedia annotation tools, on the other hand, are all designed for roughly the same task of annotating multimedia material, and in this respect are more interchangeable with respect to the actual task, though not necessarily with respect to the way in which the task needs to be performed.

7.4 Interoperability

Interoperability of annotation tools covers quite a wide range of separate issues, covering factors such as data format, ontology format, annotation format, and so on. Most of the annotation tools we tested are usable with a range of browsers (where applicable), operating systems and text formats, although the ontology format accepted tends to be more restricted. In general, the input formats are quite flexible but the output formats are not (for example, most tools do not provide a means of converting their output into different formats).

The benchmarking of OWL interoperability of semantic web tools is covered in a separate deliverable [GCDPG07], so we only summarise the results in this deliverable. Only GATE and KIM out of the textual annotation tools could be evaluated under this framework since the other annotation tools do not use OWL ontologies or have other quirks which make them impossible to be evaluated in this way. Furthermore, both GATE and KIM share the same ontology API so their results on the interoperability benchmarking would be identical (only GATE was actually evaluated). GATE itself scored quite highly on the interoperability benchmarking, interacting well with tools such as Protege, KAON, JENA and SWI-Prolog, although it fell down in a couple of places, such as not creating all the instances correctly.

It is clear from this that most textual annotation tools are not really designed with interoperability in mind, especially as far as the Semantic Web is concerned. Most current annotation tools are based on legacy information extraction tools which do not take ontologies into account, and have in fact been adapted to perform ontology-based annotation. For this reason, most of them were not developed with interoperability concerns in mind, and the effort to make this possible can be quite substantial. This was the case with the development of GATE, for example, until it became necessary to adapt GATE to deal with OWL ontologies during the course of its development, and this required a substantial amount of work and some radical redesign of the ontology API.

With respect to multimedia annotation tools, interoperability is more challenging, as it requires first to overcome the individual interoperability issues that involve the representation of the different types of knowledge involved, the formal modelling of aspects particular to multimedia (such as localisation information), and the extension of the tools so as to allow the editing of existing annotations, a functionality that would give a measurable degree of the interoperability of the different implementations. As long as the tools cover only manual annotation, any evaluation reduces to comparison of the respective annotators and the domain ontologies used, aspects that do not really provide anything in terms of the tools comparison. For this reason, given the current state of the art, a user who needs to perform ontology-based annotation of multimedia content should mostly rely on the dimensions discussed in Chapter 1. However, as soon as semi- and fully automatic approaches for the extraction of multimedia semantics reach robust performances, which would enable better integration, users will have more criteria on which to base their selection, e.g. precision and recall metrics, user satisfaction with respect to the end application that uses these annotations, etc.

7.5 Scalability

As we have discussed in this deliverable, most of the tools examined here were designed originally as small scale research prototypes rather than tools for large scale annotation. Of these, Magpie, GATE and KIM are the only tools which have really been designed for general purpose use and/or have been adapted for dealing with large scale real world applications. Both GATE and KIM perform reasonably well when used with large ontologies and data sets, although GATE can be quite slow with large datasets or with complex applications (for example, there are sometimes problems when using massive gazetteer lists). For multimedia annotation tools, scalability is really an issue at the forefront of research, since these tools mainly deal with manual annotation only, which makes it impossible to work with huge volumes of data.

7.6 The future of annotation and the semantic web

Both textual and multimedia annotation have proved their success in a number of real world applications, such as Del.ici.ous, Flickr, digital libraries such as Perseus[1], Garlik[2] (which mines data about consumers present in various sources including the web), Fizzback[3] (which provides real-time customer feedback from SMS and email feeds) and so on. However, there are several reasons why semantic annotation is not more widespread. First, it is time-consuming and complex to produce annotations in open domains. Second, there are not enough DIY cases: Flickr and Del.ici.ous have tapped into obvious needs, and Innovantage hopes to do the same, but in many cases we do not have enough folk for a folksononmy. For example, broadcast archives are a Catch-22 situation: people are not

[1]http://www.perseus.tufts.edu/
[2]http://www.garlik.com
[3]http://www.fizzback.com

aware that they need them until they use them, and they can't use them until widespread need generates either human effort or funding. Automatic methods for annotation, on the other hand, have made huge advances in recent years to a usable level, but the output still tends to be incomplete or innacurate, especially in open domains. One solution is therefore to combine automatic methods with human annotation at the lowest possible cost, and preferably done by non-experts. While we have plenty of algorithms, data structures and evaluation protocols emerging, we still currently lack a clear statement of how to specify and implement new annotation tasks, especially those oriented towards non-HLT experts. What is therefore needed is a methodology covering how to:

- decide if annotation is applicable to a problem;
- define the problem with reference to a set of examples;
- identify similarities with other problems and thus to estimate likely performance levels;
- design the annotation workflow, including automatic assistance;
- measure success.

