

Diploma Thesis

ConTag **A Tagging System Linking** **the Semantic Desktop with Web 2.0**

by
Benjamin Horak
346462

August 2006

SUPERVISORS: PROF. DR. ANDREAS DENGEL
DIPL.INFORM. LEO SAUERMANN

University of Kaiserslautern
Department of Computer Science

Benjamin Horak
Meisenweg 2
67663 Kaiserslautern

Kaiserslautern, den August 10, 2006

Erklärung

”Ich versichere hiermit, dass ich die vorliegende Diplomarbeit mit dem Thema **ConTag - A Tagging System linking the Semantic Desktop with Web 2.0** selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen, die anderen Werken dem Wortlaut oder dem Sinn nach entnommen wurden, habe ich durch die Angabe der Quelle, auch der benutzten Sekundärliteratur, als Entlehnung kenntlich gemacht.”

(Ort, Datum)

(Unterschrift)

Contents

1	Introduction	1
1.1	Semantic Desktop - Tagging - Web 2.0	1
1.2	Goals of this study	2
2	Motivation	5
2.1	Recent weaknesses in information management	5
2.2	Personal Information Management	7
2.3	Annotating resources	8
2.4	Collaborative and social aspects in annotating resources	11
2.5	A disastrous tagging example	12
2.6	Actual tagging weaknesses	13
2.7	Lessons learned	14
3	ConTag	15
3.1	Personal Information Model	16
3.2	About the nature of tags	17
3.3	Effective explanations	20
3.4	Linguistic aspects	21
3.5	The idea behind ConTag	21
3.6	Using Web 2.0 services	24
4	Use Cases	27
4.1	Use Case commonalities	27
4.2	Use Case 1: Retrieve existing instances	30
4.3	Use Case 2: Extract new instances	30
4.4	Use Case 3: Classify instances to extracted classes	31
4.5	Use Case 4: Create relations between instances	32

5	Design	33
5.1	General architecture	33
5.2	ConTag’s Tagging Process	37
5.3	Information Cloud	39
5.4	Generate ontology alignments	40
5.5	Classification in a Personal Information Model	41
5.5.1	Comparing topics and concepts	42
5.5.2	Creating the four correspondences	43
5.6	Displaying tag proposals	45
5.7	Alignment execution	47
5.8	Web 2.0	47
5.8.1	Web 2.0 services	47
5.8.2	Syndication in Web 2.0	50
6	Evaluation	51
6.1	Precision and Recall	51
6.2	Quality assurance in test environments	52
6.3	Test corpora	53
6.3.1	Reuters’ News Corpus	54
6.3.2	Index Site Corpus	54
6.3.3	Wikipedia Concept Corpus	55
6.3.4	Wikisource Historical Corpus	56
6.4	Topic extraction evaluation	56
6.5	Alignment generation evaluation	60
6.5.1	Service analysis	60
6.5.2	Precision analysis	70
7	Conclusion	73
7.1	Summary and Discussion	73
7.2	Outlook	75
A	Appendix	i
A.1	Existing tagging systems	i
A.2	Inspected Web 2.0 services	i
A.3	Test PIMO	ii
A.4	Test corpora’s contents	iv
A.5	Reuter’s News Corpus	iv
A.6	Index Site Corpus	iv
A.7	Wikipedia Concept Corpus	v
A.8	Wikisource Historical Corpus	v

A.9 Used software libraries	vi
List of Figures	vii
Bibliography	xiii

Acknowledgements

I would like to deeply thank my fiancée Denise Adrian for all her exercised patience and sensitivity, all the iced coffee and chocolate, during the time it took to write this diploma thesis.

I want to thank Leo Sauermann and Thomas Roth-Berghofer as well for all the wonderful, creative discussions and constructive guidelines about ConTag. I learned a lot during this time of development.

Thank you Martin and Tanja Roth for your proofreading of English spelling and grammar.

And also a heartfelt and deep "thank you" to all members of the Knowledge Management Department of the DFKI for all their useful advices and patience.

Chapter 1

Introduction

"He's not any kind of program, Sark. He's a User."
- Master Control Program in the movie "Tron", 1982 -

The reading of written documents is often accompanied by noting major topics mentioned in the document or categories being classified into. Using a Personal Computer enables users to manage their text annotations in machine readable formats. Supporting users to annotate documents efficiently may be useful in order to facilitate and support later on retrieval approaches.

Modern Personal Information Management systems use these opportunity to manage known documents (e.g. web pages) by indexing their existing annotations. Compared to common document retrieval approaches, such as browser bookmarks, the filesystem, or Web Search Engines, the usage of personal annotations humanizes the process of Information Retrieval.

Based on this idea, the following study describes a system called ConTag, designed to help users to annotate their documents.

1.1 Tagging resources in a Semantic Desktop environment using Web 2.0 services

ConTag is built upon three existing technologies in the World Wide Web. It is assumed and encouraged, that users work on a *Semantic Desktop* system in order to stand to benefit from a variety of functionalities to manage existing information, encountered topics and known personal concepts more efficiently on a Personal Computer [Sau03]. In these systems, users express their knowledge in a *Personal Information Model*.

Additionally it uses a combination of web technologies, often referenced as the *Web 2.0* [O'R05]. The Web 2.0 is a structure-centered view towards the WWW providing a variety of Web 2.0 services to aid users in processing digital information in different forms. Using these services enables ConTag to extract relevant topics out of written documents.

One Web 2.0 technique is in the focus. WWW based *tagging systems* [Bec06] let users extract topics from documents and other web resources, in order to note these topics as concise text annotations called tags. These tags describe contents and additional information about documents and other resources to provide a better retrieval and classification support.

As a result of these technological prerequisites, readers of this study should be familiar with the conceptual ideas of a Semantic Desktop, the Web 2.0 and common Tagging Systems. These articles should provide enough background information to understand the following study: [Sau03], [O'R05] and [Bec06].

1.2 Goals of this study

In order to support users to annotate documents and to extract contained topics, three initial goals were formulated. They all address the process of adding annotations to web pages:

1. Experienced topics, that are already described in the Semantic Desktop should be retrieved by processing a web document.
2. New topics, that do not exist in a Semantic Desktop should be proposed for creation and classification into the existing model of personal information.
3. If possible, new topics should be proposed to be inserted into suitable classes of informations. Therefore a proposal may contain the creation of a new class in the model.

In general ConTag's major philosophy can be described in one sentence:

ConTag's functionalities support users to annotate documents, written in natural language, with textual tags in regard of personal concepts managed in a semantic desktop environment.

ConTag aims at a scientifically sound approach to extract, manage and search for tags in the Web 2.0 and the evaluation of hence proposed tag annotations. The quality of the tag proposals as such is not of primary concern and is going to be inspected in follow up works

The result of executing ConTag on a hypothetical Project Description is pointed out in Figure 1.1. It displays the user interface of ConTag, containing the Project Description as a web document and corresponding tag proposals.

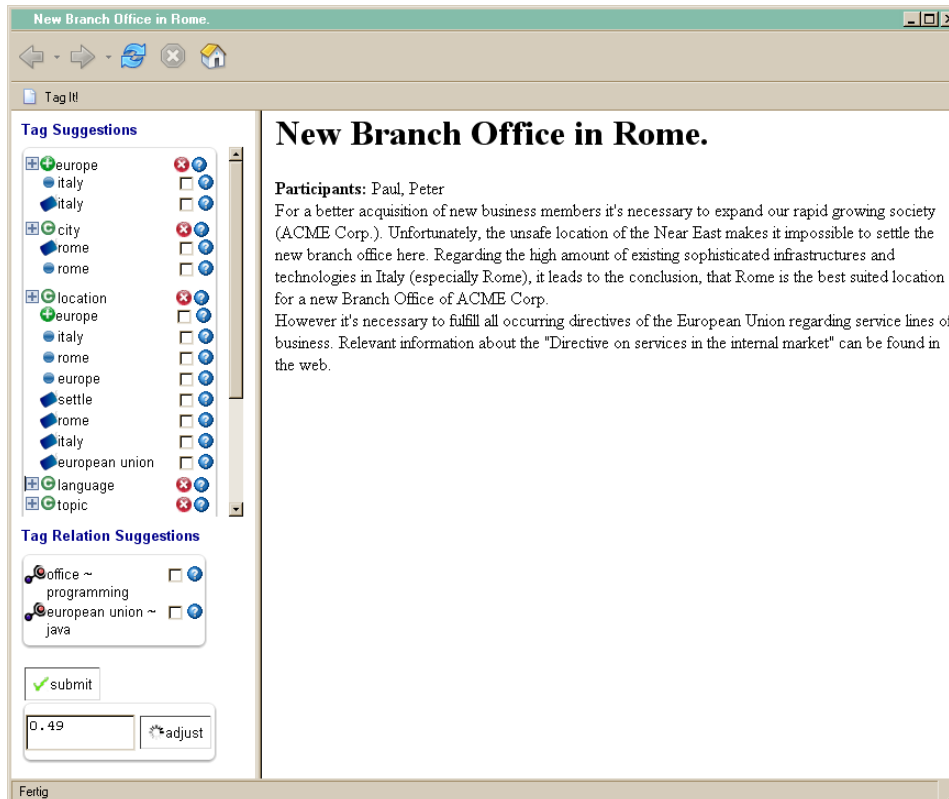


Figure 1.1: Tagging a Business Project Description, using ConTag

The thesis is structured into several chapters:

- *Chapter 2* provides an encyclopedic introduction and explains the evolutionary progress in the managing knowledge and personal information in computer domains and illustrates (1) the basic intention, (2) the impact of recent innovations and (3) the existing problems of existing approaches.
- *Chapter 3* covers the essential ideas and conceptual formulations of ConTag to be developed as a tagging system. A tagging process is described to provide a road map of extracting tags out of text documents.
- *Chapter 4* reformulates the upper mentioned goals and specifies them as formal Use Cases.

- *Chapter 5* lists the most fundamental points of ConTag's architecture and implementation.
- *Chapter 6* evaluates the architecture and used Web 2.0 services in ConTag concerning the Use Cases in Chapter 4.
- *Chapter 7* shortly summarizes the ConTag system to point out open tasks and motivates further developments.

Chapter 2

Motivation

"I sometimes find, and I am sure you know the feeling, that I simply have too many thoughts and memories crammed into my mind."

- Dumbledore, in "Harry Potter and the Goblet of Fire", Joanne K. Rowling -

Information and knowledge management is a major discipline in computer science, especially in the research of developing artificial intelligence. This chapter introduces past and recent approaches in order to figure out essential problems. It explains and classifies technologies and ideas to manage personal information concerning the World Wide Web, Information Retrieval, Personal Information Management and Annotation Systems to provide an understanding of ConTag's conceptual research domain.

2.1 Recent weaknesses in information management

Modern desktop computers are commonly used as personal storage systems for all kind of information. Actual desktop storage systems incorporate harddisk spaces in sizes between 20 to 160 GB. The price for harddisk space has been plummeting for years, so users just tend to buy additional storage devices in order to avoid the task to delete stored information. This increases the average lifetime of an average file by several years and heightens the requirements of effective data access mechanisms.

Until now, users of desktop computers have to store their information in established file systems, more precisely spoken in hierarchical directory systems [Tan92]. These systems handle information in units called files. As a result, hierarchical directory systems force users to divide their collected personal information into

separate files and organize them within a tree like directory hierarchy. This simple approach is highly efficient from the perspective of a persistent storage device. The use of directories facilitates for example the implementation of indexes and the usage of the locality of reference in file caching approaches [FE89].

Indeed, managing information embodies much more than persistent storage requirements. Concerning information storage systems in Human Computer Interaction (HCI) environments, it has to be considered the way humans think [Bus45]. The human creativity and his learning aptitude are surely based upon a mesh structured way of thinking. Different information are always connected within several relationships. For example the project description in Fig. 1.1 contains relationships to Rome, Paul, Peter, etc. An information model should be able to connect the project description with other information resources about these topics. Modern filesystems do not support comparable operations to draw arbitrary relations between two files, except the hierarchic *file-in-directory* one.

The *World Wide Web* (WWW) revolution initiated by Tim Berners-Lee [BL99] reflects the effectiveness of managing information in mesh-like networked structures. Information, stored in a mesh structure, consists of identifiable nodes, representing information and several types of labeled relations connecting these nodes in an approximate semantics as the given information contents are related in reality. Currently the WWW implements the mesh structure with a hypertext technology [W3C03]. A *hyperlink* consists of a label as well as a type, which enables web developers to span a labeled and typed graph inside the WWW. Unfortunately during the rapid expansion of the WWW, website authors tended to abstain from using typed links.

In order to access information nodes in a computable system, it is necessary to propagate system wide distinct identifiers. In closed world systems like modern filesystems a file's absolute pathname tries to fulfill this requirement. But here the system is bound within the file storage *namespace* and the reachable granularities are files. Therefore information stored inside files cannot be addressed from outside. Shifting files into other directories leads to an appropriate change of their absolute path names resulting in an accessibility, which is based upon physical positions.

In open world systems like the World Wide Web, users wish to access information located on different systems. Additionally, identifiers in the WWW have to be persistent to provide at least a minimum of accessibility. The World Wide Web Consortium (W3C¹) is maintaining an encyclopedic overview of past and recent resources addressing research topics ², especially in using *Unified Resource Identi-*

¹<http://www.w3.org>

²<http://www.w3.org/Addressing>

fiers [BLFM05]. Based upon the described structural weaknesses, modern filesystems do not support our way of thinking in an appropriate way and are not able to provide a suitable infrastructure.

2.2 Personal Information Management

The vision of the *Semantic Web* [BLHL01]³ encountered a new approach of information management. The Semantic Web should provide inherent collaboration in describing and linking distributed information resources. It reacted on the suggestion to connect resources with *typed relations*, in order to enhance links semantically with defined type meanings. In [Sau06] Leo Sauermann refined this approach of using *semantic web technologies* to build a *Personal Information Model* called PIMO. A PIMO enables users to store their personal information in their own personal semantic web. He added a conceptual separation between information storage and information access to gain application and storage independence.

Resources such as files or bookmarks are managed in a *resource store*, which is an ontology store based upon any modern filesystem, adding the functionalities to generate global valid identifiers called URIs [BLFM05] for each resource. Information is managed in the *PIMO store*, which is an ontology store to reflect the user's way of thinking. The PIMO store divides information to separate entities called things and is embedded in a Semantic Desktop Server called *Gnowsis* [Sau03].

Gnowsis is an application server providing information access with semantic web technologies. Gnowsis enables the user to search different applications and storage systems for embodied information resources. It expresses these information resources in the *Resource Description Framework* (RDF) [MM04]. With Gnowsis internal URI management, it is possible to access information on a personal computer from outside. Information is no longer tied up to files but may reside in a set of things which are handled in the PIMO store and occur inside a couple of files maintained in the Resource Store (e.g. the person called Peter can be found in several emails, word documents about his project reports, etc.). This approach enables the user to express his personal and often abstract concepts and perspectives apart from hierarchic directory structures. The overall philosophy behind the PIMO is to manage every scrap of user relevant information as simple things. Things may be either abstract to describe and classify information or more concrete to be able to express a single kind of information. (e.g. The abstract thing called "city" may classify more concrete things in a PIMO, such as "Rome" or "Athens"). Instead of imprisoning users within a delimited count of relations, the PIMO allows and also

³<http://www.w3.org/sw>

encourages to link several things together with any user defined relationship. It is now possible to create many information access graphs, instead of being bound to use the existing directory hierarchy.

Due to the employment of ontology and semantic web standards resulting from the initial idea of Tim Berners-Lee, it is possible to query the resource and PIMO store with existing query languages [HBEV04], [PS06]. This provides an easy to use and standardized information access.

2.3 Annotating resources

Discussing the scope of maintaining personal information in computers regards surely information retrieval aspects. Indexing information with any kind of technology supports the finding aspect in this problem definition, by enriching the storage application with metadata to enhance access possibilities. All kind of information resources can be annotated with additional contents. Therefore, several kinds of virtual tags can be loosely defined as sorts of short and concise content annotation to further describe resources. These annotations can be expressed as concise answers to kinds of following questions:

Q1: What is this text about?

Q2: Where was this photo taken?

Q3: Who has written this article?

Q4: When was this song published?

Treating these answers as tags, that are managed as an inverted term list, may provide an intuitive approach of indexing stored resources. But indeed handling answers without knowing the initiating questions results in the loss of important semantics. If we refer the questions specified above to the example project description in Fig. 1.1, the following answers result:

Q1: What is this description about? - A1: project description, New Branch Office in Rome, ACME Corp., acquisition of new partners

Q2: Where was this description written? - A2: Athens

Q3: Who has written this description? - A3: Peter

Q4: When was this description published? - A4: 2006-04-22

An inverted term list consisting of all tags regarding a resource is nothing else than a set of descriptors that link directly to this resource such as the Business Project Description in Fig. 1.1. This approach of describing and retrieving resources with concise terms is called *indexing*. But what is the remaining meaning of an exemplary indexing statement: $index(Peter) \rightarrow file://Figure\ 1.1/$.

Is Peter the answer of the first or the second question? It is not possible to retrieve the original question again. The relation between question and answer has no defined inverse. It remains, that this approach is not suitable to support personal information retrieval in an appropriate way. Instead of using untyped descriptors, the use of typed relationships in the syntactical triple form: *subject, predicate, object* (SPO) is an essential facility in the design of annotating resources in a personal information model. The SPO triple: *Peter wrote "BusinessProjectDescription"* contains much more semantics than an inverted term list and provides additional facilities:

- The relation type "wrote" can be further described and annotated.
- An inverse relation of type "written by" can be defined.
- Reasoning processes providing effective research can be executed on a graph spanned over information nodes and typed relation edges.

At least one lesson learned from the link usage in the WWW is, that users tend to loosely relate different resources with undefined relationships. The approach of letting the user or web author relate the contents in proper form has been not successful, yet. Before discussing possible solutions for user friendly relation approaches and their possible usage in ConTag, it is necessary to provide a brief overview about existing untyped retrieval systems. For this reason it is favorable to define an annotation system in theory (Fig. 2.1).

Let RS be a resource store filled with resources $r \in RS$. Let $uri : r \rightarrow I_{uri}$ be an Unique Resource Identifier to calculate hash values $i \in I_{uri}$ to address resources r in a following way: $RS[i] \rightarrow r$ iff $r \in RS$. $content : r \rightarrow TERM$ is a function, which extract all contents terms $t \in r \wedge t \in TERM$ existing in the resource r and a term set $TERM$. Let $TERM_G$ be such a set of all existing terms. Finally the map data structure M_{ABOUT} allows to search for known resources $M_{ABOUT} : TERM_G \rightarrow I_{uri}$. A system providing a resource store RS and a map M_{ABOUT} is defined to be an ANNOTATION SYSTEM.

Figure 2.1: Annotation System

The usage of *index terms* is widely used among common text search engines such as Google⁴ or Yahoo⁵. The basic philosophy behind this is, that the frequency of term occurrences in documents correlates to the document's major topic. According to example in Fig. 1.1, the term "Branch Office" occurs three times and is an essential topic of this text. Regarding the definition in Fig. 2.1 it is possible to define index term systems formally, in order to point out their weaknesses.

Let $TERM_{RS} \subset TERM_G$ be a set of all existing terms contained in resources $r \in RS$. Then M_{INDEX} is a map data structure similar to M_{ABOUT} , but it only allows to search about $M_{INDEX} : TERM_{RS} \rightarrow I_{uri}$. The benefit of using index words is to be able to automatically generate M_{INDEX} out of RS . Unlikely it is not possible to manage an inherent term semantic for each possible term $t \in TERM_{RS}$. Further and more detailed information about indexing documents and possible improvements are available at [Rij79].

Keywords are often used in library systems. The idea is to decrease the mass of possible search terms by using a controlled vocabulary. These keywords may be well defined in syntax and semantic. Let $TERM_{CV}$ be such a set of controlled keywords. Again there is a new map data structure $M_{CV} \subset TERM_G(\cup TERM_{RS})$ similar to M_{ABOUT} , but it only allows to search about $M_{CV} : TERM_{CV} \rightarrow I_{uri}$. This approach has to be done partially by hand and is very static towards any change in $TERM_{CV}$, because for each change in $TERM_{CV}$ all resources have to be checked again to provide a consistent accessibility in M_{CV} . Removing or adding new terms results in a reevaluation of the whole resource store RS .