These tasks are beyond the scope of the current work in this project; however, they are currently being pursued in a variety of recently started projects such as NeOn[4] and MUSING[5], by researchers at Harvard Medical School, and by some commercial users.

[4]http://www.neon-project.org
[5]http://www.musing-project.eu

Acknowledgements

The authors would like to thank the university library Mannheim, in particular Magnus Pfeffer for valuable comments and discussions on the work reported in Chapter 4. We are grateful to Collexis for granting a scientific license for the use of their indexing software and to SyynX GmbH (in particular Christian Herzog and Mario Diwersy) for providing technical and conceptual support on the use of the software. We further thank Elsevier for providing us with full text articles from their leading economics journals and the ZBW in Kiel for providing us with an electronic version of the STW Thesaurus.

Some of the experiments with GATE and the Machine Learning algorithms evaluated were developed within the context of the EU SEKT project, as were some of the experiments on the scalability of KIM and the development of the the KIM Cluster Architecture.

Related deliverables

A number of deliverables are related to this one:

- **D1.2.2.1.2 Benchmarking the interoperability of ontology development tools using OWL as interchange language** describes benchmarking the interoperability aspect of various tools including some of the annotation tools described here.
- **D1.4.2v2 Success stories and best practices** included some discussion of scaling annotation tools to the real world.
- **D2.1.1 State of the Art on the scalability of ontology-based technology** included some preliminary discussions on evaluating ontology-based annotation tools.
- **D2.1.4 Specification of a Methodology, general criteria and benchmark suites for benchmarking ontology tools** described some general methods and tools for benchmarking annotation tools.
- **D2.3.6 Prototypes of language dependent tools for evaluation** described some tools and methods for evaluating annotation tools.
- **D3.3.3 Prototype of advanced learning platform (ASPL)** describes the ASPL which is based largely on Magpie, one of the tools we investigate in this deliverable.

Bibliography

[Bag98] Amit Bagga. Analysing the complexity of a domain with respect to an Information Extraction task. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*. `http://www.itl.nist.gov/iaui/894.02/-related_projects/muc/index.html`, 1998.

[BKBK05] A. Bernstein, E. Kaufmann, C. Burki, and M. Klein. How similar is it? towards personalized similarity measures in ontologies. In *7. Int. Tagung Wirtschaftsinformatik*. Bamberg, Germany, 2005.

[BKKB05] A. Bernstein, E. Kaufmann, C. Kiefer, and C. Burki. Simpack: A generic java library for similarity measures in ontologies. Technical report, University of Zurich, Department of Informatics, 2005.

[BPS$^+$05] S. Bloehdorn, K. Petridis, C. Saathoff, N. Simou, V. Tzouvaras, Y. Avrithis, S. Handschuh, I. Kompatsiaris, and S. and M.G. Strintzis. Semantic annotation of images and videos for multimedia analysis. In *ESWC*, pages 592–607, 2005.

[CGG$^+$05] Paul Alexandru Chirita, Rita Gavriloaie, Stefania Ghita, Wolfgang Nejdl, and Raluca Paiu. Activity based metadata for semantic desktop search. In *In Proceedings of the 2nd European Semantic Web Conference*, Heraklion, Greece, May 2005.

[CHS04] P. Cimiano, S. Handschuh, and S. Staab. Towards the Self-Annotating Web. In *Proceedings of WWW'04*, 2004.

[CLC06] A. Chakravarthy, V. Lanfranchi, and F. Ciravegna. Cross-media document annotation and enrichment. In *In 1st Semantic Authoring and Annotation Workshop (SAAW), 5th International Semantic Web Conference*, Athens, GA, USA, 2006.

[CM05] Courtney Corley and Rada Mihalcea. Measuring the semantic similarity of texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 13–18. ACL, 2005.

[CMBT02] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, 2002.

[CMT00] H. Cunningham, D. Maynard, and V. Tablan. JAPE: a Java Annotation Patterns Engine (Second Edition). Research Memorandum CS–00–10, Department of Computer Science, University of Sheffield, November 2000.

[CST00] N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[DDM04] J. Domingue, M. Dzbor, and E. Motta. Magpie: Supporting Browsing and Navigation on the Semantic Web. In N. Nunes and C. Rich, editors, *Proceedings ACM Conference on Intelligent User Interfaces (IUI)*, pages 191–197, 2004.

[DHL03] M. Doerr, J. Hunter, and C. Lagoze. Towards a core ontology for information integration. *J. Digit. Inf.*, 4(1), 2003.