The *Tagging* approach just uses $TERM_G$ and M_{ABOUT} to access resources. For each resource, the user is free to invent, extract or reuse terms for further descriptions. Tagging systems like Delicious [Bid04] and Flickr⁶ implement this approach to tag different sorts of resources (see Appendix A.1 for other tagging systems). The advantage of using $TERM_G$ is obviously not to limit the usage of possible resource descriptors. But it suffers from the lack of controll in syntax and semantic, similar to former described index term systems.

Even though ConTag deals with personal information only, collaborative aspects in information publishing and sharing constitute basic requirements in the personal information management domain. The next section introduces some collaborative approaches of annotation systems and helps to point out how ConTag might overcome existing problems from the very first.

⁴<http://www.google.com>

⁵<http://www.yahoo.com>

⁶<http://www.flickr.com>

2.4 Collaborative and social aspects in annotating resources

"Information seeking is more collaborative than generally realized."
[LM95]

The quoted sentence was written in 1995, that is in terms of computer science regarded as ancient. Nevertheless compared to today's situation, commonly used retrieval systems have not been expanded with necessary collaboration approaches, yet. Only one domain extremely pushes collaborative information access, namely the tagging system's domain. It was grounded on projects such as *Annotea* [KK01], which tried to attempt a global and open annotation service to manage metadata about web resources. However, the best example is doubtless the Collaborative Bookmarking System called *del.icio.us*⁷ developed by Joshua Schachter in 2003. Here users are enabled to add a set of labeled tags, consisting of only one term, to annotate resources being accessible by an URL [BLFM05]. These *tag describes resource* relation is maintained in the user's account, which is identified by an URI (e.g. http://del.icio.us/b_horak). It is possible to browse through other user's tag accounts and to query their tagged resource store. Tagging new resources in *del.icio.us* enhances the system wide query results for the chosen tag. Of course, due to the dependency of user inputs, the *del.icio.us*' global resource store is not as huge as Google's for example, but the query results according to personal experience are much more better [GH05b].

The tagging community consisting of *del.icio.us*' users created a repository of connected resource descriptions. In his blog [Wal04], Thomas Vander Wal defined a collaborative connected tag repository to be called a *folksonomy*. The term *folksonomy* is a portmanteau word, merging the terms *folk* and *taxonomy* to one word. It remains questionable whether the usage of the term *taxonomy* is correct in this domain. Nevertheless, tags in a *folksonomy* are simply defined as labels, being attached to resources. He defined his first *folksonomy* to be initially inferred by taking the WWW's link's labels as a description about the target resource for granted. In practice this means to take the set of all existing link labels $TERM_{link} \subseteq TERM_G$ in the WWW as an existing tag set to describe and retrieve resources in the WWW. This strictly refers to the primarily idea of the WWW's design by Tim Berners-Lee [BL99]. Thomas Vander Wal also defined the tag repositories of *del.icio.us* and *flickr*⁸ to be *folksonomies*, as they describe and structure resources with a tag label and are maintainable by an open user community. Finally it remains, that tagging systems provide a collaborative and usable

⁷<http://del.icio.us>

⁸<http://www.flickr.com>

approach in managing resource annotations, because of neither using more or less static taxonomies like it is done with keywords in libraries nor computing insignificant index terms out of term frequencies like it is done in classic information retrieval systems, but let the users easily and quickly decide which descriptors have to be used. An evaluation of tag usages in folksonomies was done in [GH05a]. It shows, that even low semantic tags such as "funny" or "toread" are commonly used to manage found resources. It even evaluated, that different opinions may coincide in a collaborative tag repository.

In spite of all usability aspects, it remains that lightweight definitions of tags in folksonomies implicates essential problems. Tags do not contain any semantics and descriptions beneath their labels. There is no possibility to define what it means to tag a resource with a certain label. Furthermore it is often not possible to identify an existing tag, which enables user A to reuse tag T_B from user B . Of course the usage of the same label may appear to be an equivalent approach, but it is not. The following example shows the difference between a tag identity and a tag label.

2.5 A disastrous tagging example

Peter is planning his holiday and reads a web site about Jakarta, the capital of Indonesia, which is situated on an island called Java. He annotates this website, using a fictional tagging system called *FictionTag*, with three tags: *Jakarta*, *Java* and *Indonesia*. He further defines that these three tags are related to each other in some sense and publishes his annotations.

Paul is interested in J2EE and reads a documentation published on "<http://tomcat.apache.org>". The system Tomcat is a former Jakarta Project, a project that offers a diverse set of open source Java solutions and is a part of The Apache Software Foundation (ASF). He tags this website with the following tag set: *Tomcat*, *Jakarta*, *Apache*, *Java*. With the same motivation, that information should be connected and be published as Peter, he relates all entries in the tag set to each other and publishes them. On the next day, Peter visits his tag account on *FictionTag* and is informed, that *Jakarta* and *Java* are related to *Apache* and *Tomcat*. He consults an web dictionary and reads the following definition about Apache:

A Native American people inhabiting the southwest United States and northern Mexico. Various Apache tribes offered strong resistance to encroachment on their territory in the latter half of the 19th century. Present-day Apache populations are located in Arizona, New Mexico,

and *Oklahoma*. The Free Dictionary⁹

Now Peter wonders, for what reasons Apaches may have emigrated to Indonesia.

2.6 Actual tagging weaknesses

The upper example displays a synonym conflict. A deeper analysis of a tag usage in a philosophic introspection bares, that tags should be treated as resources either. The tagging service *del.icio.us* provides an important feature in allocating a global valid URI for each existing users' tag (e.g. see http://del.icio.us/b_horak/contag) in a user distinct namespace.

Modern tagging systems calculate weak tag relations based on tag cooccurrences, which means that different tags, annotating the same resource, are adopted to be in some sense related to each other. A relation in this manner assumes, that if the user is interested in tag A, he might also be interested in the relating tag B if A and B cooccur in some resource annotations. The tagging service called *Technorati*¹⁰ uses this approach to maintain an index for blog entries and uses tags from different tagging systems.

At first sight these lightweight tagging approaches appear to be sensible, but indeed the lack of sense in tags breeds enormous semantic problems. Polysemies, synonyms and different word inflexions are not considered. Abstraction layers cannot be used to infer obvious coherences in terms of generalization and inheritance. (e.g if the project description in Fig. 1.1 is attached to be written by Peter, it is surely semantic related to the company Peter is working for.) The key issue is, that the listed relations are based on semantics and hence should only be inferred by semantic analysis.

The linguistic coined approach of using SPO triples mentioned in Section 2.3 solves this issue. The Resource Description Language (RDF) as a base semantic web technology was designed to express semantic containing statements in an SPO form to draw relations between several resources. An exemplary tagging system providing a RDF compliant tag repository called *PiggyBank* is described in [HMK05]. *PiggyBank* is an *Annotation System* which manages tags as resources with defined URIs. The user is able to create further annotations for a single tag (even to annotate it with other tags) and is able to use ontologies to maintain them. For example it is possible to express, that the tags with the labels "Big Apple" and "New York City" are equivalent. On the other hand tags with similar labels can be diverged by using different URIs.

⁹<http://www.freedictionary.org>

¹⁰<http://www.technorati.com>

2.7 Lessons learned

Managing and structuring personal information is based on resources and should encourage to draw relations between existing resources and personal concepts. These relations have to be bidirectional to allow effective and fast to compute lookups and query executions. To avoid the loss of semantics in describing resources with tags, it is necessary to use typed relations in form of SPO triples [Daw05]. This provides a general possibility to let the user understand, why a certain tag has been used.

> Peter wrote Business Project Description

< Business Project Description writtenBy Peter

The process of adapting new resources to the Personal Information Model annotates the resource with personal concepts. These concepts are called tags that occur in the resource in an undefined manner. Tags are treated as resources as well and may be annotated with other tags. Publishing tags in a tagging community means to let other users use existing tags to annotate resources. For this reason, there has to be possibilities to express the tags meaning. Existing folksonomies (except Wikipedia) often lack the support of such semantic expressions. Finally a collaborative tagging system has to provide tags with defined identities, existing labels and further descriptions about the tags meaning to let users decide whether to reuse an existing tag or to create a new one. Realizing these requirements strongly points out to the usage of a collaborative tag ontology connecting all personal tags residing well described in personal information models. For this reason the preliminary usage of *folksonomies* should be relieved with *folksologies*, that are tag based ontologies.

Chapter 3

ConTag

"You affect the world by what you browse."
- Tim Berners Lee -

The topic of this study is the design, the implementation and the evaluation of a semantic system, that supports the user of a desktop computer in adding a web document, written in natural language, efficiently into his Personal Information Model. The system should generate proposals how to connect the new information resource into the mesh-like structure of an existing Personal Information Model. This chapter explains ideas and approaches to realize this user support, by using different technologies provided by Web 2.0 services, natural language processing techniques and other research domains.

Due to the former analysis of annotation systems and folksonomies, it was decided to build ConTag on four basic principles:

1. ConTag should behave like a tagging system to support all possible user defined tags in $TERM_G$ in order to annotate and query resources in RS . Additionally the idea of describing resources in a collaborative community should also inspire the evaluation of ConTag.
2. ConTag should propose relevant tags $t \in TERM_{RS}$ corresponding to existing concepts in the Personal Information Model, to ensure a consistent tag annotation, similar to keyword based systems. But in spite of using a static term set $TERM_{CV}$, like it is done in keyword based systems, there should only be a priority in using already existing tags in $TERM_{RS}$.
3. ConTag should compute new relevant tags out of textual contents of the given resource *content* : r and further research in the Web 2.0 environment, to be flexible in adding new information.

4. ConTag as a proactive information support system should provide an inherent explanation philosophy in documenting its tag proposals.

With this elementary definition of abstract requirements of ConTag's conceptual functionalities, it is possible to further specify more concrete requirements to ensure an efficient development of ConTag. First of all, an existing implementation of an adequate Personal Information Model has to be chosen. With the help of existing requirements of other Personal Information Systems, such as *Enquire* [BL80] or *Gnowsis* [SBD05], it is possible to point out several functional quality criteria in order to evaluate Personal Information Models implementations.

3.1 Personal Information Model

In order to act on the resolutions in Section 2.7, [BL80] and [SBD05] the following requirements have to be accomplished in an implementation of ConTag's basic principles:

PIMO A Personal Information Model (*PIMO*) is a storage structure. A *PIMO* manages resources, concepts and relations.

- Let $C_{personal}$ be a set of personal concepts, RS a resource store preliminary defined in Fig. 2.1 and $R_{relation}$ a set of relations r . A storage structure is defined to be a *Personal Information Model* iff $PIMO := C_{personal} \times RS \times R_{relation}$.

SPO Triples Annotations should be expressed in the semantic of subject, predicate, object [Daw05]. A Personal Information Model should therefore provide the administration of relations in such a form.

- Let S be a set of subjects $S := (C_{personal} \cup RS)$, O be a set of objects ($O := C_{personal} \cup RS$) and P be a set of predicates. $R_{relation}$ contains SPO Triples iff $R_{relation} := \{r | r \in R_{relation} \wedge r := (s, p, o)\}$ and $s \in S \wedge p \in P \wedge o \in O$.

Invertible Relations To provide an easy crossing between annotations and resources and backwards, relations should always be invertible.

- A relation r is called to be *invertible* iff a reverse relation $r^{-1} \in R$ exists, which is defined as follows: If $r := (s, p, o)$ then $r^{-1} := (o, p^{-1}, s)$ and $p^{-1} \in P$.

Sound Taggings A Personal Information Model is called to support *Sound Taggings* if it supports the previous requirements.

ConTag as a semantic tagging system should support *Sound Taggings* and therefore needs a tag repository which accomplish the above described requirements. *Gnowsis*¹, a *Semantic Desktop* [SBD05] implementation by Leo Sauermann embeds a personal ontology approach called *PIMO* [Sau06]. The user is enabled to maintain his collected informations in his Gnowsis PIMO separately, according to different independent abstraction layers [XC05]. Abstract mental models such as concepts are stored in Gnowsis' *PIMO store*, where they are simply defined as user aware *things*. Physical resources like files, emails or web sites are accessible via a *resource store*, where they are stored as *resource manifestations* independently from underlying data formats. Things are connected in an RDF graph with different kinds of loosely defined relations. All relationships have to be bidirectional according to [Roh05] to support forward and backward inference and search mechanisms. Resource manifestation and things can be connected with an occurrence relationship. A thing participating in an occurrence relationship is defined to be a tag for the target resources. According to the definition in Fig. 2.1 the resource store is equivalent to *RS* whereas the PIMO store corresponds to a semantic term set $TERM_{PIMO}$. Additionally, the PIMO store manages all relationships between things or resources. The PIMO can therefor be handled as an *annotation system*. Furthermore the Gnowsis PIMO supports *Sound Taggings*, because it is a formal *PIMO*, using *SPO Triples* in form of RDF Statements and constraining the PIMO store to use *Invertible Relations* solely.

3.2 About the nature of tags

In order to explain the semantic of tags, the following citation provide capital importances:

It's very much people tagging information so that they can come back to it themselves or so that others with the same vocabulary can find it.
(Thomas Vander Wal [Ter05] - dedicated inventor of folksonomies)

"The job of tags is not to organize all the world's information into tidy categories," ... "It's to add value to the giant piles of data that are already out there." (Stewart Butterfield [Ter05] - one of Flickr's co-founders).

Thomas Vander Wal explained, that the necessity of annotating resources with tags, is to relocate these resources in future more easily. Stewart Butterfield commented, that the annotation process of tagging resources helps to add more semantics to

¹<http://www.gnowsis.org>

resources. Due to the fact, that tags are published and maintained by individual and self-contained users, the following comment by us should complete the explanation of tags:

Each user uses his own set of tags. Annotating resources with these tags is a highly subjective procedure. Considering tagging as a subjective procedure results in the fact, that the evaluation of right or wrong tags, might not be realizable. Indeed, no annotation system is able to propagate a set of generally accepted relevant tags for a resource like preaching the gospel.

Summarizing the value of a tag in one sentence leads to the following definition by Dave Becket [Bec06]:

A tag is a word or short phrase which has a meaning to a person, not taken from any pre-designed system.

The design of ConTag considers this definition and manages tags as fully liberated resources in the semantic web domain.

In nearly all Tagging Systems, tags are strongly characterized by their *label*. ConTag recognizes this and emphasises the label in computing tag semantics.

Tags are always created by a certain user and should therefore be related to an existing *namespace*, that corresponds to the user's identity. Del.icio.us uses the URL of an user account to express the tag's affiliation (see http://del.icio.us/b_horak). This approach provides an intuitive correlation between a tag and its origin. Therefore a tag's namespace may also be defined as the tags's *location*. This definition accommodates the concept of URIs in [BLFM05] with the following format:

```
<scheme>://<authority><path>?<query>
```

In order to access a certain tag with a specified URI, it is necessary to define an access interface. Although ConTag won't implement an inherent access mechanism to list or alter tag properties, it is generally recommended to provide interfaces based on the HTTP [FGM⁺99] protocol like the Representational State Transfer (REST) designed by Roy Fielding [Fie00]. A typical identification of such tags might result in URIs like:

```
http://www.dfki.uni-kl.de/~horak/tags/ConTag
```

RESTful queries might be able to provide views on tags' properties. For example, the URI <http://www.dfki.uni-kl.de/~horak/tags/ConTag>

occurrences is being attached with a query called *occurrences* and might return a list in RSS [BDBD⁺01] design containing annotated resources. Another example <http://www.dfki.uni-kl.de/~horak/tags/ConTag?relatedTags> might also return a list formatted in RSS containing tags being related to a tag labeled with "ConTag". Due to simplicity, RESTful lightweight interfaces seem to be most popular. The online book store Amazon reports, that 95% of the usage of offered webservice is of the lightweight REST service [O'R05]. Due to the fact, that tags as resources may be annotated with several properties and relations, it is obvious, that the basic descriptions of a tag should be expressible in a collaborative tag ontology. Richard Newman proposed a first approach in building a tag ontology (see Fig. 3.1 found at <http://robustai.net/folksonomy/Tag-ontology.jpg>) in [New05]. It is inspired by several semantic web technologies (RSS, SKOS [MB05] and FOAF [BM05]) and should be able to express all necessary informations to manage and share distributed tags.

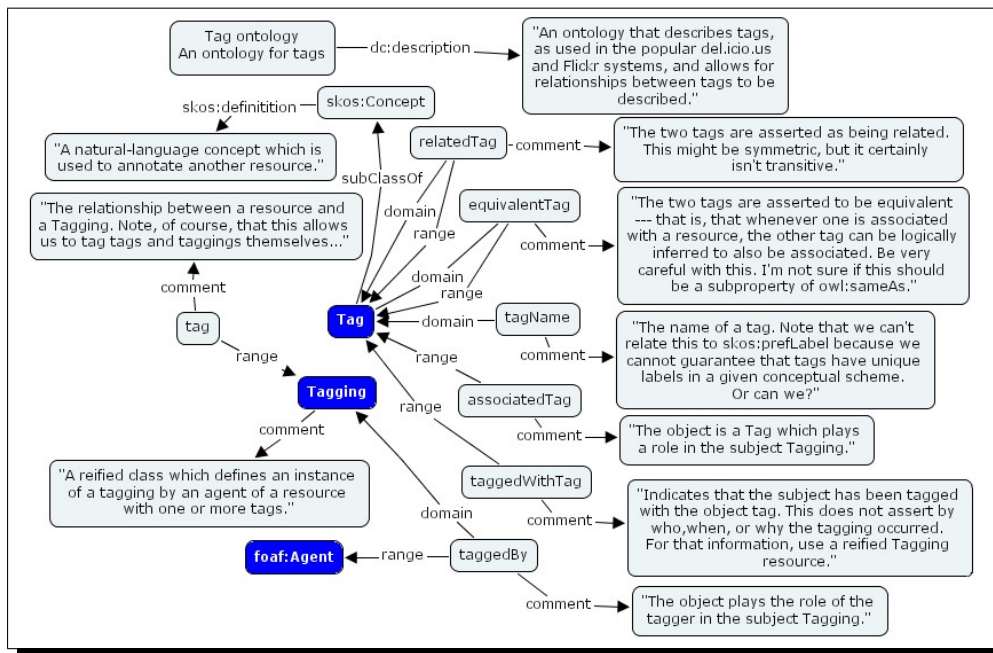


Figure 3.1: Tag Ontology

The tag management in ConTag covers on the one hand the proposal of existing tags in the Personal Information Model for being relevant regarding a certain text resource, on the other hand the proposal of the creation of new tags to figure out topics, that have been unknown in the PIM yet, but occur in the resource. These

proposals have to be well explained to take into consideration, that the user might not have heard about it.

3.3 Effective explanations

ConTag acts as a *Decision Support System*, that helps users to maintain their information. In Personal Information Models, users try to describe the world the way they see and perceive it. Out of this description, ConTag computes an understanding in order to propose, how new resources might fit best into it. It is supposed, that the Personal Information Model is dealt with great care and might be a sanctum on the user's desktop. Each proposed alteration should withstand critical user examinations, to provide arguments to accept or decline the proposal. On the other hand it enables the user to quickly annotate new information about the text-contained topics. In order to support the user in understanding generated proposals, the following set of questions should always be answerable, when proposing a tag:

- What does the proposed tag mean?
- Why does the tag occur in the resource?
- Why should the tag be aligned to a certain concept in the Personal Information Model?
- In what degree of significance does an extracted topic of the resource correlate to an existing concept in the Personal Information Model?

In [BL97] Tim Berners-Lee described the concept of an "*Oh yeah?*" *button*. It should refer to a summary of trustworthy explanations and background information for existing relations and resources in the WWW. This concept of an "*Oh yeah?*" *button* has to be realized in ConTag to provide concise answers to the listed questions above.

In science the descriptive sense of any concept is mostly expressed by using definitions. Defining topics means to express and serialize their meaning by using prose and logic expressions. This seems to be also a promising procedure to explain tags' semantics in ConTag.

Out of the multiplicity of existing definition types, only *lexical definitions* fit to ConTag's application context. A lexical definition may also be called a *dictionary definition* and reports the meaning of a term as it is normally used. Aristoteles specified a special technique to express definitions in a proper and understandable manner in the following loosely translated lemma:

*You create a definition by relating the higher abstraction of a given concept and mention the existing concept's specific differences.*²

The same way humans lookup in dictionaries to research for unknown meanings of terms, ConTag might either get used to compute tags's semantics out of definitions in accessible web dictionaries like WordNet³ and describe tag proposals by listing found definitions in the user interface. Additionally, it might be possible to parse these definitions written in natural language, by using Aristoteles' lemma to extract contained hypernyms.

3.4 Linguistic aspects

In order to facilitate the computation of tags, their semantics and possible alignments to match the personal information model, it was decided to confine ConTag's linguistic coverage to the English language. Nevertheless, a couple of linguistic problems have to be considered during the generation of tag proposals.

A set of tags may have the same label but different meanings. These terms are called *synonyms* and should be supported by ConTag.

On the opposite one dictionary definition may define the same meaning for different terms. These are called *polysemies* and should be expressible in the Personal Information Model, by permitting the usage of several labels for one defined concept.

Texts written in natural language frequently may contain one and the same concepts in different grammatical flections (e.g. ontology and ontologies). Additionally to that, occurring concepts might be expressed with acronyms (e.g. WWW). Due to the usage of definitions, it should be possible to connect or normalize these linguistic versions in ConTag. The use of existing algorithms in the Information Retrieval domain may provide the reduction of terms to their normal form (Porter stemming algorithm [Por97]: ontologies → ontolog) or principal form (Kuhlen's normalization [Kuh76]: ontologies → ontology).

Considering effective explanations and linguistic aspects in a the development of Tagging System increases the quality of the tag management as such.

3.5 The idea behind ConTag

The essential idea of ConTag is described to take a text and an existing ontology as input, then compute a topic map out of the text, which finally is aligned to the

²Definitio fit per genus proximum et differentiam specificam.

³<http://wordnet.princeton.edu>

user's Personal Information Model. This alignment shall be presented to the user in an user interface (UI), to enable him to accept or decline the given alignment proposals. ConTag's emphasis is the realization of this process as a whole. ConTag's tagging process is drafted in Fig. 3.2 and consists of several intermediate steps.

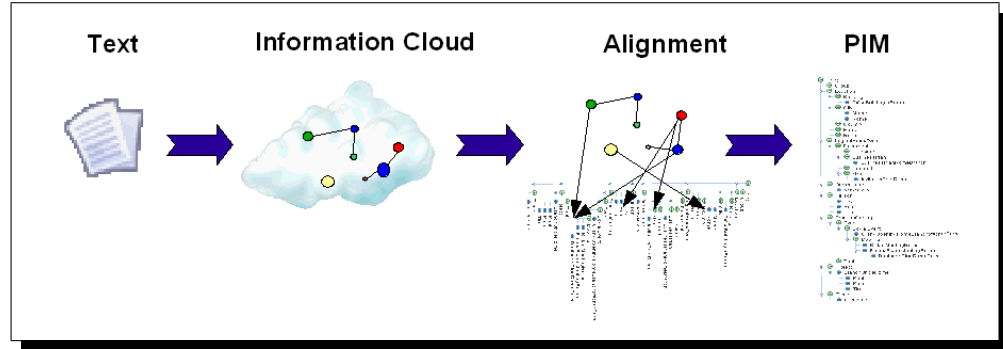


Figure 3.2: Tagging Process

1. In the first step, the user opens a document on his desktop. It is assumed, that a Semantic Desktop server is running in background.
2. In the second step the user decides to insert the text resource into his PIMO. Therefore, the application that displays the text resource, has been enhanced with a button to start the ConTag computation.
3. During the third step called *Topic Extraction*, ConTag tries to extract all topics mentioned in the text and notes them in a data structure called *Information Cloud*.
4. The fourth step is called *Alignment Generation*. ConTag tries to align each topic in the Information Cloud with semantic corresponding existing concepts in the Personal Information Model.
5. The fifth step, called *Alignment Execution*, executes the generated alignment and converts contained topics into PIMO internal concepts. In terms of the Gnowsis PIMO these concepts are called *things*.
6. In the last step, the user is enabled to review his tagged resources by browsing through his Personal Information Model.

A huge amount of technologies and sophisticated approaches may be suitable to take part in each of ConTag's tagging process's steps. Only a few of them will

be used as representatives in this thesis. Due to the wide span of research fields, ConTag touches, it was decided to use existing Web 2.0 services to solve problems so far it is possible. The following list describes how several kinds of computer science technologies may support the maintainance of tag proposals during all described process steps. Each usage of a certain approach should increase the degree of quality measured by information retrieval units like (Precision, Recall, Fallout). These measurements are further described in Chapter 6.

1. At first sight, opening a document in a certain application may be classified as a trivial and uninteresting action. But certainly, it contains a set of semantic background information. First of all it can be assumed, that the document is focused by the user's interest. User context aware technologies, like they are described in [Sch05] may support the tagging process, by generating priorities of current relevant concepts in the PIM. These priorities could be used to create better alignments to increase the degree of precision.
2. The decision to insert a text resource into the user's PIMO, proves the resource's quality, at least in some existing text sections. Allowing the user to mark interesting sections or to underline certain terms will decrease the amount of extracted topics in information cloud, which provides a better recall, by decreasing the degree of fallout. Technologies like *Bookmarklets*⁴ or *Greasemonkey*⁵ may be usable for it.
3. Extracting information from texts is the major topic of text retrieval systems or rather text mining systems like they are described in [AI99]. Challenges such as the detection of simple phrases or names, locations and other topics are in focus of Natural Language Processing (NLP) methods. This *Information Extraction* is the most valuable step in ConTag's tagging process. It should result in building an *topic map*, called Information Cloud, to describe the text's content. Due to the fact, that NLP based computations are very intensive, the existence of independent Web 2.0 services may be used to source out the computation of different information extraction jobs. These services are described in Section 5.8.1. Additionally, the use of effective explanations, which are mentioned and demanded in Section 3.3 depend on the existence of a well documented generated resulting topic map in the third step of ConTag's tagging process.
4. The process of aligning the results of step three into an existing Personal Information Model is topic of a research field called *Ontology Alignments*.

⁴<http://www.bookmarklets.com>

⁵<http://greasemonkey.mozdev.org>

The *PhaseLibs*⁶ project provides a set of state-of-the-art tools to realize and express alignments between two ontologies. One requirement of generating alignments is, that the measurement of alignment confidences has to be normalized. Alignments should be rateable by a user to enable him in inserting only relevant and correct tag proposals into his Personal Information Model.

5. The execution of an alignment, has to assure, that the transitive closure of all relations in an ontology alignment doesn't produce any semantic conflicts like inconsistencies in the user's Personal Information Model (e.g. cyclic part-of relationships). It is also assumed, that an alignment maybe serialized in an ordering to ensure that the executor may implement every relation in that ordering at at least one possible point of execution (e.g. cyclic part-of relations of two new concepts have to be avoided).
6. Several views on tags exist in modern tagging systems. The most popular one (used in del.icio.us, flickr, technorati, etc.) is the *tag cloud*, which is pointed out on the left side of Fig 3.3. Here tags are mostly sorted by label. The font size of a tag entry expresses the count of resources being tagged by it. The tag cloud provides a flat view on a tag set. A hierarchical perspective is used in Gnowsis by rendering relationships like *subclass*, *instanceOf* or *partOf* (see Fig. 3.3 on the right side). Another promising approach is done in Semanlink [?], where a user is enabled to browse through his tag set and all existing annotations in a wiki⁷ manner. Nearly all tagging systems provide at least weak semantic relations enabling users to browse through all system managed tags by following their relations.

ConTag won't provide any browsing mechanisms and uses the existing Gnowsis view to enable the user in searching for tags and attached resources.

3.6 Using Web 2.0 services

Tim O'Reilly introduced in [O'R05] the term *Web 2.0*, to express a change of perspective in many fields of the WWW. He explained, that the community driven approach of collecting and publishing information entered a new phase of development. Small web services were provided to publish and access information of encapsulated domains. Companies like Amazon or Ebay developed easy to use web service interfaces to query their product store and search engines like Google and Yahoo provided similar interfaces to enable developers to use their retrieval

⁶<http://phaselibs.opendfki.de/>

⁷see <http://www.semanlink.net/tag/favori> as an example



Figure 3.3: Tag Views

engines. Tagging systems, which are Web 2.0 technologies, published simple interfaces to let other systems benefit of their existing resource annotations. Tim O'Reilly defined three lessons to develop in a Web 2.0 environment:

1. Support lightweight programming models that allow for loosely coupled systems.
2. Think syndication, not coordination.
3. Design for "hackability" and remixability.

The development of ConTag follows these instructions and is therefore based on a choreography of composed Web 2.0 services, that is explained in Chapter 5.

Chapter 4

Use Cases

"We are stuck with technology when what we really want is just stuff that works."

- Douglas Adams -

The scope of functionalities in a modern software system should be well defined using engineering principles. It is profitable to express each functionality, the user is aware of, with a technique called Use Case. The following Use Cases in this chapter specify the desired and specified behavior of ConTag formally, out of an user perspective. Considering the *Use Case Template* of Alistair Cockburn in [Coc98], three essential Use Cases were defined. Additionally an optional fourth Use Case is listed, that was specified during the development of ConTag.

All Use Cases share a set of commonalities:

4.1 Use Case commonalities

Scope

The scope of each of ConTag's functionalities is to support the user in annotating web documents with tags in the user's Personal Information Model. While processing the document, certain tags may not exist in the personal information model. In this case ConTag should propose the creation of new tags to let the user annotate the web document more efficiently.

Level

All Use Cases in this study specify functional requirements at user level. That means, they describe system functionalities out of the user's perspective.

Trigger

The scope of ConTag is reached, when the user accesses a web document in a suitable extended browser. The triggered action to start ConTag's computation is defined to be a mouse click on the button in the browser extension.

Precondition

ConTag is designed to communicate with an installed Semantic Desktop system on the user's personal computer. The user's browser to be extended with a button to start ConTag's procedure. It is assumed, that the PIMO contains at least a minimal set of initial concepts. Each concept contains basic semantic descriptions to allow a significant computation of tag proposals. Additionally, it is suspected that the user possesses an open and stable internet connection.

Success

The scenario of using ConTag will be generally concerned as successfully solved, if the generated proposals seem to be reasonable in most of the cases with regard to the underlying PIMO. Accepting a proposal means that the current web document will be added as a new occurrence of a tag corresponding *Thing* in the PIMO. The evaluation of ConTag validates each functionality against these success thresholds.

Scenario

To explain all functionalities used in the following Use Cases, each Use Case shortly outlines a short user story based upon the example in Fig. 1.1 and a hypothetical user Paul.

The UI of ConTag is designed to behave like a common tagging system. Figure 4.1 outlines these communication interfaces in a simple UML Use Case diagram. The overall scenario is as follows:

The hypothetical user Paul opens a web document in his browser. The content of this web site can be found Fig. 1.1. Paul is interested in this resource and wants to add occurring topics into his PIMO (Paul's PIMO is printed out in Appendix A.3). Therefore he presses a button in his browser to start ConTag. ConTag generates several tag proposals containing extracted topics from out the document. Now Paul is enabled to decide which proposal to adopt into his PIMO, by inspecting and committing them in a graphical user interface.

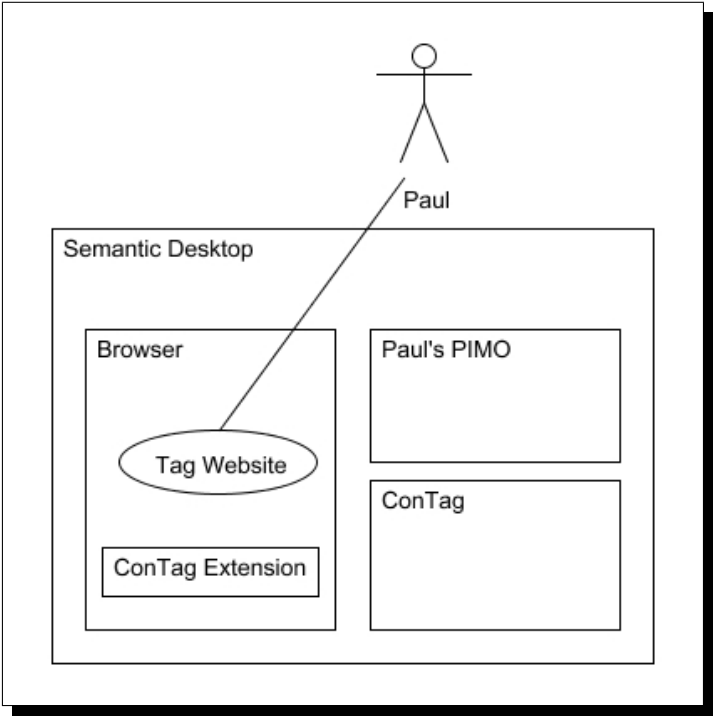


Figure 4.1: Use Case: Tagging a web document using ConTag

4.2 Use Case 1: Retrieve existing instances

Paul already created *Italy*, *Rome* and *Europe* as things in his PIMO. Processing the document about a *New Branch Office in Rome* with ConTag reveals, that these PIMO things occur in th document.

Goal

Processing a website with ConTag results in a set of documented and weighted proposals of existing PIMO instances in the user's PIMO, which ConTag recognizes to occur in the website. The user can decide to accept these proposals into his PIMO or reject them.

Success

This scenario will be concerned as successfully solved, if the generated proposals are well documented, the user interface enables the selection of a subset of accepted proposals and the extracted existing PIMO instances seem to really occur in the document.

Failure

This scenario will be concerned as not successfully solved, if the generated proposals contain wrong PIMO instances or misses apparent instances with regard to the quality of the underlying PIMO.

4.3 Use Case 2: Extract new instances

Paul hasn't created a PIMO thing called *European Union*, yet. Processing the document about a *New Branch Office in Rome* with ConTag reveals, that the *European Union* occurs in the document and is therefore proposed to be inserted into his PIMO by classifying it as a location.

Goal

Processing a website with ConTag results in a set of documented and weighted proposals of new and not existing PIMO instances of existing PIMO classes in the user's PIMO, which ConTag decided to occur in the website. The user can decide to accept these proposals into his PIMO or reject them. Accepting instances means, that each instance will be created in the PIMO and the web resource is added as a new occurrence.

Success

This scenario will be concerned as successfully solved, if the generated proposals (the proposed PIMO class and PIMO instance) seem to be reasonable in most of the cases with regard to the underlying PIMO.

Failure

This scenario will be concerned as not successfully solved, if the generated proposals of existing PIMO classes and new PIMO instances seem to be unreasonable in most of the cases or if apparent proposals are missing with regard to the underlying PIMO. Additionally proposed PIMO classification proposals may be too abstract or too concrete in the PIMO class hierarchy. Unreasonable instance proposals may be duplicates of existing corresponding instances in the PIMO, or instances of wrong classes.

4.4 Use Case 3: Classify instances to extracted classes

Paul hasn't created a PIMO class called *region*, yet. Processing the document about a *New Branch Office in Rome* with ConTag reveals, that *Europe* occurs in the document. A deeper semantic analysis inside ConTag's Tagging Process results in the assumption, that *Europe* may be an instance of *region*. Therefore *Region* is proposed to be a new PIMO subclass of *location* and the new PIMO instance *Europe* is proposed to be additionally an instance *region*.

Goal

Processing a website with ConTag results in a set of documented and weighted proposals of new or existing PIMO instance of new PIMO classes, which ConTag decided to occur in the website. Accepting a proposed PIMO class and its PIMO instances means, that the PIMO class will be created under a proposed existing PIMO superclass and its instances will be created or linked as PIMO instances, either. The web page will be added as a new occurrence of all mentioned PIMO instances.

Success

This scenario will be concerned as successfully solved, if the generated proposals (the proposed PIMO class and PIMO instances) seem to be reasonable in most of the cases with regard to the underlying PIMO.

Failure

This scenario will be concerned as not successfully solved, if the generated proposals of new classes or instances seem to be unreasonable in most of the cases.

4.5 Use Case 4: Create relations between instances

This Use Case was added during the development of ConTag. It specifies an additional feature resulting as a side effect of ConTag's Tagging Process. Therefore it will not be evaluated.

Paul's PIMO contains a thing called *BusinessPlanRomeBranch*. Processing the document about a *New Branch Office in Rome* with ConTag reveals, that the *European Union* occurs in the document and is related to *BusinessPlanRomeBranch*. Therefore it is proposed to create a new semantic relation between both instances, if *European Union* is inserted into Paul's PIMO.

Goal

Processing a website with ConTag results in a set of proposed PIMO things, which ConTag decided to occur in the website. Some of these proposed PIMO instances are also proposed to relate to other existing PIMO things. The user can decide to accept these relation proposals into his PIMO or reject them. It is only possible to accept relations, between existing or accepted PIMO instance proposals.

Success

Generally, this scenario will be concerned as successfully solved, if the generated proposals of concept relations seem to be reasonable in most of the cases. Adopting the relation proposals will relate each of the given tag pairs to each other with an PIMO relation called `relatedTo`. But due to the lightweight specification of this Use Case, no evaluation is going to be performed.

Chapter 5

Design

"I really love Artificial Intelligence - It's so natural."

- Captain Bird in the musical "Abydos", Andy Kuntz -

The following chapter describes the implementation of ConTag as a Tagging System, following the requirements of Chapter 3. The general architecture and included services are summarily explained in Section 5.1. In Section 5.2, a complete iteration of ConTag's computations based upon the example Project Description in Fig. 1.1 is outlined to point out the procedure of querying and processing information in a Web 2.0 service choreography.

ConTag is designed to realize the ideas mentioned in Section 3.5. It strictly follows the *Tagging Process*' steps, specified in Fig. 3.2, as the priority of ConTag is defined to provide at least a method of resolution for each step. Due to this focus on the whole process, the design of ConTag encourages to source out a set of well defined problems and let them be solved by external Web 2.0 services. In order to gain the ability to evaluate the effects of used web services on the quality of the resulting tag suggestions, ConTag's runtime environment can be controlled by different runtime parameters.

5.1 General architecture

ConTag consists of different programming packages. Each package contains independent functionalities, that are separated into a set of services (see UML Package Diagram in Fig. 5.1) in order to be compliant to modern service oriented software architectures (SOA) and especially be conform to the Gnows Architecture. The following list outlines functionalities of implemented services and other programming modules.

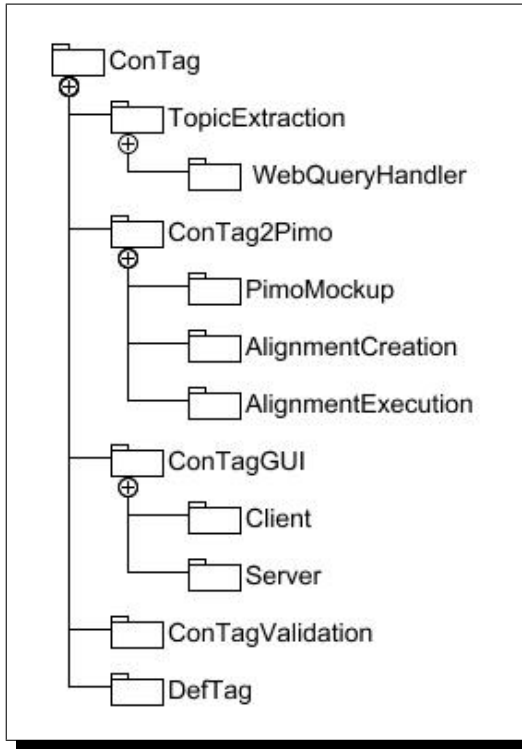


Figure 5.1: Architecture

ConTag This is the platform of the *ConTag Runtime Environment (CRE)*. It maintains possible configuration parameters to controll the execution of internal and external services. Configuration parameters are for example declarations to specify the Web Services to query, general timeout values to delimit the service’s response times, thresholds to specify string or set comparisons in terms of fuzzy logic, etc.