[DKS04] O. Dekel, J. Keshet, and Y. Singer. Large Margin Hierarchical Classification. In *Proceedings of the 21st International Conference on Machine Learning (ICML-2004)*, Canada, 2004.

[DS05] M. Dzbor and A. Stutt. Prototype of advanced learning platform (ASPL-v1). Technical Report D3.3.3, KnowledgeWeb Deliverable, 2005.

[DTCP05] M. Dowman, V. Tablan, H. Cunningham, and B. Popov. Web-assisted annotation, semantic indexing and search of television and radio news. In *Proceedings of the 14th International World Wide Web Conference*, Chiba, Japan, 2005. `http://gate.ac.uk/sale/www05/web-assisted-annotation.pdf`.

[Euz07] Jérôme Euzenat. Semantic precision and recall for ontology alignment evaluation. In *IJCAI*, pages 348–353, 2007.

[GCDPG07] R. García-Castro, S. David, and J. Prieto-González. D1.2.2.1.2 benchmarking the interoperability of ontology development tools using owl as interchange language. Technical report, Knowledge Web, September 2007.

[GCMW+05] R. Garcia-Castro, D. Maynard, H. Wache, D. Foxvog, and R. Gonzalez Cabero. Specification of a methodology, general criteria and benchmark suites for benchmarking ontology tools. Technical Report D2.1.4, KnowledgeWeb Deliverable, 2005.

[Gri95] R. Grishman. TIPSTER Architecture Design Document Version 2.0 (Tinman Architecture). Technical report, Department of Computer Science, New York University, 1995. `http://www.cs.nyu.edu/tipster`.

[Heg05] Jon Heggland. Ontolog: Flexible management of semantic video content annotations. PhD Thesis, Norwegian University of Science and Technology (NTNU), Department of Computer and Information Science (IDI), 2005.

[HS98a] U. Hahn and K. Schnattinger. Towards text knowledge engineering. In *Proc. of 15th National Conference on Artificial Intelligence (AAAI-98)*, pages 524–531, Menlo Park, CA, 1998. MIT Press.

[HS98b] Udo Hahn and Klemens Schnattinger. Towards text knowledge engineering. In *AAAI/IAAI*, pages 524–531, 1998.

[HSC02] S. Handschuh, S. Staab, and F. Ciravegna. S-CREAM — Semi-automatic CREAtion of Metadata. In *13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)*, pages 358–372, Siguenza, Spain, 2002.

[HSM01] S. Handschuh, S. Staab, and A. Maedche. CREAM – creating relational metadata with a component-based, ontology-driven framework. In *Proceedings of K-CAP 2001*, Victoria, BC, Canada, 2001.

[Hun01] J. Hunter. Adding multimedia to the semantic web: Building an mpeg-7 ontology. In *SWWS*, pages 261–283, 2001.

[HWGS$^+$05] C. Halaschek-Wiener, J. Golbeck, A. Schain, M. Grove, B. Parsia, and J. Hendler. Photostuff Ů an image annotation tool for the semantic web. In *In Y. Gil, E. Motta, V.R. Benjamins, M.A. Musen (Ed.), Poster Proceedings of the 4th International Semantic Web Conference*, Galway, Ireland, 2005.

[JC97] Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Natural Language Engineering - Proceedings of International Conference Research on Computational Linguistics (ROCLING X)*, pages 81–111. Tapei, Taiwan, 1997.

[KPO$^+$04] A. Kiryakov, B. Popov, D. Ognyanoff, D. Manov, A. Kirilov, and M. Goranov. Semantic annotation, indexing and retrieval. *Journal of Web Semantics, ISWC 2003 Special Issue*, 1(2):671–680, 2004.

[KSP99] V. Karkaletsis, C.D. Spyropoulos, and G. Petasis. Named Entity Recognition from Greek texts: the GIE Project. In S.Tzafestas, editor, *Advances in Intelligent Systems: Concepts, Tools and Applications*, pages 131–142. Kluwer Academic Publishers, 1999.

[LBC05a] Y. Li, K. Bontcheva, and H. Cunningham. SVM Based Learning System For Information Extraction. In M. Niranjan J. Winkler and N. Lawerence, editors, *Deterministic and Statistical Methods in Machine Learning*, LNAI 3635, pages 319–339. Springer Verlag, 2005.

[LBC05b] Y. Li, K. Bontcheva, and H. Cunningham. Using Uneven Margins SVM and Perceptron for Information Extraction. In *Proceedings of Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, 2005.

[LBC06] Y. Li, K. Bontcheva, and H. Cunningham. Perceptron-like learning for ontology based information extraction. Technical report, University of Sheffield, Sheffield, UK, 2006.

[LBC07] Y. Li, K. Bontcheva, and H. Cunningham. Hierarchical, Perceptron-like Learning for Ontology Based Information Extraction. In *16th International World Wide Web Conference (WWW2007)*, pages 777–786, 2007.

[LBM02] Yuhua Li, Zuhair Bandar, and David McLean. Measuring semantic similarity between words using lexical knowledge and neural networks. In *IDEAL 2002: Third International Conference*, page 111. Springer Berlin/Heidelberg, 2002.

[LC98] C. Leacock and M. Chodorow. Combining local context and wordnet similarity for vord sense identification. In C. Fellbaum, editor, *WordNet: An Electronic Lexical Database*, chapter 11, pages 265–283. MIT Press, 1998.

[Lin98] Dekang Lin. An information-theoretic definition of similarity. In *Proc. 15th International Conf. on Machine Learning*, pages 296–304. Morgan Kaufmann, San Francisco, CA, 1998.

[Mar02] J.M. Martinez. Mpeg-7: Overview of description tools. *IEEE Multimedia*, 9(3):83–93, 2002.

[May05a] D. Maynard. Benchmarking ontology-based annotation tools for the semantic web. In *UK e-Science Programme All Hands Meeting (AHM2005) Workshop "Text Mining, e-Research and Grid-enabled Language Technology"*, Nottingham, UK, 2005.

[May05b] Diana Maynard. Benchmarking ontology-based annotation tools for the semantic web. In *UK e-Science Programme All Hands Meeting (AHM2005) Workshop "Text Mining, e-Research and Grid-enabled Language Technology"*. Nottingham, UK, 2005.

[MP05] D. Manov and B. Popov. D2.6.1 Massive Automatic Annotation. Technical report, SEKT EU Project Deliverable, 2005.

[MPL06a] D. Maynard, W. Peters, and Y. Li. Metrics for evaluation of ontology-based information extraction. In *WWW 2006 Workshop on "Evaluation of Ontologies for the Web" (EON)*, Edinburgh, Scotland, 2006.

[MPL06b] D. Maynard, W. Peters, and Y. Li. Metrics for evaluation of ontology-based information extraction. In *WWW 2006 Workshop on "Evaluation of Ontologies for the Web" (EON)*, Edinburgh, Scotland, 2006.

[MPSC06] D. Maynard, W. Peters, M. Sabou, and P. Cimiano. Prototypes of language dependent tools for evaluation. Technical Report D2.3.6, KnowledgeWeb Deliverable, 2006.

[MSY$^+$07] D. Maynard, H. Saggion, M. Yankova, K. Bontcheva, and W. Peters. Natural Language Technology for Information Integration in Business Intelligence. In *10th International Conference on Business Information Systems (BIS-07)*, Poznan, Poland, 2007.

[MTC$^+$02] D. Maynard, V. Tablan, H. Cunningham, C. Ursu, H. Saggion, K. Bontcheva, and Y. Wilks. Architectural Elements of Language Engineering Robustness. *Journal of Natural Language Engineering – Special Issue on Robust Methods in Analysis of Natural Language Data*, 8(2/3):257–274, 2002.

[MTU$^+$01] D. Maynard, V. Tablan, C. Ursu, H. Cunningham, and Y. Wilks. Named Entity Recognition from Diverse Text Types. In *Recent Advances in Natural Language Processing 2001 Conference*, pages 257–274, Tzigov Chark, Bulgaria, 2001.

[MVVD$^+$02] E. Motta, M. Vargas-Vera, J. Domingue, M. Lanzoni, A. Stutt, and F. Ciravegna. MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup. In *13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)*, pages 379–391, Siguenza, Spain, 2002.

[MYKK05] D. Maynard, M. Yankova, A. Kourakis, and A. Kokossis. Ontology-based information extraction for market monitoring and technology watch. In *ESWC Workshop "End User Apects of the Semantic Web")*, Heraklion, Crete, 2005.

[PABC05] W. Peters, N. Aswani, K. Bontcheva, and H. Cunningham. Quantitative Evaluation Tools and Corpora v1. Technical report, SEKT project deliverable D2.5.1, 2005.

[PKK$^+$04] B. Popov, A. Kiryakov, A. Kirilov, D. Manov, D. Ognyanoff, and M. Goranov. KIM – Semantic Annotation Platform. *Natural Language Engineering*, 2004.

[PLM$^+$07] J. Pan, L. Lancieri, D. Maynard, F. Gandon, R. Cuel, and A. Leger. Success stories and best practices. Technical Report D1.4.2v2, KnowledgeWeb Deliverable, 2007.

[Res95] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, 1995.

[Res99] Philip Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130, 1999.