Topic Extraction The Topic Extraction service provides a basic programming interface to extract existing topics out of a given text resources, written in natural language. It generates a topic map, that contains occurring topics, relations between these topics and additional found topic definitions. Therefore the Topic Extraction service executes three intermediate steps in ConTag’s Tagging Process: (1) to look up significant topics occurring in the text resource, (2) to search for definitions about these topics and (3) to query for topic relationships to other topics. It was decided not to use existing topic map vocabulary standards like *SKOS* [MB05] or *XTM* [PM01], but

to use a self defined vocabulary which is optimized for ConTag's computations and compilable with the *RDF2Java* [SS03] technology. *RDF2Java* is a smart toolkit to wrap RDFS Classes in Java Classes in order to provide an information access to RDF resources, that is similar to existing *Component Frameworks* like *Java Beans* ¹.

WebQueryHandler The Web Query Handler is an abstract interface to provide a standard access mechanism to Web 2.0 services. It manages queries using communication protocols like REST [Fie00] or DICT [FM97] and uses different *WebQueryResultHandler* to parse service results in order to return relevant information in a standardized data structures. Due to the fact, that most of existing Web 2.0 services return XML marked up values, many *WebQueryResultHandler* are implemented using the XML query language called XPATH [CD99].

ConTag2Pimo The ConTag2Pimo service manages all occurring data transfers between Gnowsis (especially the PIMO) and ConTag. It wraps the Gnowsis interface, to provide a Gnowsis independent development of ConTag. This independent development permits the possibility to connect ConTag to other ontology based systems.

PimoMockup In order to validate ConTag's tag proposals, it is necessary to work on a static and ideal Personal Information Model, filled with significant test data. That means, all existing things in the PIMO Mockup are well defined concerning the computability of their semantics according to ConTag's Topic Extraction, Alignment Creation and Alignment Execution.

AlignmentCreation The Alignment Creation service provides an interface, that takes a topic map as input and computes an ontology alignment regarding an existing PIMO as output. The resulting alignment between topic map and PIMO contains the set of tag proposals to be displayed in the ConTag GUI service. The creation of an alignment is implemented with a Nearest Neighbor Classification approach (see Section 5.5).

AlignmentExecution User reviewed tag proposals are merged into the user's Personal Information Model by executing the Alignment Execution service. This service serializes the reviewed ontology alignment and executes each contained relation between topic map and PIMO in a specific and executable ordering. (see Section 5.7)

¹<http://java.sun.com/products/javabeans>

ConTag GUI ConTag's generated tag proposals are displayed in the ConTag GUI service. This UI provides users a decision base to rate tags as being relevant or not. The ConTag GUI service renders a new HTML Frame beside the current web page in the currently used web browser. The implementation is based on a technology called *AJAX* (Asynchronous Javascript and XML) [Gar05], which allows to manipulate information on a certain web page without rerequesting the whole page after each alteration, by using asynchronous protocols between client and server.

Server The GUI's server component is implemented as a common *Java Servlet*². It controls the model containing the tag proposals and implements all actions enabling the user to review ConTag's set of tag proposals. The design is conform to the *Facade Design Pattern* [GHJV95] and was additionally inspired by a J2EE Pattern called *Session Facade* [Mic06]. Concerning the *Session Facade*, *Business Objects* were implemented as Jena2Java Resource Wrapper Classes using the *RDF2Java* [SS03] toolkit.

Client The GUI's client component renders a HTML based overview out of the server's model and allows the user to interact, especially to declare which tags he wants to merge into his PIM. It is implemented in Java using the *Google Web Toolkit*³ and finally transformed to HTML and Javascript.

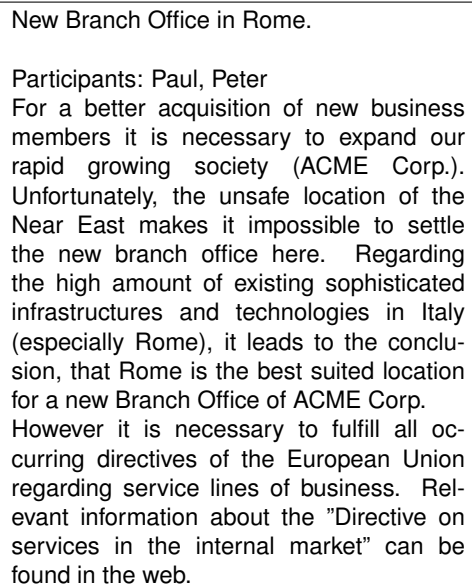
ConTag Validation This service provides different API's, test corpora and their corresponding Ground Truths to create a significant validation and evaluation of the final generated tag proposals or different intermediate results like topic maps, phrase extraction, etc.

DefTag DefTag is a self made Web 2.0 service. It takes a definition written in natural language of a certain term as input and computes a set of hypernyms or genuses concerning to Aristoteles' lemma of writing definitions. DefTag is used by the *TopicExtraction* service to compute superordinate concepts of a certain topic and its definitions.

This compendious explanation of ConTag's services should introduce ConTag's solutions concerning the Tagging Process described in Section 3.5. The following section describes the progress of extracting tag proposals using these services based on the hypothetical web page contents listed in Figure 5.1.

²<http://java.sun.com/products/servlet>

³<http://code.google.com/webtoolkit>



New Branch Office in Rome.

Participants: Paul, Peter

For a better acquisition of new business members it is necessary to expand our rapid growing society (ACME Corp.). Unfortunately, the unsafe location of the Near East makes it impossible to settle the new branch office here. Regarding the high amount of existing sophisticated infrastructures and technologies in Italy (especially Rome), it leads to the conclusion, that Rome is the best suited location for a new Branch Office of ACME Corp. However it is necessary to fulfill all occurring directives of the European Union regarding service lines of business. Relevant information about the "Directive on services in the internal market" can be found in the web.

Figure 5.2: Business Project Description

5.2 ConTag's Tagging Process

The Tagging Process starts with a user decision concerning a web page displayed in a browser, to be processed by ConTag. Therefore he clicks on a button, being situated on his browser's bookmark pane. This button has been implemented with a technique called *Bookmarklet*⁴, which is a small *one-liner* written in the *Javascript* programming language, that is supported by nearly all existing browsers. Activating a Bookmarklet means to execute the script on the currently displayed web page contents. The following Bookmarklet takes the URL of the actual web site and passes it via a HTTP-request to a local HTTP-server.

```
javascript:qurl=location.href;void(window.  
  open('http://localhost:8888/CONTAG?uri='+  
  escape(qurl)))
```

Pressing the button calls the *ConTag runtime environment* to start the *Topic Extraction Service*. This is the beginning of a Web 2.0 choreography. First of all, denoted content analysis services are queried for occurring topic information. At the moment, two content analysis services are in use:

⁴For further research about Bookmarklets see <http://en.wikipedia.org/wiki/Bookmarklet>

1. Tagthe.net (see Subsection 5.8.1) and
2. Yahoo's web service called Term Extraction. (see Subsection 5.8.1)

The resulting terms are managed in a data structure called *Information Cloud* (see Section 5.3).

During the next step, for each topic in the Information Cloud two further researches query for more information. A definition lookup queries web dictionaries, by using the *DICT* protocol, which is explained in detail in Subsection 5.8.1, to gain existing definitions written in natural language. These definitions are attached to their respective topics residing in the *Information Cloud*. A succeeding lookup calls a hypernym extraction service called *DefTag* (see Subsection 5.8.1) to extract a set of superordinate concepts. These superordinate concepts are managed in the *Information Cloud* either and are related to their origin topics.

At the same time, another lookup queries different Web 2.0 services to gain *word associations* concerning each topic. The lookup considers four services at the moment: (1+2) A web service, providing an access to the *Wikipedia Online Encyclopedia*, a collaborative web dictionary system (see Subsection 5.8.1). This service enables ConTag to search on the one hand for existing articles about certain topics, on the other hand it extracts outgoing and inbound links to or from other articles. (3+4) Two services are queried using the *DICT* protocol: The *Moby Thesaurus II*, containing synonyms and other thesaurus data and the *WordNet Dictionary* to extract a set of synonyms for a given term.

After finishing the *Topic Extraction* step, the Information Cloud contains all relevant information occurring in the grounded website. This graph data structure can be serialized into an RDF/XML format, which is defined by the internal topic map vocabulary (see Fig. 5.3). The resulting RDF model is send to the *AlignmentCreation* service to create tag proposals, by generating an alignment between the topic and the user's PIMO (see Section 5.4).

After the *Alignment Generation* step, tag proposals can be displayed in the current browser besides the original website. The ConTag GUI service renders a new HTML frameset, that includes two frames drawn in Fig. 1.1. The left frame displays the tag proposals, strictly speaking it lists the different alignment relations between the web site's topics an the user's PIMO.

Using ConTag's GUI enables users to decide which proposals to adopt into their PIMO by selecting provided HTML check boxes.

After finishing this review, the user clicks on a button called *submit*. The ensuing *Alignment Execution* service inserts all successfully reviewed proposals into the user's Personal Information Model and relates new and existing tags with an occurrence relationship to the current web site.

At last the user is able to browse through his PIMO and may perceive each existing thing as tag, which annotates a set of web resources, he has tagged before.

5.3 Information Cloud

The Information Cloud is a graph based data structure to manage information (topics, relations, definitions) about one specific resource in terms of topic maps. The specification of an Information Cloud is given in the EER diagram in Figure 5.3.

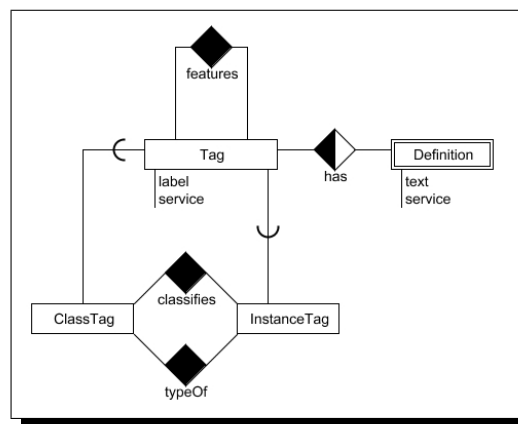


Figure 5.3: Internal Topic Map

A *Tag* describes a concrete topic with a certain label. Tags may refer to each other by using the *feature* relationship, which is left undefined in its semantics. That means two tags, being connected with this relationship are related semantically in some unknown sense. A *Tag* may possess a set of *Definitions* consisting of a text, and an originating web service. Tags may be *ClassTags* if they participate in a relationship called *classifies* relating a *ClassTag* with an *InstanceTag*. On the other hand, an *InstanceTag* is a *Tag* participating in a relationship called *typeOf*, relating it with a *ClassTag*. It is defined, that the relation *classifies* is an *inverse relation* of the relation *typeOf*. Both relations are based on the *hypernym extraction* and existing classifications in the *topic extraction*.

The Information Cloud provides methods, to calculate a closure called *TagCluster* of any tag, by following the outbound feature relations in a breadth-first search⁵ approach. It is inspired by a technique called word associations, which is also used in the *Edinburgh Word Association Thesaurus*⁶, that might serve well for ConTag

⁵http://en.wikipedia.org/wiki/Breadth-first_search

⁶<http://www.eat.rl.ac.uk>

in future versions.

The Information Cloud can be serialized in RDF/XML using the vocabulary of Figure 5.3.

5.4 Generate ontology alignments

ConTag tries to generate tag proposals based on two models. The Personal Information Model (PIMO) holds concepts and relations of the user's knowledge space. The extracted topic map (TM) constitutes the second model and contains all topics occurring in the grounding text resource. Without loss of generality it can be assumed, that the PIMO contains more entries than the topic map. Additionally, the PIMO may be structured with multiple deeper hierarchies, whether the topic map is rather flat structured (In the actual version of ConTag, only hierarchies of depth one exists).

In general, the retrieval of correspondences between TM and PIMO can be stated as follows:

Let $t \in PIMO$ be a concept, that is stored in the *PIMO*, and $c \in TM$ be a topic occurring in the Topic Map *TM*. An alignment generator $a : TM \rightarrow PIMO$ creates tuple relations $r := (t, c)$. The set containing all computed relations r in an alignment is called R . Additionally, in order to rate different relations concerning significance, $q : R \rightarrow [0, 1]$ states a quality rate for each relation $r \in R$.

Four relevant and possible correspondences between PIMO and TM exist and are to be implemented, regarding the Use Cases in Chapter 4.

Equivalence Relations According to Use Case 1, the occurrence of existing PIMO concepts in the text resource is implemented by drawing equivalence relations between topics and concepts.

$$eq : InstanceTag \rightarrow PimoThing$$

$$eq : ClassTag \rightarrow PimoClass$$

e.g. A ClassTag labeled with "Person" extracted by the tagthe.net service is identified as an existing PimoClass called "Person" in the user's PIMO.

Classification Relations According to Use Case 2, the occurrence of new and yet unknown instances is implemented by using classification relations between

topics and concepts (here: InstanceTag and PimoClass).

$$cl : InstanceTag \rightarrow PimoClass$$

e.g. The InstanceTag labeled with "Peter" is a new instance of the existing PimoClass "Person" in the user's PIMO.

Superordinate Relations According to Use Case 3, the occurrence of new and yet unknown classes is implemented by using superordinate relations between topics and concepts (here: ClassTag and PimoClass).

$$so : ClassTag \rightarrow PimoClass$$

e.g. The ClassTag labeled with "Europe" is superordinated by a PimoClass called "Region".

Untyped Relations According to Use Case 4, simple relations between existing PIMO concepts and occurring topics is implemented by using untyped relations.

$$rel : InstanceTag \rightarrow PimoThing$$

e.g. The InstanceTag labeled with "Peter" is related to an existing PimoThing "Paul" in the user's PIMO.

The implementation considers these four relations, is adaptable to regard the actual contents in the PIMO and works always on the actual version.

Due to the vast amount of text annotations and text resources in a Personal Information Model, it was decided to take proved Document Classification techniques to create an alignment between the topic map and the PIMO.

5.5 Classification in a Personal Information Model

Document Classification approaches classify documents into a set of existing classes. Classes may either be described as documents. We call it a *Taxonomic Document Classification*, if all classes are managed inside a class hierarchy. There are many approaches of performing a Document Classification. ConTag uses a statistical approach based on Bayes Conditional Probabilities, called Nearest Neighbor Classification, which sorts each document into a class, which seem to be similar concerning contents:

$$P(i \in c | \vec{x}_i \wedge \vec{x}_c) := f(\vec{x}_i, \vec{x}_c)$$

This mathematical expression can be reformulated as follows: How likely is document i classified by class c , if \vec{x}_i is a feature vector of i and \vec{x}_c is a feature vector of

c ? A feature vector $\vec{x} := (x_1, \dots, x_n)$ contains $n > 0$ features, to describe the contents of a certain document in a computable manner. In a document classification store, each feature represents one relevant term, occurring in the certain document. Relevancy is mainly expressed using the formula $tf \times idf$, which means the *term frequency* (occurrences of a certain term in a document) is multiplied with the *inverse document frequency* (normalized ratio of documents containing the term). A classification index *index* of a document classification store contains the function $f(\vec{x}_i, \vec{x}_c)$ and a threshold vector \vec{t} to decide whether a classification is significant or not ($index(f(\vec{x}_i, \vec{x}_c), \vec{t})$). Concerning classification indices in a PIM, each concept states as a class c . The computation of a concept's feature vector is based on all documents the concept is attached with, which means the concept and all related resources are concerned to be one document in terms of document classification methods.

Adding a taxonomy based on hierarchic relationships like (subclassOf, instanceof or partOf) into the classification index provides performance and quality enhancements. A taxonomic classification index uses inheritance approaches in refining the specification of feature vectors. If $C1$ is a superconcept of $C2$ concerning the used taxonomy, then the feature vector of $C1$ contains all features of $C2$: $\vec{x}_{C2} \subseteq \vec{x}_{C1}$. This approach permits the reduction of possible classifications, by reducing comparisons of all sub concepts of a concept C , if the resulting degree of f using a technique s doesn't reach the specified threshold for this technique in the classification index ($f_s(\vec{x}_i, \vec{x}_C) < \vec{t}[s]$).

The computation of a topic's feature vector is done using the TagCluster computation mentioned in the specification of Information Clouds in Section 5.3. Each feature corresponds to one relation in the TagCluster.

5.5.1 Comparing topics and concepts

The comparison of topics and concepts is done in a ConTag specific implementation of f using the following signature: $f_{ConTag}((\vec{x}_i, label_i), (\vec{x}_c, label_c))$. Therefore the classifier f is enhanced with the labels of instance i and class c . The threshold vector \vec{t} used in the index contains two threshold parameters, namely the *label similarity bound*, narrowing the similarity computation of both labels and the *content similarity bound*, narrowing the similarity computation between two feature vectors. Both thresholds are managed in ConTag's runtime environment. The result of f is a vector containing two values (label similarity and content similarity). $f_{ConTag}((\vec{x}_i, label_i), (\vec{x}_c, label_c)) \rightarrow (sim_x, sim_l)$ and $0.0 < sim_x < 1.0, 0.0 < sim_l < 1.0$. It is required, that the resulting similarities have to reach uniform values between zero and one.

The computation of a label similarity between $sim : (label_i, label_c)$ can be based

on different metrics. Therefore, the implementation of ConTag uses an open source library of similarity metrics called *SimMetrics*⁷, written by Sam Chapman. This enables an easy switch between existing similarity metrics. By default the ConTag runtime environment starts using a *Dice Coefficient* based metric.

The computation of a content similarity is a sophisticated problem. The Topic Extraction service is designed to collect as much information about a topic as possible. The goal is to retrieve a large feature vector to provide significant classification results. The resulting problem is the normalization of returned similarity values to reach only a real interval spanned between zero and one. It is not suitable to normalize using one of the feature vector's sizes, because both vector sizes may differ tremendously without any correlation.

Let $m(\vec{x}_i, \vec{x}_c) \rightarrow N_0^+$ be a function to compute similar matches between both feature vectors. Due to the fact, that the size of a feature vector is not delimited, it can be stated, that increasing the similarity and size between both vectors results in an diverge rise of matches. The following transformation $norm : N_0^+ \rightarrow [0.0, 1.0]$ maps a sequence of real values in $[0.0, \infty]$ into the desired interval $[0.0, 1.0]$ of real numbers in a uniform way: $norm(m) := 1 - \frac{1}{\log_{10}(m+1)+1}$. It is used in the Alignment Creation service. Matches in feature vectors are computed the upper described label similarity.

5.5.2 Creating the four correspondences

The use of a taxonomic nearest neighbor classification approach enabled us to express four rules, by using the types of o be classified topics t , the resulting classified PIMO concepts c and their similarity values (sim_t, sim_c) to build four correspondences as described in Section 5.4:

Equivalence Relations Iff t is type of `InstanceTag`, c is type of `PimoThing`, $sim_t \geq t_l$ and $sim_c \geq t_c$, then t and c are defined to be equivalent. That means a `PimoThing` c occurs in the text resource.

Iff t is type of `ClassTag`, c is type of `PimoClass`, $sim_t \geq t_l$ and $sim_c \geq t_c$, then t and c are defined to be equivalent. That means a `PimoClass` c occurs in the text resource.

Classification Relations Iff t is type of `InstanceTag`, c is type of `PimoClass` and $sim_c \geq t_c$, then t is defined to be a new possible instance of c . That means a yet not existing `PimoThing` t of an existing type c occurs in the text resource.

⁷<http://www.dcs.shef.ac.uk/~sam/stringmetrics.html>

Superordinate Relations Iff t is type of `ClassTag`, c is type of `PimoClass`, $sim_l < t_l$ and $sim_c \geq t_c$, then t is defined to be a new possible subclass of c . That means a yet not existing `PimoClass` t of an existing type c occurs in the text resource.

Untyped Relations Iff t is type of `InstanceTag`, c is type of `PimoThing`, $sim_l < t_l$ and $sim_c \geq t_c$, then t and c are defined to be loosely related. That means a new yet unknown `PimoThing` t occurs in the text resource and is related to an existing `PimoThing` c .

The resulting alignment is described in the *Phaselibs Alignment Ontology*⁸ (see Fig. 5.4), which is defined in RDFS. In spite of expressing alignments, this ontology enables the user to set acknowledgments for each relation in order to accept or reject a certain proposal. A relation's confidence is calculated by averaging the content and label similarity values.

The alignment ontology contains relations, that correspond to `ConTag`'s tag proposals, which means they explain how to introduce topics occurring in a text resource into the user's PIMO. `ConTag`'s GUI service is designed to visualize this alignment for further user evaluation.

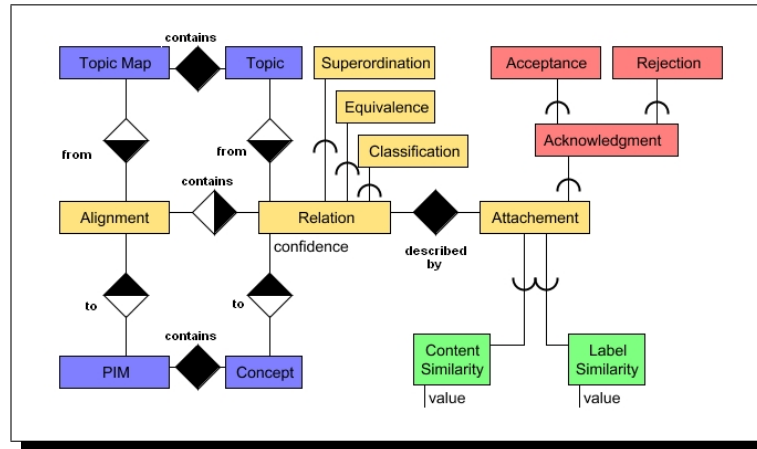




Figure 5.4: Alignment Ontology

⁸<http://phaselibs.opendfki.de/wiki/AlignmentOntology>

5.6 Displaying tag proposals

In order to obtain the usability and consuetude of the underlying Gnowsis or PIMO system, it was decided to use an UI design, that is abutted to a Gnowsis internal user interface called *Miniquire* (see Fig. 3.3). The screen shot in Figure 1.1 displays ConTag's GUI in detail. As described formerly, a button labeled "Tag It!" is placed on the browser's bookmark pane. Pressing this button enables the user to decide whether to let ConTag process the current web page or not. After the button is pressed, ConTag's GUI service opens a new HTML-Frameset, to show a new sidebar on the left side (implemented as HTML-Frame) of the page display, which contains the computed tag proposals in a hierarchical design.

The GUI consists of four major areas (see Fig. 5.6). The top area is titled "Tag Suggestions" and contains three of four proposal types: Equality, Classification and Superordination. Each relation type is declared with a certain icon, which is explained shortly in Fig. 5.5.

The user may discharge several tag proposals out of the current view, by pressing the red button . To gain more information about a tag proposal, the blue button labeled with a question mark  provides a small popup frame. It contains detailed descriptions about the relations's confidence, a list of the tag's definitions, its feature vector, etc. Regarding the "Oh Yeah Button", by Tim Berners-Lee [BL97] this popup contains an explanation possibility to let the user research more about a certain tag proposal.

The next area from above is titled with "Tag Relation Suggestions" and contains all untyped relations realizing the last Use Case 4.

Due to the fact, that the user interface may list a large set of proposals with existing confidence values, the user gets the possibility to filter only those proposals reaching a certain threshold. Therefore, the UI's bottom panel contains a text field to set certain threshold values between zero and one. A button labeled "adjust" enables to execute this confidence filter.

Every proposal in this UI is attached with a small HTML-Select Box. Throughout this widget, the user is able to accept proposals by selecting their select boxes. Each selection attaches an *Acceptance* entity, defined in the Phaselibs Alignment Ontology, to mark the selected relation as valid and user accepted.

Generally, after the user has finished his evaluation of the tag proposals, he presses the "submit" button to start the Alignment Execution service.

- This PIMO-instance icon characterizes an *Equality Relation* and declares, that an existing thing in the PIMO occurs in the text resource.
- This PIMO-class icon characterizes a *Classification Relation* and expresses, that an existing class in the PIMO occurs in the text resource. Naturally, either a PIMO Instance or a ConTag Instance is classified by this icon. (e.g. Italy in Fig. 5.6)
- This ConTag-instance icon takes part is part of a *Classification Relation* and declares, that a new thing is proposed to be inserted into the PIMO under a certain class. (e.g. european union in Fig. 5.6)
- This ConTag-class icon describes a *Superordination Relation*, that a new class is proposed to be inserted into the PIMO. (e.g. europe in Fig. 5.6)
- This PIMO-relation icon describes an untyped *Relation* between a new ConTag instance and an existing PIMO instance.

Figure 5.5: Icons



Figure 5.6: Sidebar

5.7 Alignment execution

The execution of user rated tag proposals is implemented within a simple algorithm in ConTag's Alignment Execution service. At first it executes all existing *Superordination Relations*, to assure that every new created PIMO-thing can be created with an existing PIMO-class. After creating the new PIMO-classes, all remainder relation of type *Classification* and *Equivalence Relations* are free for execution. For each relation execution being involved with PIMO-things, the respective thing is attached with a PIMO-property called "occurrence" to link it with the currently processed web page in an RDF triple Statement (thing, occurrence, resource). The last execution step processes all *Untyped Relations* to connect the involved PIMO-things with an PIMO-relation called "related". After finishing the Alignment Execution, the user is able to review and manage his tag collection in his PIMO.

5.8 Web 2.0

ConTag's design structure and information flow is based on the existence of Web 2.0 services. Like Tim O'Reilly introduced it in [O'R05], these Web 2.0 functionalities afford a rapid solution development of problems like Content Analysis or Dictionary Lookups. It is possible to use these services quickly and without the use of heavy frameworks to process their output data and request other services for further researches. This Web 2.0 Choreography makes it possible to implement a complex system like ConTag within five man months.

5.8.1 Web 2.0 services

The following list of used Web 2.0 services and service providers reports about their functionalities and adoptions. During the development of ConTag several other Web 2.0 were inspected but not used in the final implementation. They are listed in the Appendix A.2.

Yahoo!

Yahoo! is one of the biggest yet existing Web 2.0 service providers. On <http://developer.yahoo.com/search> it hosts several Web 2.0 services, like the used Term Extraction Service, to extract relevant and significant phrases out of a text. Processing the general example in Fig. 5.1 with this service leads to the following result of extracted phrases:

```
rome; acme; european union; unfortunately;  
business members; infrastructures; fulfill;  
settle; sophisticated; acquisition;  
conclusion; leads; new branch office
```

Yahoo's services are accessible via REST queries. (<http://www.yahoo.com>)

Google

Notwithstanding Google is one of the precursor of Web 2.0 technologies, the amount of provided Web 2.0 services is rather low. During the development of ConTag one service was in terms of interests, which is unlikely not a Web 2.0 service. The Google Glossary Service researches for web definitions written in natural language in the WWW concerning a certain search phrase. Unfortunately Google does not offer any open interface to access this service, whereby the service classification by Web 2.0 fails. (<http://www.google.com>)

tagthe.net

Tagthe.net is a Content Analysis Service like the above Term Extraction service. It is hosted and maintained by Knallgrau New Media Solution GmbH. However its results of processed texts contain a simple content classification. Extracted phrases are instantly classified to meet classes like: person, topic, metatopic, location, language, etc. Processing the general example in Fig. 5.1 with tagthe.net leads to the following result of extracted classifications:

```
topic acquisition; business; society; Corp; Branch;  
East; ACME; office; amount
```

```
person Peter; Paul
```

```
location Italy; Rome
```

```
language english
```

Tagthe.net is accessible via REST queries. (<http://www.tagthe.net>)

DICT

The Dictionary Server Protocol (DICT) [FM97] is a TCP transaction based Internet protocol that allows a client to access dictionary definitions from a set of natural

language web dictionary databases. It is an advancement of the proprietary Webster protocol, to provide a standardized access to multiple web dictionaries.

ConTag queries the dataset of WordNet ⁹, a semantic lexicon for the English language and the Moby Thesaurus II, a list of word associations by using DICT. (<http://www.dict.org>)

Ontok Wikipedia API

Ontok Web Services provide a REST based Web 2.0 service to access the database of the Wikipedia Encyclopedia, which is a collaborative and open web dictionary, available for different languages.

Ontok's Wikipedia Web Service provides several access possibilities to Wikipedia's articles. ConTag uses two of them, namely *GetWikipediaPageLinks* to retrieve outbound links to other relevant articles and *GetWikipediaReversePageLinks* to get inbound links from other relevant articles. (<http://www.ontok.com/wiki/index.php/Wikipedia>)

DefTag

DefTag is a self written Web 2.0 service. It performs a hypernym extraction based on definitions written in the English language. DefTag offers a REST based interface to use its computations. The following example expresses the usability of DefTag for ConTag:

Kaiserslautern is a wonderful city in Rhineland Palatine.

DefTag extracts a list of proposed hypernyms:

```
city in Rhineland Palatine; wonderful city
in Rhineland Palatine; Rhineland; Palatine;
city
```

The implementation of ConTag is based on Aristoteles' lemma *Definitio fit per genus proximum et differentiam specificam.*, which is already mentioned and explained in Section 3.3. It uses natural language processing (NLP) techniques like Part-of-Speech-Tagging to identify terms as a part of speech (noun, adjective, etc.) and sentence detection to extract existing sentence structures, which are both based on corpus based methods.

DefTag's algorithm can be described in one sentence by explaining its basic heuristic:

⁹<http://wordnet.princeton.edu>

In a definition, based on Aristoteles' lemma and in its first occurring sentence, a list of hypernym nouns is likely being placed after the first verb.

To implement this heuristic, DefTag uses open source NLP techniques and existing projects published at OpenNLP ¹⁰.

5.8.2 Syndication in Web 2.0

The dream of weaving a Web 2.0 is heavily based on collaboration. Tim O'Reilly called it Syndication and explained, that small Web Services provide lightweight solutions for tiny and encapsulated problems. Connecting these micro services might result in a more powerful macro service. ConTag as a macro service is good example, representing this procedure.

The scenario of intercommunication between web services is as old as their existence. However, the difference between the traditional web services (mainly based on technologies like: WSDL ¹¹, UDDI ¹² and SOAP ¹³ [MG03]) and Web 2.0 services is the degree of being well defined concerning interface and communication descriptions. Web 2.0 services, based on REST interfaces do not provide methods, like automatic proxy code generation after parsing a WSDL based interface description. But they allow not only computer science experts to understand and use the, existing Web Services. According to the missing interface definitions, existing web service choreography and composition languages like BPEL do not fit into an open and lightweight Web 2.0 example like ConTag.

Apart from this, quality attributes of Web 2.0 servers like availability or response times do not tend to be dependable. Programming with Web 2.0 Services means to use strict timeouts and existing alternative service calls to provide usable and definite return values. Due to this inconsistencies stable and repeatable application calls are not possible in ConTag.

¹⁰<http://opennlp.sourceforge.net>

¹¹Web Service Description Language [EC01]

¹²Universal Description, Discovery and Integration [OAS04]

¹³Simple Object Access Protocol

Chapter 6

Evaluation

”It doesn’t matter how beautiful your theory is, it doesn’t matter how smart you are. If it doesn’t agree with experiment, it’s wrong.”

- Richard Feynman -

In order to state the quality of ConTag’s generated tag proposals, it is necessary to perform result evaluations resulting in different types of quality ratios. This chapter describes the evaluation of ConTag’s Tagging Process. It shortly summarizes four used test scenarios in Chapter 6.3. The evaluation of ConTag’s *Topic Extraction* is briefly described in Chapter 6.4. Chapter 6.5 summarizes the evaluation of the resulting final tag proposals after the *Alignment Creation*.

6.1 Precision and Recall

The Information Retrieval domains offers two main quality measures called *Precision* and *Recall* [Rij79]. Whereas Precision defines the accuracy of calculated result values, Recall defines the completeness. Two sets are defined to specify these quality ratios. The first set contains several *test corpora* (TC). A test corpus contains a list of test resources $r \in TC$ which are used as input values of the to be tested computation. The second set is called *Ground Truth* (GT_r). It defines the desired output values after the computation for each resource $r \in TC$ in a test corpus. With a test corpus and the corresponding Ground Truth, it is possible to define two types of rated results:

Found Results (F_r) The set called Found Results contains all computed results for a resource r in a test corpus.

Relevant Results (R_r) The set called Relevant Results contains those computed results, that are also listed in Ground Truth of the corresponding resource r .
 $(R_r \in F_r \wedge R_r \in GT_r)$

The following definitions of the quality measurements Precision and Recall are used to express the quality of ConTag's result values, namely the topic and tag proposals occurring in step two and three of ConTag's Tagging Process. ¹

$$Precision := \frac{F \cap R}{F} \quad (6.1)$$

$$Recall := \frac{F \cap R}{R} \quad (6.2)$$

6.2 Quality assurance in test environments

High quality evaluation in Software Engineering depends on the quality of the underlying test corpora and the hence developed Ground Truth. Therefore, three evaluation quality aspects are conscientiously emphasized during the validation of ConTag to generate meaningful quality ratios, namely:

1. *High coverage* in the test corpora, concerning possible input data, which were specified in Chapter 3.
2. *Objectivity*, concerning the relation between the internal functionalities and the evaluation of test data in Ground Truth.
3. *Strong correlation* between each specified use case and existing test runs. Every use case should be validated by at least one test scenario.

These quality aspects are realized as follows:

1. The existence of a *high coverage* is realized by defining four different test corpora. Each test corpus contains possible input data out of different domains. Every resource is a document written in the natural English language. Each test corpus in ConTag's validation contains different characteristics and distribution of topics, that are listed further in Section 6.3. The evaluation started with test data, based on several hundred news entries, provided by the news agency Reuters. Due to high expenditure of time to spent creating Ground Truths, it was decided to use only ten represents out of each test corpus.

¹For a better readability the set occurrences (F, R, I) are redefined as cardinalities $(|F_r|, |R_r|, |I_r|)$.

2. The four resulting ground truth sets were created by an independent person, who was not involved in the development of ConTag. The person was given a static Test PIMO (see Appendix A.3) containing already several classes and instances. The general conceptual formulation was to note all topics occurring in each resource in each test corpus to be inserted into the PIMO, by using three different approaches. Each approach corresponds to one Use Case listed in Chapter 4. The first approach concerns Use Case 1 and pretends to note those topic, that already exist in the PIMO. The second approach tends to note all not yet existing topics in the PIMO, that occur in a resource anyhow to be inserted into the PIMO. The last approach aims to classify existing or new extracted PIMO instances and new PIMO classes.

It is not possible to eliminate the subjectivity in creating a ground truth for ConTag. It can be supposed, that every person possesses his own classification of things, based on different social economic backgrounds. But nonetheless, ConTag's Ground Truth is not influenced by any knowledge of the existing computation.

For this reason, the validation of ConTag focuses on the analysis and generation of Recall ratios, because proposals that might be relevant for one person were concerned to be irrelevant for ConTag's Ground Truth creator.

Use Case 4 was not validated, because the task of creating untyped relations between proposed tags and existing things in the PIMO was formulated too late in ConTag's development process and is therefore concerned to be a nice side effect. The validation of ConTag's generation of tag proposals is separated into two major divisions, corresponding to two steps in ConTag's tagging process (see Fig. 3.2), namely the *Topic Extraction* which is done in step three and the *Alignment Generation* performed in step four.

6.3 Test corpora

ConTag's validation is based on four different test corpora to gain as much coverage of possible input resources as possible. An additional importance of this high coverage is, in spite of the quality analysis of Contag's results, but in terms of software development, the detection of existing programming failures. The following sections describe every test corpus providing necessary information to interpret later occurring test results. These descriptions do not correspond to one of the Use Case specific Ground Truths, but describe their unification of contained classification proposals to provide an overall and characteristic summary. More details about the test corpora's contents can be found in the Appendix A.4.

6.3.1 Reuters' News Corpus

The *Reuters Group plc* is best known as a news service that provides news reports from around the world to newspapers and broadcasters.² The Reuters' News Corpus contains several news tickers, that are marked-up in a XML dialect called NewsML. Each ticker contains several short news entries of about one to three sentences. Each news ticker is inherently classified with a set of topics describing the occurring countries or basic topic classes like politics, economy, etc. These classifications have been adopted in the GroundTruth and were extended with additional topics based on the existing test PIMO. Figure 6.1 shows, that the Reuters New Corpus focuses on *topics*, *locations* and *organizations*. Nevertheless, the other class distributions are particular frequent compared to other test corpora.

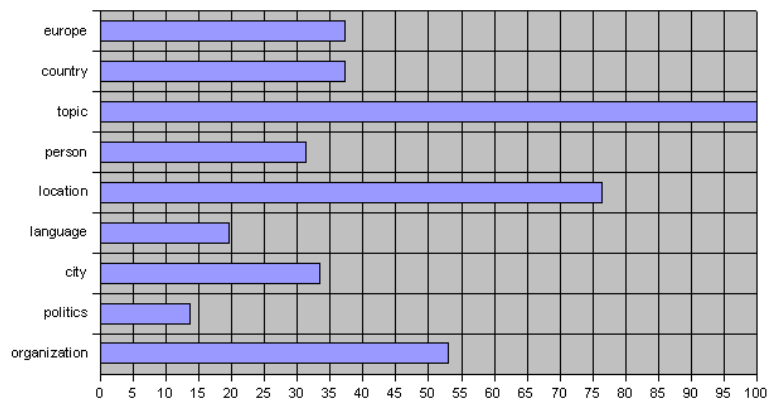


Figure 6.1: Distribution of desired classifications in the *Reuters' News Corpus*

6.3.2 Index Site Corpus

The Index Site Corpus was generated manually. It contains welcome pages of several organization and project descriptions. Figure 6.2 shows, that the majority of classifications occurring in inviting index pages map to *topics*. In spite of this, there is a significant emergence of *java* and *language* classes which is explainable by the high usage of index sites concerning the programming language Java .

²<http://en.wikipedia.org/wiki/Reuters>

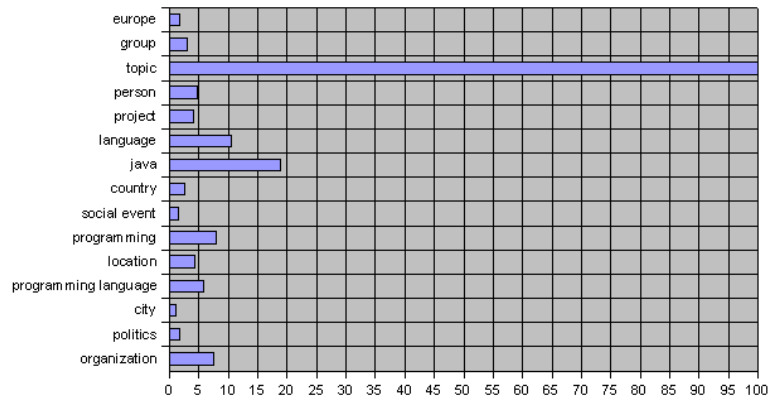


Figure 6.2: Distribution of desired classifications in the *Index Sites Corpus*

6.3.3 Wikipedia Concept Corpus

The Wikipedia Concept Corpus was created manually. It concerns high-level scientific topics like mathematics, computer science, linguistics, etc.. Throughout the introducing nature of these articles, the Wikipedia Concept Corpus, which is described in Figure 6.3, contains a huge amount of *topics* and *persons*.