[RJH05] D. Reidsma, N. Jovanovic, and D. Hofs. Designing annotation tools based on properties of annotation problems. In *Measuring Behavior 2005, 5th International Conference on Methods and Techniques in Behavioral Research*, Wageningen, The Netherlands, 2005.

[SCB$^+$03] H. Saggion, H. Cunningham, K. Bontcheva, D. Maynard, O. Hamza, and Y. Wilks. Multimedia Indexing through Multisource and Multilingual Information Extraction; the MUMIS project. *Data and Knowledge Engineering*, 48:247–264, 2003.

[SGJ01] S. Santini, A. Gupta, and R. Jain. Emergent semantics through interaction in image databases. *IEEE Trans. Knowl. Data Eng.*, 13(3):337–351, 2001.

[SP05] P. Sazedj and H. Sofia Pinto. Time to evaluate: Targeting annotation tools. In *Proceedings of ISWC 2005 SemAnnot Workshop*, Galway, Ireland, 2005.

[SPA$^+$06] C. Saathoff, K. Petridis, D. Anastasopoulos, N. Timmermann, I. Kompatsiaris, and S. Staab. M-ontomat-annotizer: Linking ontologies with multimedia low-level features for automatic image annotation. In *In Proceedings of the 3rd European Semantic Web Conference, Demos and Posters*, Budva, Montenegro, 2006.

[Sun95] Beth Sundheim, editor. *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, Columbia, MD, 1995. ARPA, Morgan Kaufmann.

[Sun98] Beth Sundheim, editor. *Proceedings of the Seventh Message Understanding Conference (MUC-7)*. ARPA, Morgan Kaufmann, 1998.

[SVH04] Nuno Seco, Tony Veale, and Jer Hayes. An intrinsic information content metric for semantic similarity in wordnet. In *Proceedings of the 16th European Conference on Artificial Intelligence*, pages 1089–1090. Valencia, Spain, 2004.

[TPC07] C. Tsinaraki, P. Polydoros, and S. Christodoulakis. Interoperability support between mpeg-7/21 and owl in ds-mirf. *IEEE Trans. Knowl. Data Eng.*, 19(2):219–232, 2007.

[vONH04] J. van Ossenbruggen, F. Nack, and L. Hardman. That obscure object of desire: Multimedia metadata on the web, part 1. *IEEE MultiMedia*, 11(4):38–48, 2004.

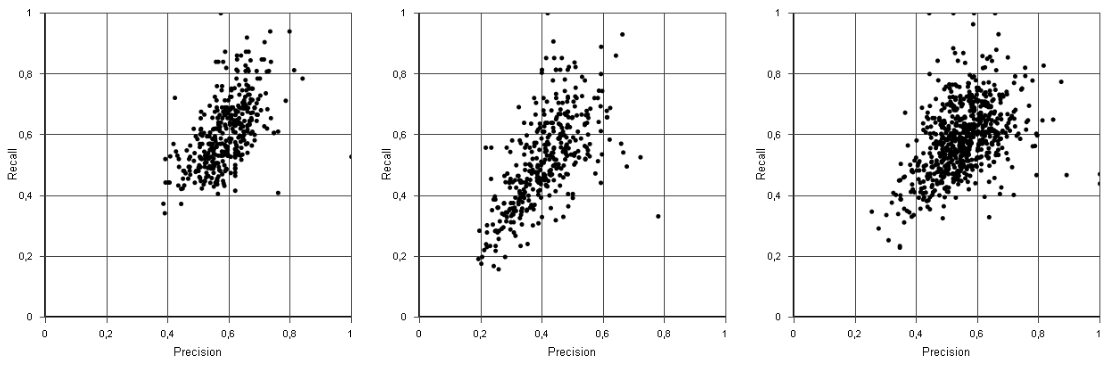
[WP94] Zhibiao Wu and Martha Palmer. Verb semantics and lexical selection. In *32nd. Annual Meeting of the Association for Computational Linguistics*, pages 133 –138, New Mexico State University, Las Cruces, New Mexico, 1994.

Appendix

In this Appendix we show the results for the experiments reported in Chapter 4 looking at different similarity measures.