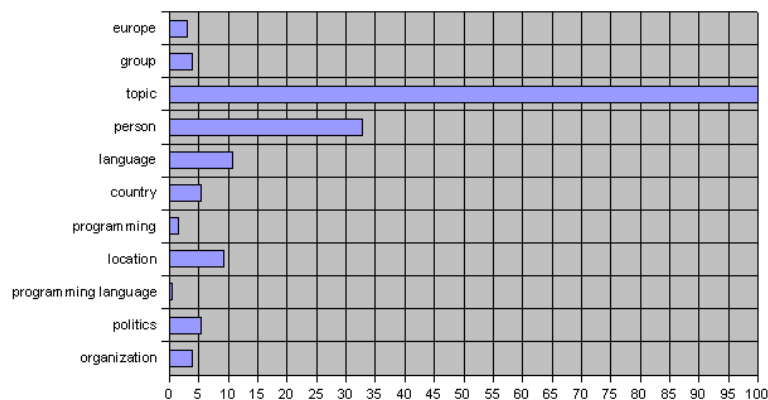


Figure 6.3: Distribution of desired classifications in the *Wikipedia Concept Corpus*

6.3.4 Wikisource Historical Corpus

The Wikisource Historical Corpus contains historical documents, written in or translated into the English language. It contains a high amount of *persons*, *topics* and *locations*, which is expressed in Figure 6.4

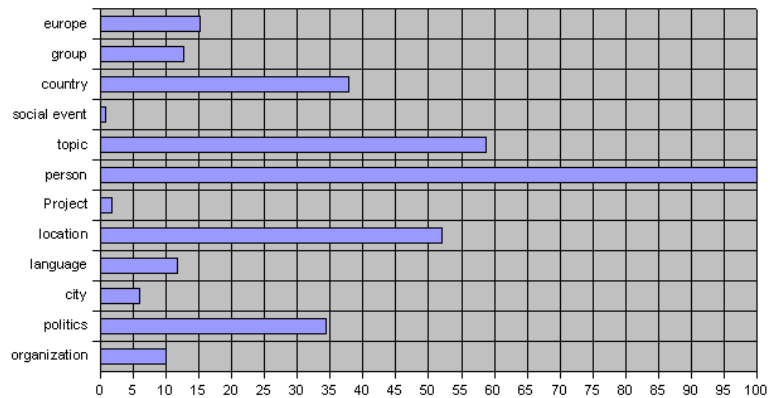


Figure 6.4: Distribution of desired classifications in the *Wikisource Historical Corpus*

6.4 Topic extraction evaluation

The Topic Extraction was validated by doing a Recall Analysis to figure out the amount of relevant topics corresponding to noted topics in the Ground Truth. The analysis iterated over all test corpora. In this validation the consideration of existing Use Cases is set aside, because the Topic Extraction is performed at the beginning of the Tagging Process and extracts as much topics as possible to provide a base for the next step, the Alignment Generation.

This test run iterates over four specified configuration settings:

none This configuration queries all possible Web 2.0 services. It is the normal setting of ConTag's runtime environment.

tagthenet The configuration labeled as tagthenet omits the content analysis service called *TagThe.net*, which provides a basic classification of extracted topics.

yahoo This configuration setting omits the content analysis service called *Term Extraction* provided by Yahoo!, which returns occurring phrases.

deftag Here, the definition tagging service called `DefTag` is left out, which returns a list of hypernyms for a given topic definition.

In order to provide the possibility to compare generated test results using these configurations, all remaining runtime parameters are constant. All web service calls in the Topic Extraction step are canceled after a span of 30 seconds counted after sending the request. The Alignment Generation is configured to use the following Similarity configurations, which are based on experience values to provide best results.

```
Label Similarity := 0.5
Content Similarity := 0.23
```

A further validation of optimal similarity parameters is described in Subsection 6.5.2. The diagrams in the figures printed on the following pages: 6.5, 6.6, 6.7 and 6.8 show Recall ratios of generated and already classified topics. Every bar on each PIMO class corresponds to a Web Service which is left out during the current test run. PIMO classes in the Test PIMO without any topic occurrences in a test run are not printed, to provide a better readability. It is possible to identify them by comparing the printed test result with the corresponding desired Ground Truth classification. It may be seen, that the Web 2.0 service `Tagthe.net` exerts a high influence on the quality of generated topic maps in step two of the Tagging Process.

Generally, each test run not using `Tagthe.net` results in a worse classification than using it!

Another less significant but important evaluation result is, that Yahoo's Term Extraction service provides a significant additional amount of overall topics. The hypernym extraction service `DefTag` does not influence the quality of `ConTag`'s result in the Topic Extraction step. So don't the other Web 2.0 services which are therefore not displayed in these diagrams. A final quality feature concerning `Tagthe.net` is the fact, that it is able to identify the correct language in most cases. It can be supposed, that the high influence of `Tagthe.net` is continued in the successive steps of `ConTag`'s Tagging Process.

The analysis of the degrees labeled as *total* reveals, that only about 40 % of desired topics were found during the Topic Extraction step.

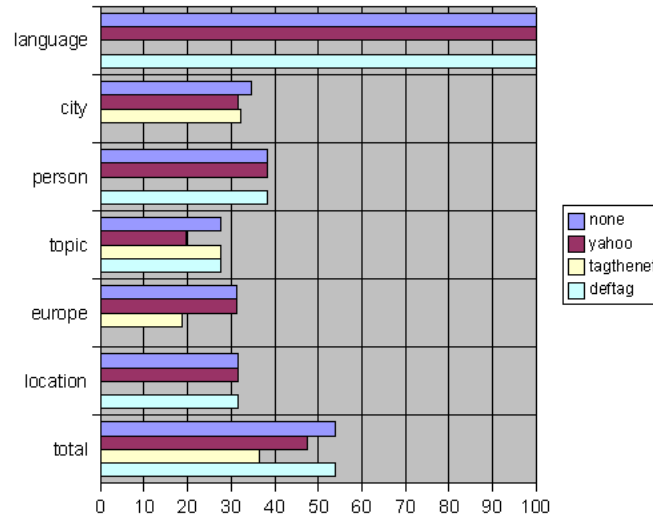


Figure 6.5: Distribution of relevant topic classifications found in the *Reuters News Corpus*

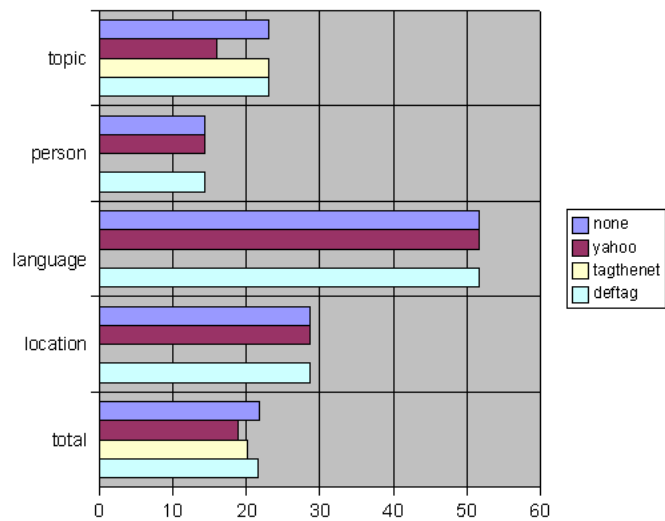


Figure 6.6: Distribution of relevant topic classifications found in the *Index Sites Corpus*

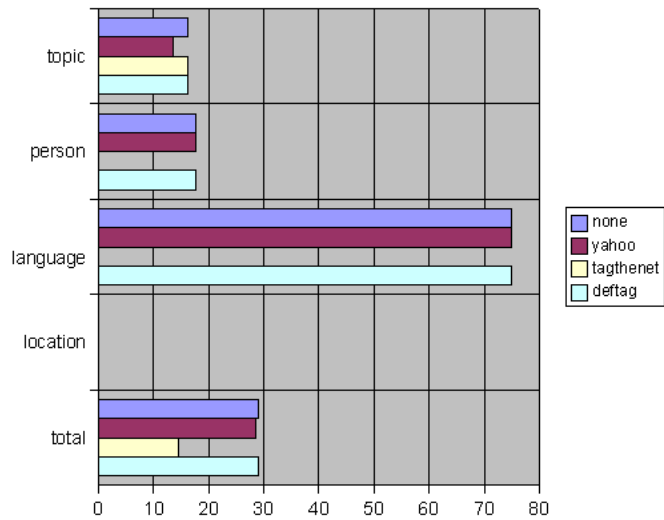


Figure 6.7: Distribution of relevant topic classifications found in the *Wikipedia Concept Corpus*

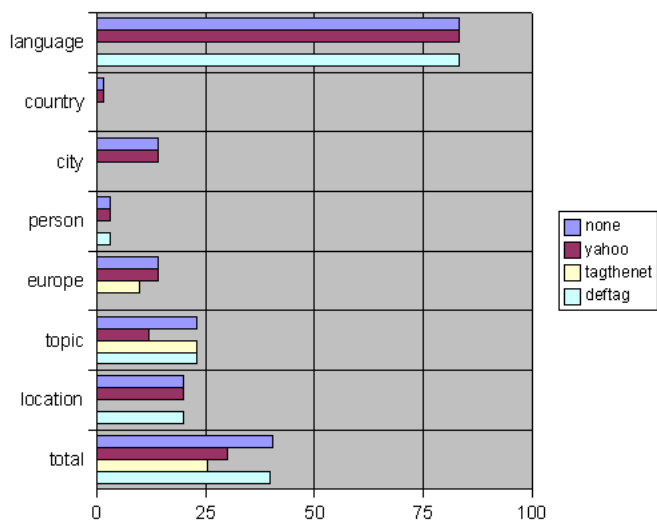


Figure 6.8: Distribution of relevant topic classifications found in the *Wikisource Historical Corpus*

6.5 Alignment generation evaluation

The Alignment Generation step was evaluated by performing several analytic methods. The *Service Analysis* in Subsection 6.5.1 calculates Recall ratios regarding the first three Use Cases. The *Similarity Analysis* in Subsection 6.5.2 computes quotes about Precision and Recall ratios in tag proposals (concerning the Ground Truth) by using different similarity configurations. This leads to an approximation to optimize ConTag's similarity threshold parameters.

6.5.1 Service analysis

During the Service Analysis each evaluation run was iterated over seven sets of different configurations in order to inspect the influence of omitting a certain Web 2.0 service as information source. The following list describes all configurations to analyse the influence of different web service usages on the quality of ConTag's return values.

none This configuration queries all possible Web 2.0 services, described in Section 5.8.1. It is the normal setting of ConTag's runtime environment.

tagthenet The configuration labeled as tagthenet leaves out the content analysis service called *TagThe.net*, which provides a basic classification of extracted topics.

yahoo This configuration setting misses the content analysis service called *Term Extraction* provided by Yahoo!, which returns occurring phrases.

deftag Here, the definition tagging service called *DefTag* is left out, which returns a list of hypernyms for a given topic definition.

wordnet This configuration omits the web dictionary service called *Wordnet*, which returns a list of existing definitions for a certain topic.

wikipedia The configuration labeled as wikipedia omits two dictionary services accessing the web encyclopedia called *Wikipedia*, that return a list of semantically related concept labels for one term. Both services provided in this API have been left out in the same configuration, to emphasize the semantic similarity of their result values, based on an equal source of data sets.

moby The web thesaurus called *Moby Thesaurus II* is omitted in this configuration. which returns a list of word associations for a given term.

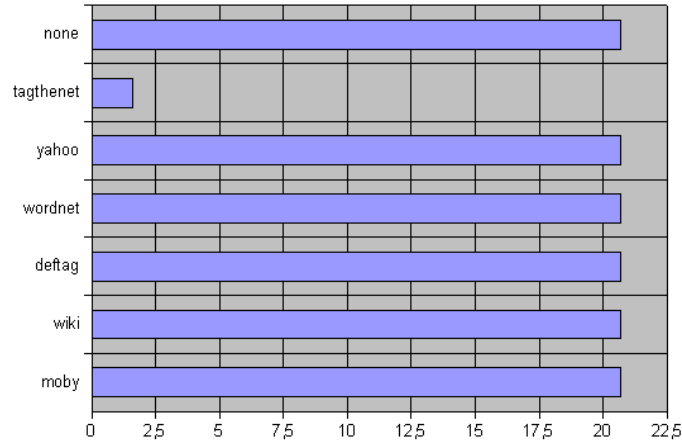


Figure 6.9: Use Case 1: Distribution of identified existing topics in the *Wikisource Historical Corpus*

Use Case 1

The evaluation of the first Use Case figures out the amount of extracted and proposed tags, that already exist in the Test PIMO. The below figures 6.9, 6.10, 6.11 and 6.12 clarify and underline the importance of Tagthe.net concerning the quality of tag proposals. As in the validation of the Topic Extraction, every bar on each class corresponds to a Web Service which is left out during this test run. Omitting the service Tagthe.net at least halves the amount of identifiable topics in all test corpora. The key essence of evaluating Use Case 1 is, that primarily Tagthe.net and secondarily Yahoo's Term Extraction provide relevant information. These dependencies are inherited from the Topic Extraction Step, that already possesses a high dependency on Tagthe.net and Yahoo's Term Extraction. Omitting other services does not vary any test results in positive or negative ways.

Use Case 2

The evaluation of Use Case 2 in a service analysis clarifies, how much tag proposals, that do not exist in the PIMO, are correctly classified, according to the Ground Truth. Again the impact of used web services sheds light on their quality ratios in ConTag. These impacts are clarified for each test corpus iteration in figures: 6.13, 6.14, 6.15 and 6.16. Every bar on each class corresponds to a web service which is omitted during the specific test run.

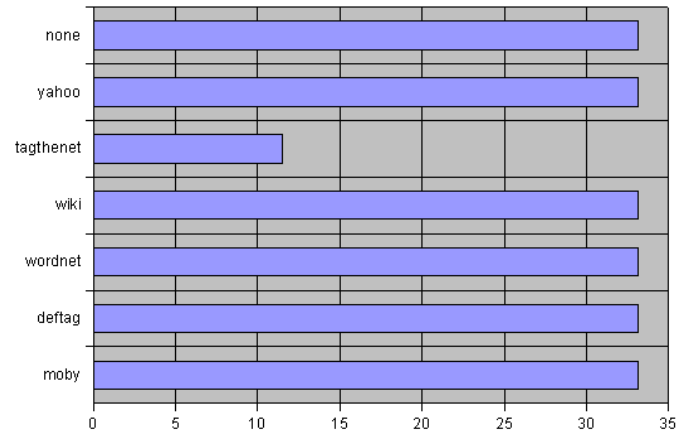


Figure 6.10: Use Case 1: Distribution of identified existing topics in the *Wikipedia Concept Corpus*

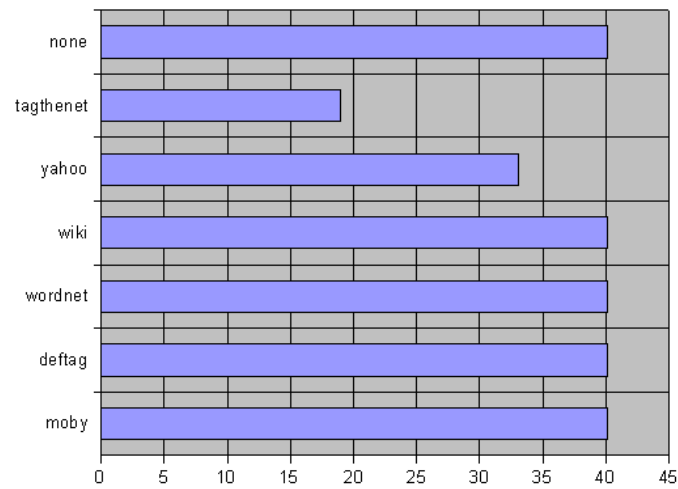


Figure 6.11: Use Case 1: Distribution of identified existing topics in the *Reuters' News Corpus*

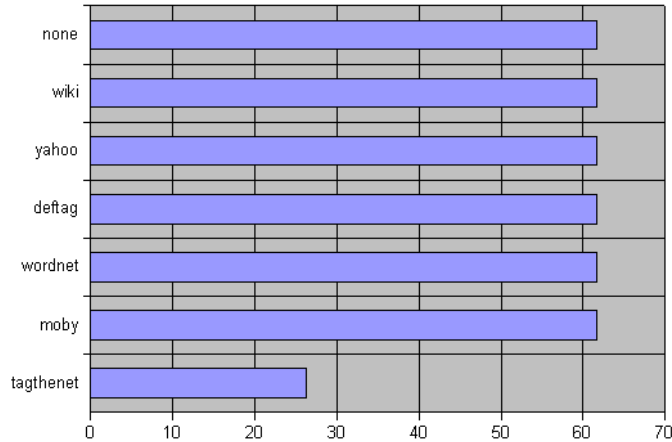


Figure 6.12: Use Case 1: Distribution of identified existing topics in the *Index Sites Corpus*

The inherited dependency on Tagthe.net and Yahoo continue in Use Case 2. Additionally, the analysis of Use Case 2 can reveal, that Tagthe.net highly influences classifications to certain classes. Especially the classification of *language* (see *Topic Extraction Validation* in Section 6.4) is highly based on Tagthe.net's usage. In spite of *language*, other classifications like *location*, *person* and *topic* also vary significantly when missing Tagthe.net. This phenomenon can be explained by inspecting the inherent classification of Tagthe.net, which already uses internal classes:

```
topic, metatopic, person, location,
language.
```

These Tagthe.net specific classes correspond to existing equally labeled classes in the PIMO and therefore influence the alignment generation.

Due to the usage of hierarchical classification approaches in the Alignment Generation step, this dependency on Tagthe.net is inherited to all concerning subclasses of language, person and location, like country or city.

Surprisingly, the usage of the web service provided by Ontok to access Wikipedia articles produces particularly better classification, which can be seen in e.g. Fig. 6.13 by analysing the PIMO classes country and its subclass location.

The usage of DefTag, Wordnet and Moby Thesaurus remains insignificantly towards an increase in quality of ConTag's tag proposals.

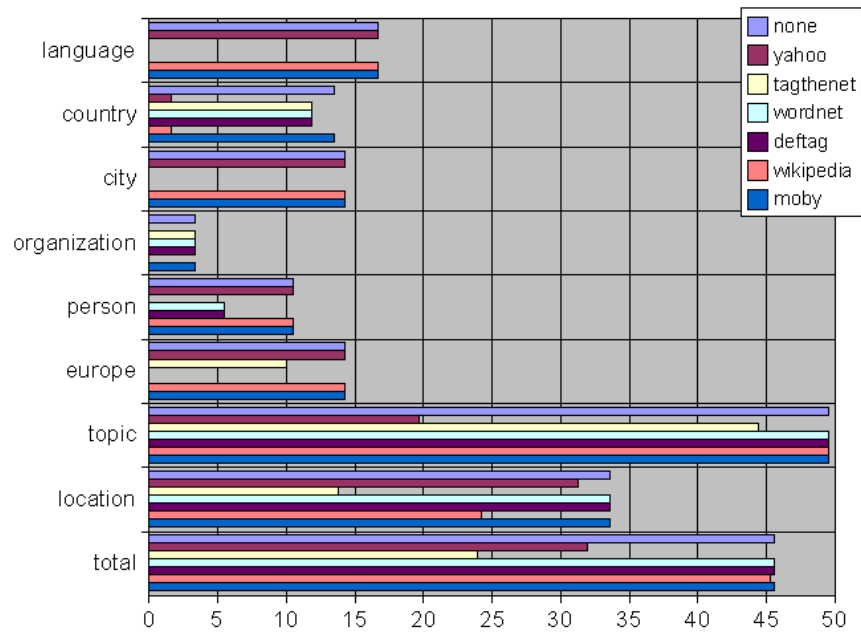


Figure 6.13: Use Case 2: Distribution of topic proposals that do not exist in the Test PIMO, but occur in *Wikisource's Historical Corpus*

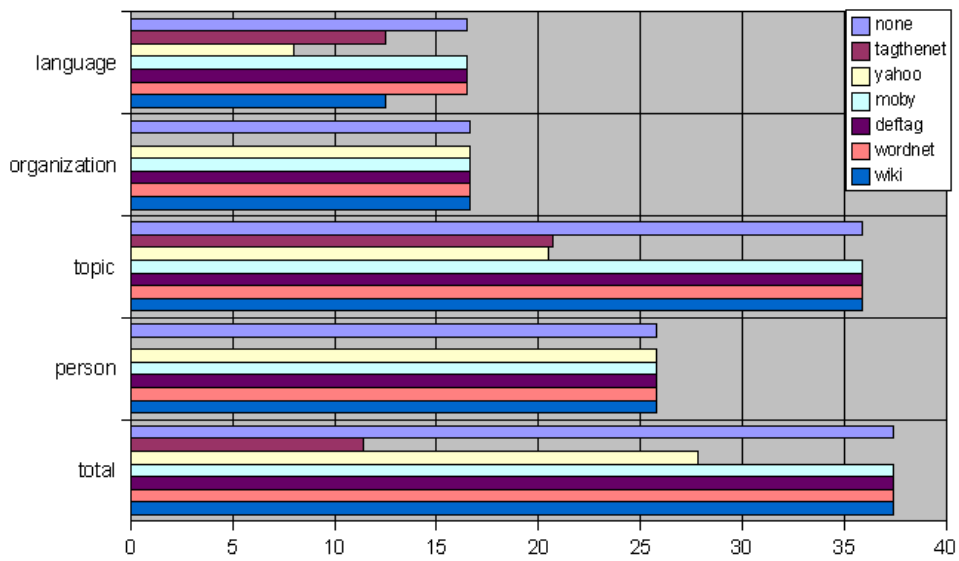


Figure 6.14: Use Case 2: Distribution of topic proposals that do not exist in the Test PIMO, but occur in *Wikipedia's Concept Corpus*

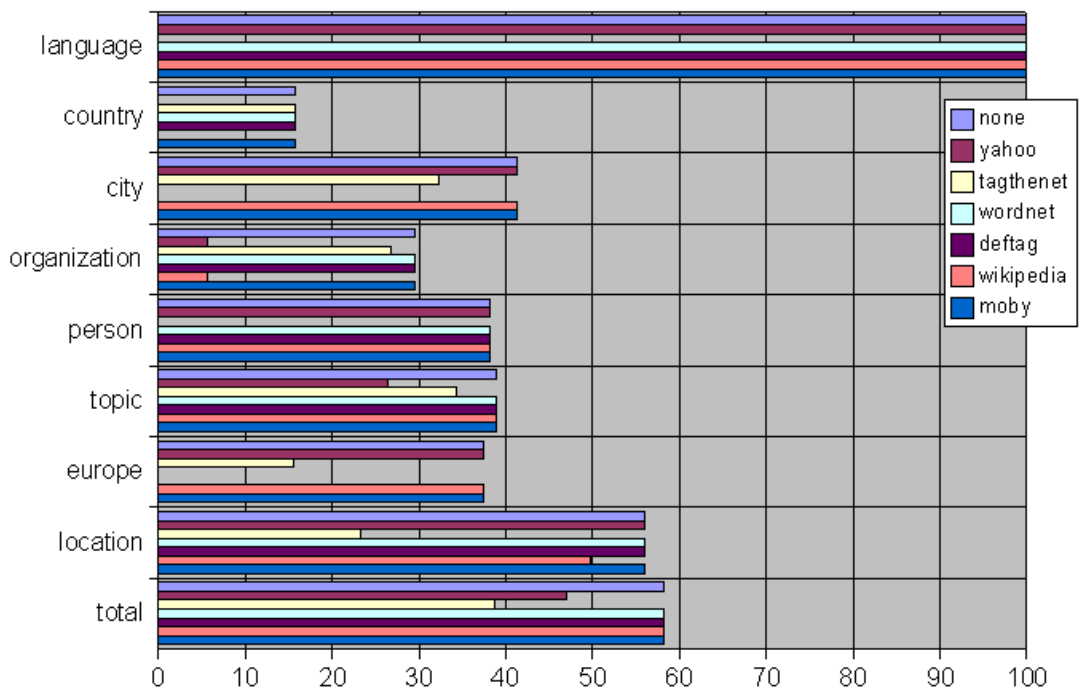


Figure 6.15: Use Case 2: Distribution of topic proposals that do not exist in the Test PIMO, but occur in *Reuters' News Corpus*

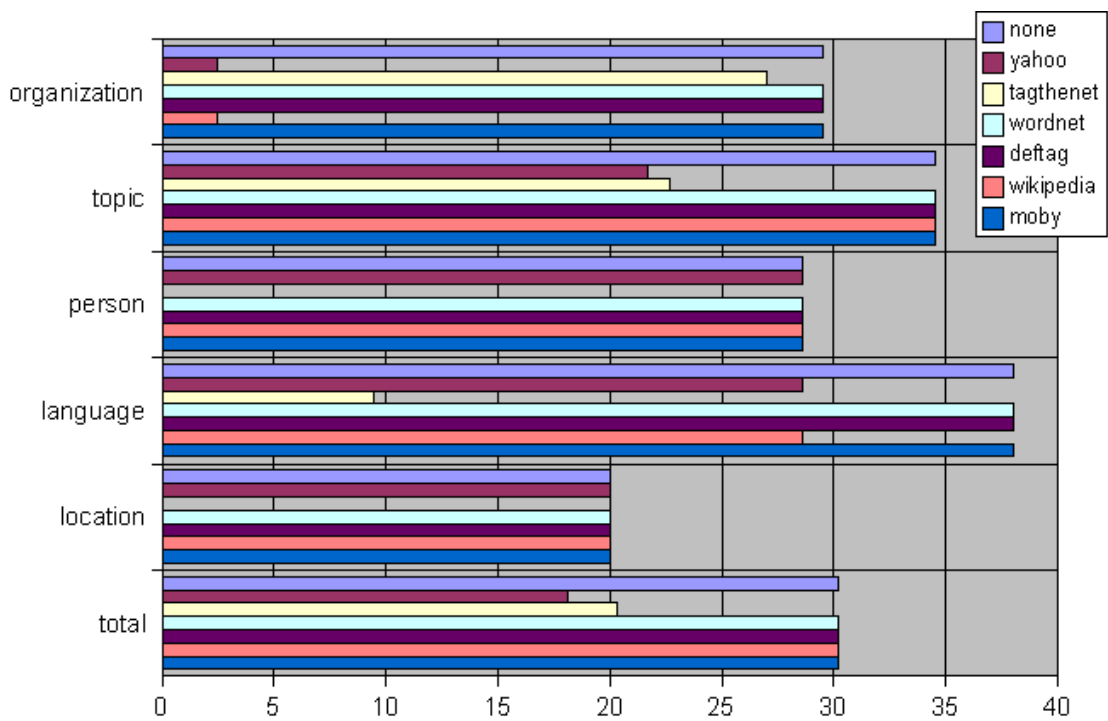


Figure 6.16: Use Case 2: Distribution of topic proposals that do not exist in the Test PIMO, but occur in *Index Sites' Corpus*

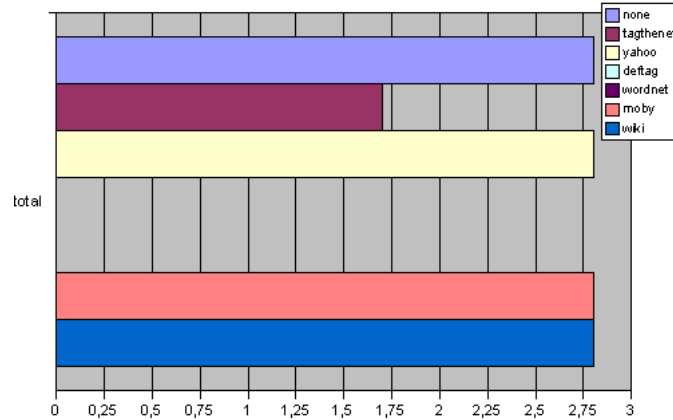


Figure 6.17: Use Case 3: Distribution of new class proposals in *Wikisource's Historical Corpus*

Use Case 3

The Evaluation of Use Case 3 and the influences of used web services should reveal, the quality of ConTag in creating new classes and assigning them to existing PIMO classes. The following figures 6.17, 6.18, 6.19 express successfully generated proposals of new PIMO classes. Unfortunately, the iteration of Reuters' News corpus did not result in any proposal of new classes. Therefore there are not figured. Again each bar of a specific class corresponds to a web service which has been omitted.

The figures describing Wikisource's Historical Corpus and Wikipedia's Concept Corpus reveal, that none of the proposed classes correspond to desired subsumptions in the Ground Truths. At least a small percentage of correct classes were extracted, but not correct classified. However, the iteration of Index Sites' Corpus results in a correct subclass proposal of location, which was Europe in this case. Obviously the combination of DefTag and Wordnet reach their highest quality in this third Use Case. Every test result suffers in omitting one of the two services.

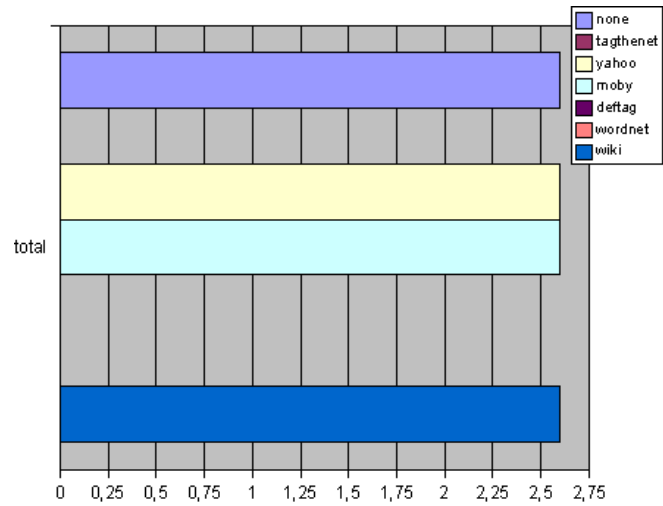


Figure 6.18: Use Case 3: Distribution of new class proposals in *Wikipedia's Concept Corpus*

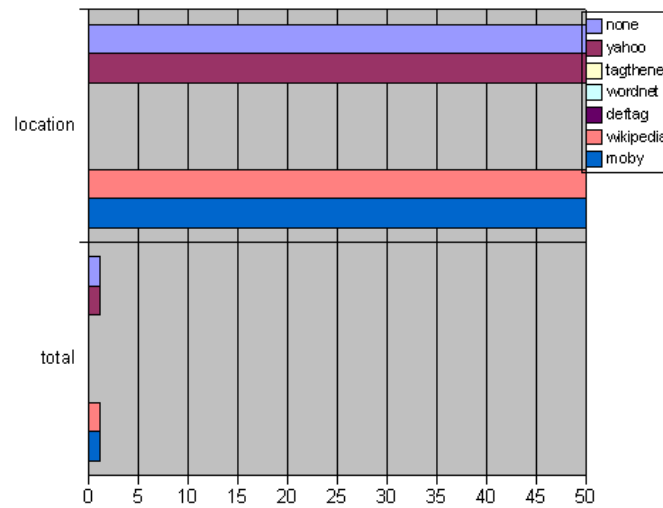


Figure 6.19: Use Case 3: Distribution of new class proposals in *Index Site's Corpus*

6.5.2 Precision analysis

The ratio of relevant tags in an overall tag proposal (called Precision) can be manipulated in ConTag, by changing the threshold values for similarities used in the Alignment Generation step. Therefore two configuration sets, one for each similarity threshold parameter (Content Similarity and Label Similarity) were created. In each configuration set one parameter is fixed, the other varies in a range between 0 and 1 in increments of 0.2. All supported Web 2.0 services were used in these test run configurations:

Content Similarity Analysis The Label Similarity is fixed to 0.5.

Label Similarity Analysis The Content Similarity is fixed to 0.23.

The fixed threshold degrees result from experience values during several test runs of ConTag and were proposed to be the configuration of best results. Figure 6.20 contains two Precision diagrams and expresses (a.) the Label Similarity Analysis and (b.) the Content Similarity Analysis for an exemplary test resource. The configuration values, returning best Precision results regarding Fig. fig:PrecSim can be loosely defined as follows:

```
Content Similarity := 0.2 to 0.3
Label Similarity := 0.8
```

A comparison between these configuration parameters, providing optimal Precision ratios, and corresponding Recall values should reveal the overall and optimal similarity configuration. Therefore the figures 6.21 and 6.22 contain Recall progressions based on the same test scenario as above. In this case optimal similarity parameters, returning best Recall results, are:

```
Content Similarity := 0.2 to 0.3
Label Similarity := 0.4
```

The fixed similarity thresholds used in the Content and Label Similarity Analysis can be used to execute ConTag with a compromise providing a configuration between optimal Recall and Precision ratios.

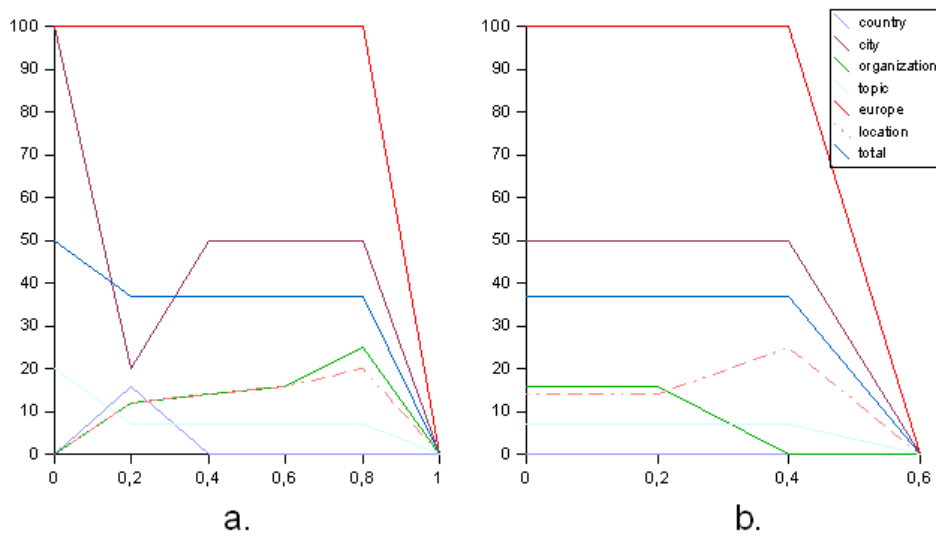


Figure 6.20: Precision degrees concerning different similarity thresholds. (a.) Label Similarity; b.) Content Similarity)

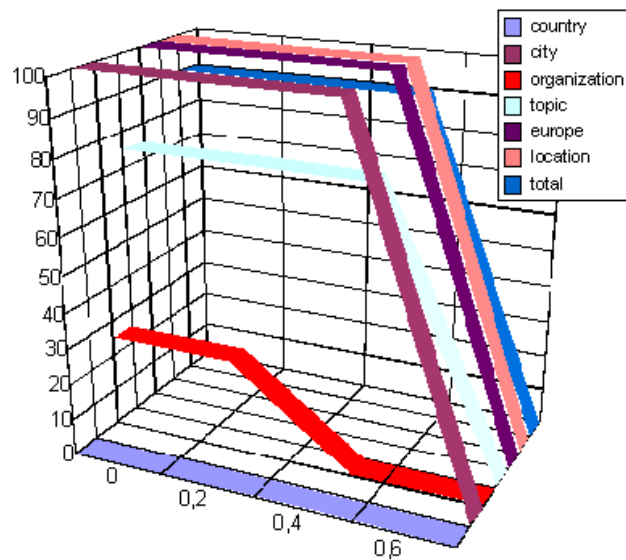


Figure 6.21: Recall ratios concerning different Content Similarity thresholds.

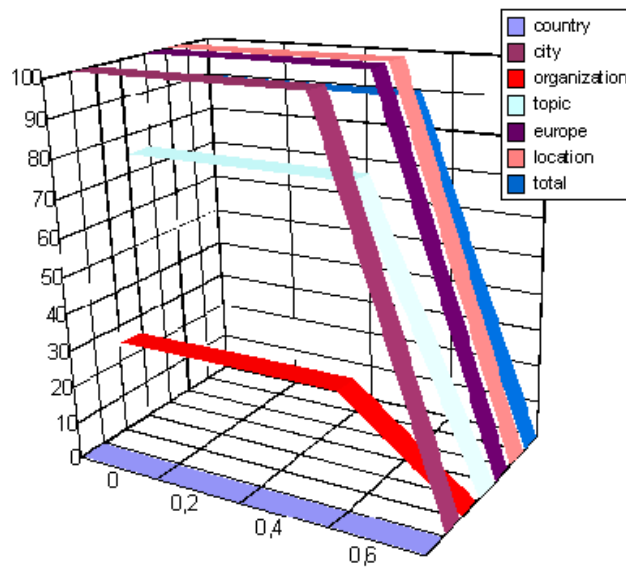


Figure 6.22: Recall ratios concerning different Label Similarity thresholds.

Chapter 7

Conclusion

“Science fiction is no more written for scientists than ghost stories are written for ghosts.”

- Brian Aldiss -

ConTag is an approach to connect several technologies and branches of research together in order to support users to manage their knowledge concerning new encountered information in text documents. This chapter shortly summarizes the course of action, decided to be developed in ConTag. At the same time a review of critical used methods figures out existing open problems.

7.1 Summary and Discussion

ConTag’s functionalities support users to annotate documents, written in natural language, with textual tags, that concern to personal concepts managed in a Semantic Desktop environment.

The initial and basic motivation to develop ConTag is expressed in the one sentence above. Three basic use cases were derived from this philosophy. In order to solve them, it was decided to design ConTag as a Tagging System to annotate text resources with tags specified in the Gnowsis PIMO. This architecture reduced the complexity of ConTag, described as a semantic aware software system. Instead of implementing software, that understands documents in order to classify contained topics into the user’s PIMO, a Tagging System Architecture just annotates documents with specified things in the Personal Information Model.

The evolution of tag proposals in ConTag followed a stepwise production along ConTag’s *Tagging Process*. This approach enabled the encapsulation of closed

ways of posing a problem. Each step is well defined in its functionalities and therefore the whole process offers starting points to enhance or completely exchange existing implementations and approaches to follow one iteration.

It was decided to build a complete *Topic Map* on top of the text resource in a step called *Topic Extraction*. This Topic Map, implemented by a data structure called Information Cloud, was intended to be coherent towards the written scope of contents in the document and describes occurring topics in note form. In order to provide conformity to existing approaches, it is suggested to express the Information Cloud's Topic Map in a standard vocabulary, that is both compliant to existing standards such as SKOS and also efficient to be traversed in ConTag's *Alignment Execution*

Additionally, in future version of ConTag's Tagging Process, the user should possibly be enabled to interrupt the tagging process after the *Topic Extraction* step, in order to inspect and change the describing Topic Map in his own discretion.

The use of new Web 2.0 services solving problems in domains of content analysis, hypernym extraction, definition lookups and the generation of word association should be promoted in order to strengthen the expressiveness of each generated Topic Map. A list of such relevant services can be found in Appendix A.2. This may lead to more complex topologies inside the Information Cloud and possibly manages to build *Topic Ontologies* on top of a web document.

Generally, the evaluation of ConTag's *Topic Extraction* points out, that the additional usage of new Web 2.0 services increases the degree of *Recall* values in Topic Maps.

The computation of properly classified tag proposals was decided to be solved by a *taxonomic and statistical Nearest Neighbor Classification* approach. In regard to the development of ConTag as a balanced system, the evolved classification service was forced to be lightweight. Therefore, the computation of an ontological alignment between entities in Topic Maps and entities in a Personal Information Model is completely based on untyped feature similarities. The degree of *Precision* in resulting tag proposals may be increased by using typed feature vectors to be able to inspect additional semantic properties between a pair of entities. An example of typical typed features is the comparison of relevant Wikipedia articles regarding found and existing things and concepts. Maybe a thing in a Personal Information Model is annotated by a descriptive Wikipedia article. The Topic Extraction Phase encounters a topic with an existing Wikipedia article about it. The comparison of both Wikipedia articles may shed light on equalities or relations. In spite of an increased feature vector management, other ontology alignment methods (e.g. topological similarities) should be tested in ConTag to enhance the degree of *Precision* of classified tag proposals. Due to the fact that the *Alignment Generation* causes high CPU utilization and may last several minutes, the used classification

approach should be refined in terms of performance issues.