Document Base	# Documents	# Keywords	$\sum$Similarity	Recall
STW (broken)/Elsevier	391	1658	1005	0.61
STW/Elsevier	391	1646	849	0.52
MeSH/Medline	706	8143	4754	0.58
		# Concepts	$\sum$Similarity	Precision
STW (broken)/Elsevier	391	2980	2980	0.59
STW/Elsevier	391	3377	1371	0.41
MeSH/Medline	706	10041	4578	0.55

Table 7.1: Generalised Precision and Recall (Leacock Chodorow)

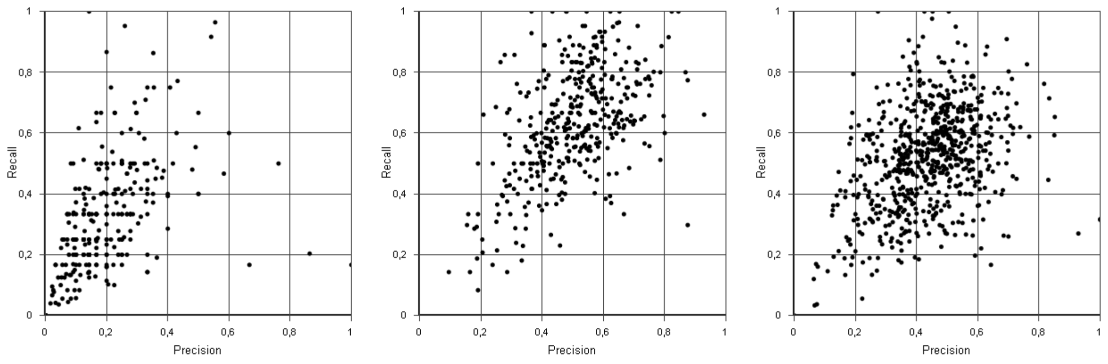


(a) STW Leacock Chodorow (b) STW (broken) Leacock Chodorow (c) MeSH Leacock Chodorow

Figure 7.1: Generalised Precision and Recall (Leacock Chodorow)

Document Base	# Documents	# Keywords	$\sum$Similarity	Recall
STW (broken)/Elsevier	391	1658	388	0.23
STW/Elsevier	391	1646	1045	0.63
MeSH/Medline	706	8143	4380	0.54
		# Concepts	$\sum$Similarity	Precision
STW (broken)/Elsevier	391	2980	420	0.14
STW/Elsevier	391	3377	1688	0.5
MeSH/Medline	706	10041	4200	0.52

Table 7.2: Generalised Precision and Recall (Wu Palmer)

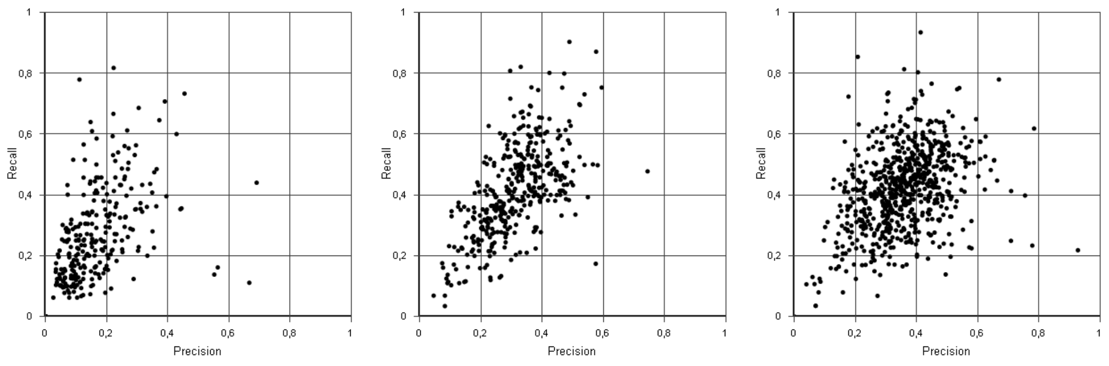


(a) STW Wu Palmer (b) STW (broken) Wu Palmer (c) MeSH Wu Palmer

Figure 7.2: Generalised Precision and Recall (Wu Palmer)

Document Base	# Documents	# Keywords	$\sum$Similarity	Recall
STW (broken)/Elsevier	391	1658	316	0.19
STW/Elsevier	391	1646	686	0.42
MeSH/Medline	706	8143	3406	0.44
		# Concepts	$\sum$Similarity	Precision
STW (broken)/Elsevier	391	2980	343	0.12
STW/Elsevier	391	3377	1049	0.31
MeSH/Medline	706	10041	3562	0.35

Table 7.3: Generalised Precision and Recall (Resnik)

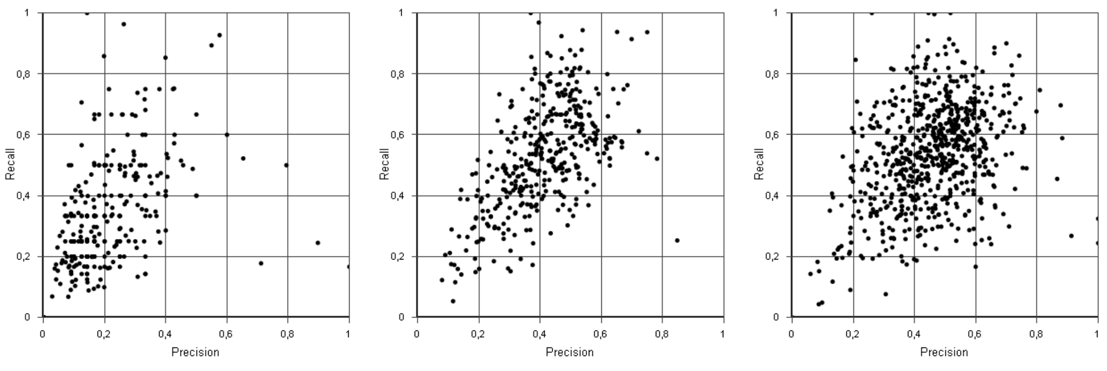


(a) STW Resnik (b) STW (broken) Resnik (c) MeSH Resnik

Figure 7.3: Generalised Precision and Recall (Resnik)

Document Base	# Documents	# Keywords	$\sum$Similarity	Recall
STW (broken)/Elsevier	391	1658	407	0.25
STW/Elsevier	391	1646	883	0.54
MeSH/Medline	706	8143	4347	0.53
		# Concepts	$\sum$Similarity	Precision
STW (broken)/Elsevier	391	2980	445	0.15
STW/Elsevier	391	3377	1380	0.41
MeSH/Medline	706	10041	4578	0.46

Table 7.4: Generalised Precision and Recall (Lin)

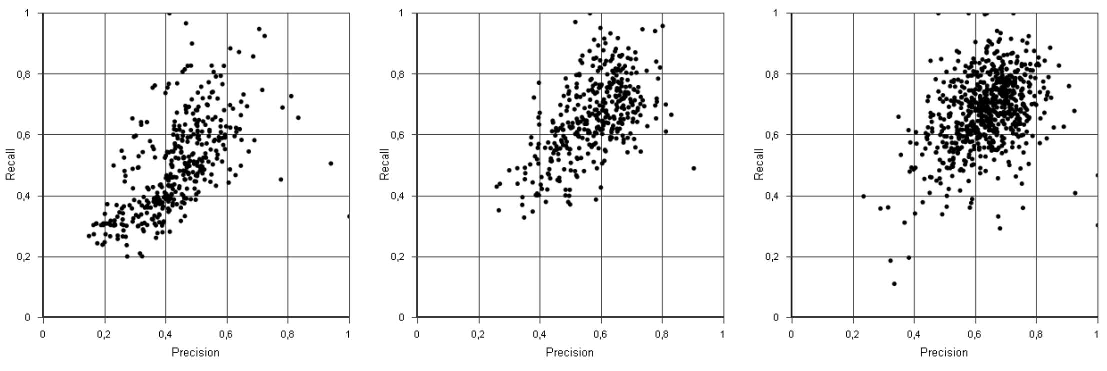


(a) STW Lin (b) STW (broken) Lin (c) MeSH Lin

Figure 7.4: Generalised Precision and Recall (Lin)

Document Base	# Documents	# Keywords	$\sum$Similarity	Recall
STW (broken)/Elsevier	391	1658	826	0.5
STW/Elsevier	391	1646	1092	0.66
MeSH/Medline	706	8143	5490	0.67
		# Concepts	$\sum$Similarity	Precision
STW (broken)/Elsevier	391	2980	1279	0.43
STW/Elsevier	391	3377	1932	0.57
MeSH/Medline	706	10041	6371	0.63

Table 7.5: Generalised Precision and Recall (Jiang Conrath)



(a) STW Jiang Conrath (b) STW (broken) Jiang Conrath (c) MeSH Jiang Conrath

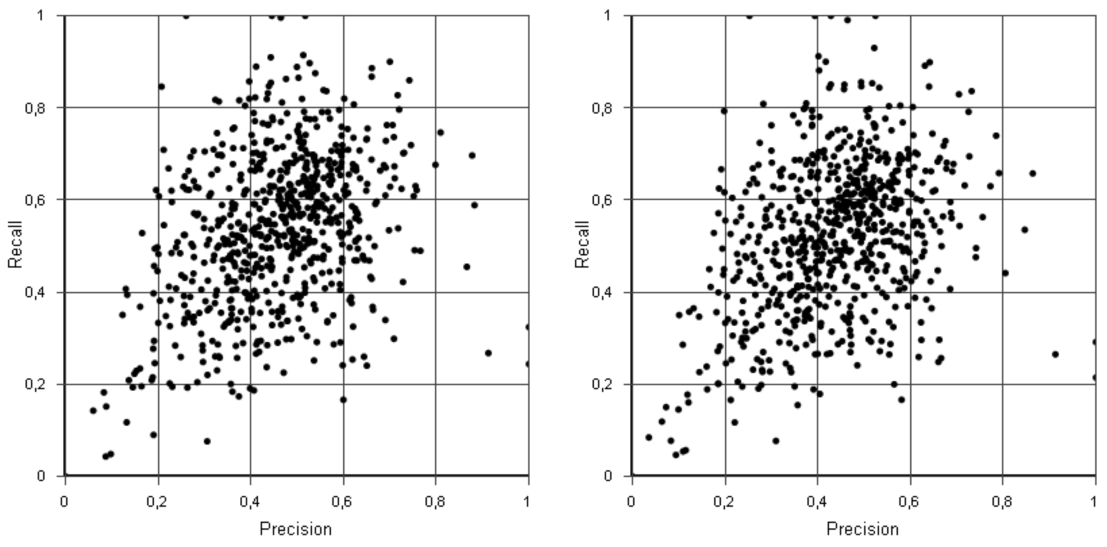
Figure 7.5: Generalised Precision and Recall (Jiang Conrath)

Document Base	# Documents	# Keywords	$\sum$Similarity	Recall
STW (broken)/Elsevier	391	1658	401	0.24
STW/Elsevier	391	1646	862	0.52
MeSH/Medline	706	8143	4197	0.52
		# Concepts	$\sum$Similarity	Precision
STW (broken)/Elsevier	391	2980	438	0.15
STW/Elsevier	391	3377	1331	0.39
MeSH/Medline	706	10041	4353	0.43

Table 7.6: Generalised Precision and Recall (Lin Intrinsic)

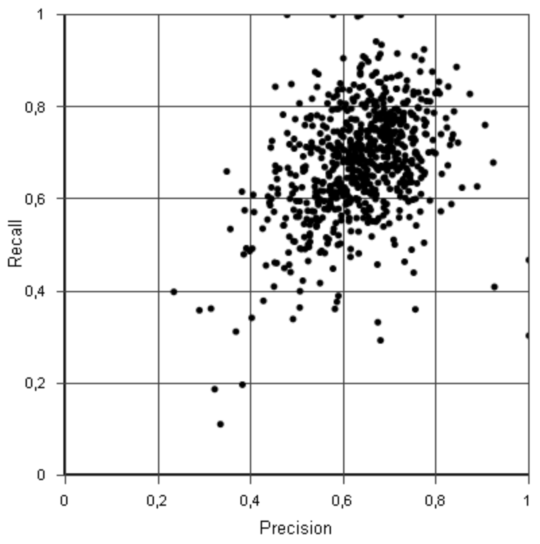
Document Base	# Documents	# Keywords	$\sum$Similarity	Recall
STW (broken)/Elsevier	391	1658	702	0.42
STW/Elsevier	391	1646	999	0.61
MeSH/Medline	706	8143	5081	0.62
		# Concepts	$\sum$Similarity	Precision
STW (broken)/Elsevier	391	2980	888	0.29
STW/Elsevier	391	3377	1621	0.48
MeSH/Medline	706	10041	5642	0.56

Table 7.7: Generalised Precision and Recall (Jiang Conrath Intrinsic)

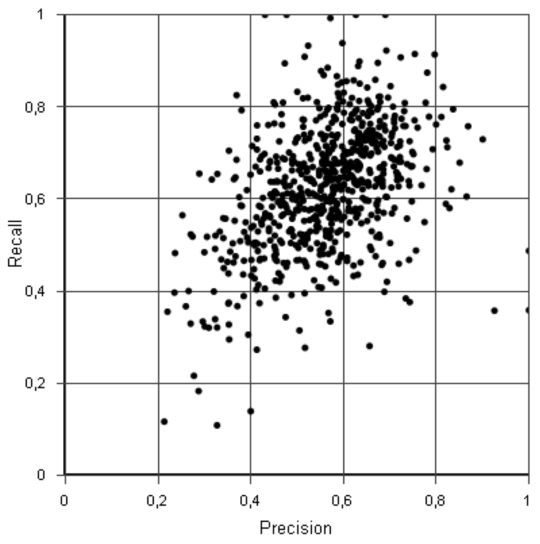


(a) MeSH Lin (b) MeSH Intrinsic Lin

Figure 7.6: Generalised Precision and Recall (MeSH, Lin, IC vs IIC)



(a) MeSH Jiang Conrath



(b) MeSH Intrinsic Jiang Conrath

Figure 7.7: Generalised Precision and Recall (MeSH, Jiang Conrath, IC vs IIC)