It is supposed, that the *Alignment Generation* is the major starting point to enhance Precision values in tag proposals.

The evaluation results of ConTag calculated in Chapter 6 shows that, after finishing the *Alignment Creation*, the user is supported with tag proposals of which round about 40% are desired and relevant. Unfortunately this degree of *Recall* is correlated with a ratio of about 30% *Precision*. Nevertheless, the quality criterias described in each Use Case are proposed to be at least partially solved. These values correlate with a static Ground Truth specification, which is just a temporary snapshot of user desired tag proposals. The Ground Truth creator did not rate generated proposals, he rather specified, what he wanted to be proposed. This approach may be defined as a pessimistic evaluation, because generated proposals, that do not occur in the Ground Truth are not forced to be irrelevant at all.

The efficient and concise presentation of tag proposals to end users in a GUI is difficult to solve. The essential question is, how to present tags to a user in way that provides an easy and quick understanding and uses inherent and user familiar usability techniques. The implemented technique of Tim BernersLee's described 'Oh yeah! button' in combination with an AJAX driven HTML interface provides a basis.

The major result of summarizing this study about ConTag reveals, that by using ConTag users (such as Paul) are supported to merge all types of information resources, occurring in text documents (e.g. the project description in Fig. 1.1), into their Personal Information Model.

7.2 Outlook

ConTag supports users to maintain software systems based on personal ontologies. It is not restricted to be used in Semantic Desktop system, but may be deployed in any semantic aware Personal Information Management system used to process information written in natural language. Tagging systems, task management systems, e-mail systems and other software solutions may benefit by ConTag's tag extractions to provide more efficient retrieval possibilities.

The further development of Use Case 4, relating existing or to be created concepts with semantically weak defined relationships based on the analysis written information, is proposed to contain a great capability of increasing the cohesion of a Personal Information Model.

The further evolution of existing and new Web 2.0 services has to be supported by using them. The development of ConTag reveals the hidden semantic strength of todays World Wide Web, provided by a variety of tiny services.

Collaborative web approaches are proposed to be rising stars in the firmament of a worldwide WWW community to guide the way to the Semantic Web we all desire.

Appendix A

Appendix

A.1 Existing tagging systems

Tagging Service Name	Resource Types	URL
Delicious	web pages	http://del.icio.us
Flickr	images	http://www.flickr.com
Technorati	Blog posts	http://www.technorati.com
Citeulike	bibliographies	http://www.citeulike.org
Bibsonomy	bibliographies	http://www.bibsonomy.org
PiggyBank	web pages	http://simile.mit.edu/piggy-bank
Semanlink	web pages	http://www.semanlink.net/sl/new
Tag Triples	web pages	http://tagtriples.sourceforge.net

A.2 Inspected Web 2.0 services

Tagyu Tagyu is a hosted service that automates the process of creating document metadata by discovering what keywords, tags, and categories are relevant for a document.

...

Tagyu can categorize based on the wisdom present in a controlled set of documents that you provide, or based on a larger set of content found "in the wild" content from social bookmarking, blogs, and other Web sources.
<http://tagyu.com/>

This Web 2.0 service shows promise. Although Tagyu doesn't provide such a meaningful classification of text contained tags as Tagthe.net, it provides a browsing mechanism about each tag it proposes. Every tag existing in

supported Tagging Systems is also accessible and described by Tagyu. (e.g. <http://tagyu.com/tag/contag>). Unfortunately this Web 2.0 service was in development during ConTag's evaluation.

Flickr Services Flickr Services are a set of REST queries, to access several information inside Flickr. The REST query called *flickr.tags.getRelated* returns a list of tags "related" to the given tag, based on clustered usage analysis. <http://www.flickr.com/services/api/flickr.tags.getRelated.html>

Ontok Wikipedia API The Web 2.0 service hosted by Ontok provides two additional REST queries to access Wikipedia, that are adjudged as highly valuable:

- *GetWikipediaPageInfo* returns the page info for strings in a text detected to be Wikipedia pages .
- *GetCategories* detects strings in a text and maps them onto several popular category spaces (Amazon.com, Shopping, "magazines" and "DMOZ"), returning the log likelihoods of the strings being generated by each category.

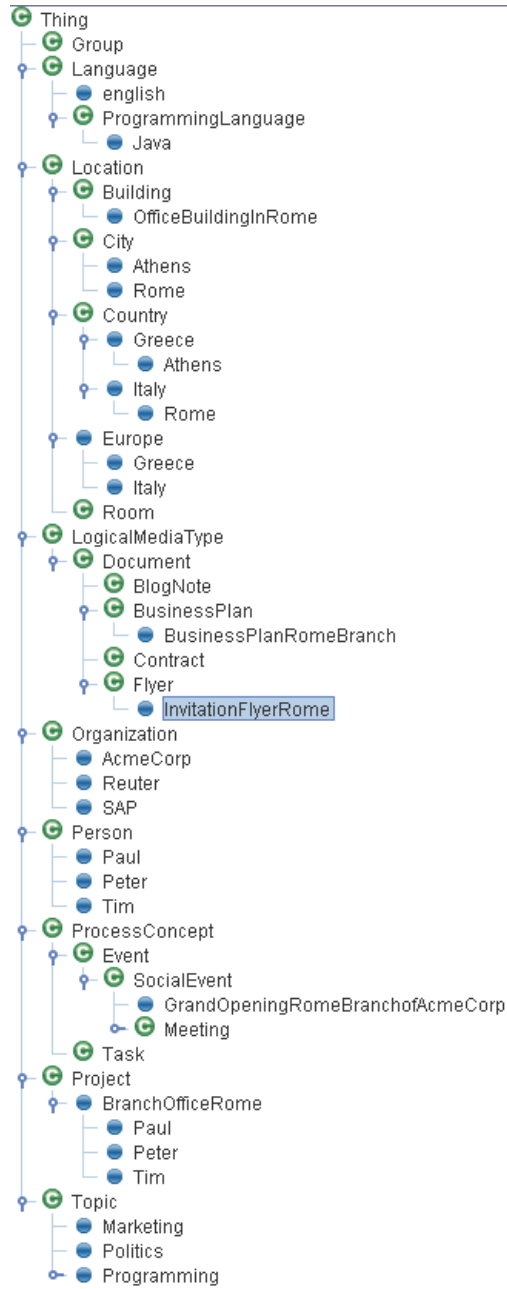
<http://www.ontok.com/wiki/index.php/Wikipedia>

Edinburgh Associative Thesaurus The Edinburgh Associative Thesaurus (EAT) is a set of word association norms showing the counts of word association as collected from subjects. This is not a developed semantic network such as WordNet, but empirical association data. <http://www.eat.rl.ac.uk>

Yahoo! Related Tags service The *Related Tags* service by Yahoo returns tags, being related to a given tag, by using cooccurrences in supported Tagging Systems such as Flickr or del.icio.us. <http://developer.yahoo.com/search/myweb/V1/relatedTags.html>

A.3 Test PIMO

The figure below describes the Test PIMO, used in ConTag's validation to create a expressive Ground Truth. Classes that were not used in the Ground Truth and do not contain any instances are not listed.



A.4 Test corpora's contents

The following list of contents describe the exact source location of each test resource in a test corpus. Due to the fact, that apart from Reuter's News Corpus all other resources are web pages, it may occur that the given URL points to different contents than ConTag's validation used before.

A.5 Reuter's News Corpus

The content of Reuter's News Corpus is copyrighted. For the sake of completeness, the following list of corpus entries describe the used test data.

1. 807590newsML.xml
2. 807591newsML.xml
3. 807592newsML.xml
4. 807594newsML.xml
5. 810479newsML.xml
6. 810503newsML.xml
7. 810543newsML.xml
8. 810570newsML.xml
9. 810588newsML.xml
10. 810597newsML.xml

A.6 Index Site Corpus

These documents were copied to be used as test data in the Index Site Corpus.

1. http://en.wiktionary.org/wiki/Main_Page.html
2. http://ec.europa.eu/index_en.htm
3. <http://java.source.net>
4. <http://java.sun.com>

5. <http://www.apache.org>
6. <http://www.gnowsis.org>
7. <http://www.linux.org>
8. <http://www.microsoft.com/windows>
9. <http://www.ricoh.rlp-labs.de>
10. <http://www.sap.com>

A.7 Wikipedia Concept Corpus

These articles were descended from a web encyclopedia called wikipedia. (see <http://en.wikipedia.org>)

1. <http://en.wikipedia.org/wiki/Biology>
2. http://en.wikipedia.org/wiki/Computer_science
3. <http://en.wikipedia.org/wiki/Engineering>
4. <http://en.wikipedia.org/wiki/Law>
5. <http://en.wikipedia.org/wiki/Linguistics>
6. <http://en.wikipedia.org/wiki/Medicine>
7. <http://en.wikipedia.org/wiki/Philosophy>
8. <http://en.wikipedia.org/wiki/Physics>
9. http://en.wikipedia.org/wiki/Political_science
10. <http://en.wikipedia.org/wiki/Psychology>

A.8 Wikisource Historical Corpus

These articles were descended from a web library called wikisource. (see <http://en.wikisource.org>)

1. http://en.wikisource.org/wiki/Canadian_Charter_of_Rights_and_Freedoms

2. http://en.wikisource.org/wiki/Constitution_of_the_United_States_of_America
3. http://en.wikisource.org/wiki/Fuehrer_Directive_21
4. http://en.wikisource.org/wiki/German_Instrument_of_Surrender1945
5. http://en.wikisource.org/wiki/Greeting_to_American_Soldiers_by_the_women_of_France
6. http://en.wikisource.org/wiki/Law_of_Administration_for_the_State_of_Iraq
7. http://en.wikisource.org/wiki/Mayflower_Compact
8. http://en.wikisource.org/wiki/Protokoll_der_Wannsee-Konferenz
9. http://en.wikisource.org/wiki/Warning_to_Travellers_to_Great_Britain
10. http://en.wikisource.org/wiki/Zimmermann_Telegram

A.9 Used software libraries

ConTag is implemented by using the Java 1.5 programming language. The used Integrated Development Environment was Eclipse 3.1.

The following public software libraries were used to create ConTag:

SimMetrics	http://sourceforge.net/projects/simmetrics
OpenNLP	http://opennlp.sourceforge.net
Jena	http://jena.sourceforge.net
RDF2Java	http://rdf2java.opendfki.de/cgi-bin/trac.cgi
Google Web Toolkit	http://code.google.com/webtoolkit
JDOM 1.0	http://www.jdom.org
Lucene	http://lucene.apache.org
OpenJGraph	http://openjgraph.sourceforge.net
Xerces Java Parser	http://xerces.apache.org/xerces&j
Gnowsis	http://www.gnowsis.org
Yahoo! Web Service SDK	http://developer.yahoo.com/search
Porter Stemming Algorithm	http://tartarus.org/~martin/PorterStemmer

List of Figures

1.1	Tagging a Business Project Description, using ConTag	3
2.1	Annotation System	9
3.1	Tag Ontology	19
3.2	Tagging Process	22
3.3	Tag Views	25
4.1	Use Case: Tagging a web document using ConTag	29
5.1	Architecture	34
5.2	Business Project Description	37
5.3	Internal Topic Map	39
5.4	Alignment Ontology	44
5.5	Icons	46
5.6	Sidebar	46
6.1	Distribution of desired classifications in the <i>Reuters' News Corpus</i>	54
6.2	Distribution of desired classifications in the <i>Index Sites Corpus</i> . .	55
6.3	Distribution of desired classifications in the <i>Wikipedia Concept Corpus</i>	55
6.4	Distribution of desired classifications in the <i>Wikisource Historical Corpus</i>	56
6.5	Distribution of relevant topic classifications found in the <i>Reuters News Corpus</i>	58
6.6	Distribution of relevant topic classifications found in the <i>Index Sites Corpus</i>	58
6.7	Distribution of relevant topic classifications found in the <i>Wikipedia Concept Corpus</i>	59
6.8	Distribution of relevant topic classifications found in the <i>Wikisource Historical Corpus</i>	59

6.9	Use Case 1: Distribution of identified existing topics in the <i>Wikisource Historical Corpus</i>	61
6.10	Use Case 1: Distribution of identified existing topics in the <i>Wikipedia Concept Corpus</i>	62
6.11	Use Case 1: Distribution of identified existing topics in the <i>Reuters' News Corpus</i>	62
6.12	Use Case 1: Distribution of identified existing topics in the <i>Index Sites Corpus</i>	63
6.13	Use Case 2: Distribution of topic proposals that do not exist in the Test PIMO, but occur in <i>Wikisource's Historical Corpus</i>	64
6.14	Use Case 2: Distribution of topic proposals that do not exist in the Test PIMO, but occur in <i>Wikipedia's Concept Corpus</i>	65
6.15	Use Case 2: Distribution of topic proposals that do not exist in the Test PIMO, but occur in <i>Reuters' News Corpus</i>	66
6.16	Use Case 2: Distribution of topic proposals that do not exist in the Test PIMO, but occur in <i>Index Sites' Corpus</i>	67
6.17	Use Case 3: Distribution of new class proposals in <i>Wikisource's Historical Corpus</i>	68
6.18	Use Case 3: Distribution of new class proposals in <i>Wikipedia's Concept Corpus</i>	69
6.19	Use Case 3: Distribution of new class proposals in <i>Index Site's Corpus</i>	69
6.20	Precision degrees concerning different similarity thresholds. (a.) Label Similarity; b.) Content Similarity)	71
6.21	Recall ratios concerning different Content Similarity thresholds.	71
6.22	Recall ratios concerning different Label Similarity thresholds.	72

Bibliography

- [AI99] Douglas E. Appelt and David J. Israel, *Introduction to information extraction technology*, A tutorial prepared for IJCAI-99, Stockholm, Sweden, 1999.
- [BDBD⁺01] Gabe Begeed-Dov, Dan Brickley, Rael Dornfest, Ian Davis, Leigh Dodds, Jonathan Eisenzopf, David Galbraith, R.V. Guha, Ken MacLeod, Eric Miller, Aaron Swartz, and Eric van der Vlist, *RDF site summary (RSS) 1.0*, 2001.
- [Bec06] Dave Beckett, *Semantics through the tag*, XTech 2006 Building Web 2.0 (2006).
- [Bid04] Matt Biddulph, *Introducing del.icio.us*, online article at <http://www.xml.com/pub/a/2004/11/10/delicious.html>, November 2004.
- [BL80] Tim Berners-Lee, *The enquire system*, Manual, Cern, October 1980.
- [BL97] ———, *Cleaning up the user interface*, web article at <http://www.w3.org/DesignIssues/UI.html>, September 1997.
- [BL99] Tim Berners-Lee, *Weaving the web*, Texere Publishing Ltd., November 1999.
- [BLFM05] Tim Berners-Lee, Roy Fielding, and L. Masinter, *Uniform resource identifier (URI): Generic syntax*, RFC 3986, Internet Engineering Task Force, January 2005.
- [BLHL01] Tim Berners-Lee, James Hendler, and Ora Lassila, *The semantic web*, Scientific American (2001).
- [BM05] Dan Brickley and Libby Miller, *FOAF vocabulary specification*, XML namespace document, July 2005, Namespace Document 27 July 2005 - ('Pages about Things' Edition).

- [Bus45] Vannevar Bush, *As we may think.*, The Atlantic Monthly **176** (1945), no. 1, 101–108.
- [CD99] James Clark and Steve DeRose, *Xml path language (xpath)*, W3c recommendation, W3C, <http://www.w3.org/TR/xpath>, November 1999.
- [Coc98] Alistair Cockburn, *Basic use case template*, online article at http://alistair.cockburn.us/index.php/Basic_use_case_template, October 1998.
- [Daw05] Phil Dawes, *Folksonomy and structured metadata*, <http://www.phildawes.net/blog/2005/02/03/folksonomy-and-structured-metadata>, February 2005.
- [EC01] Greg Meredith Sanjiva Weerawarana Erik Christensen, Francisco Curbera, *Web services description language (wsdl) 1.1*, Tech. report, W3C, 2001.
- [FE89] R.A. Floyd and C. Schlatter Ellis, *Directory reference patterns in hierarchical file systems*, IEEE Transactions on Knowledge and Data Engineering **01** (1989), no. 2, 238–247.
- [FGM⁺99] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, *Hypertext transfer protocol – HTTP/1.1*, RFC 2616, Internet Engineering Task Force, June 1999.
- [Fie00] Roy Thomas Fielding, *Architectural styles and the design of network-based software architectures*, Ph.D. thesis, University of California, 2000, Chair-Richard N. Taylor.
- [FM97] R. Faith and B. Martin, *A dictionary server protocol*, Request for Comments RFC2229, U. North Carolina, Chapel Hill and Miranda Productions, <http://www.ietf.org/rfc/rfc2229.txt>, October 1997.
- [Gar05] Jesse James Garrett, *Ajax: A new approach to web applications*, online article at <http://adaptivepath.com/publications/essays/archives/000385.php>, February 2005.
- [GH05a] Scott Golder and Bernardo A. Huberman, *The structure of collaborative tagging systems*, Journal of Information Science **32(2)** (2005), 198–208.

- [GH05b] Heather Green and Robert D. Hof, *Picking up where search leaves off*, online article at http://www.businessweek.com/magazine/content/05_15/b3928112_mz063.htm, april 2005.
- [GHJV95] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, *Design patterns*, Addison-Wesley Professional, January 1995.
- [HBEV04] Peter Haase, Jeen Broekstra, Andreas Eberhart, and Raphael Volz, *A comparison of rdf query languages*, Proceedings of the Third International Semantic Web Conference, Hiroshima, Japan, 2004., NOV 2004.
- [HMK05] D. Huynh, S. Mazzocchi, and D. Karger, *Piggy bank: Experience the semantic web inside your web browser*, International Semantic Web Conference (E. and Motta, V. R. Benjamins, and M. A. Musen, eds.), 2005.
- [KK01] José Kahan and Marja-Ritta Koivunen, *Annotea: an open RDF infrastructure for shared web annotations*, Proceedings of the 10th International World Wide Web Conference, 2001, pp. 623–632.
- [Kuh76] Rainer Kuhlen, *Experimentelle Morphologie in der Informationswissenschaft*, Datenbasen Datenbanken Netzwerke, Saur München, 1976.
- [LM95] D. Levy and C. C. Marshall, *Going digital: A look at assumptions underlying digital libraries*, Commun. ACM (1995), no. 38, 77–84.
- [MB05] Alistair Miles and Dan Brickley, *SKOS core vocabulary specification*, W3c working draft, World Wide Web Consortium, November 2005.
- [MG03] Noah Mendelsohn Jean-Jacques Moreau Henrik Frystyk Nielsen Martin Gudgin, Marc Hadley, *Soap version 1.2 part 1: Messaging framework*, W3c recommendation, World Wide Web Consortium, June 2003.
- [Mic06] Sun Microsystems, *Core J2EE patterns - session facade*, Sun blueprint, Sun Microsystems, July 2006.
- [MM04] Frank Manola and Eric Miller, *RDF primer*, W3c recommendation, World Wide Web Consortium, February 2004.

- [New05] Richard Newman, *Tag ontology design*, blog entry at <http://www.holygoat.co.uk/projects/tags>, March 2005.
- [OAS04] OASIS, *Introduction to UDDI: Important features and functional concepts*, Tech. report, OASIS, 2004.
- [O'R05] O'Reilly, Tim, *O'reilly network: What is Web 2.0*, online article at <http://www.oreillynet.com/lpt/a/6228>, September 2005.
- [PM01] S. Pepper and G. Moore., *XML topic maps (XTM) 1.0*, Tech. report, TopicMaps.Org Authoring Group, <http://www.topicmaps.org/xtm/index.html>, August 2001.
- [Por97] M. F. Porter, *An algorithm for suffix stripping*, 313–316.
- [PS06] Eric Prud'hommeaux and Andy Seaborne, *SPARQL query language for RDF*, W3C candidate recommendation, W3C, April 2006.
- [Rij79] C. J. van Rijsbergen, *Information retrieval*, 2 ed., Butterworths, London, 1979.
- [Roh05] Jean Rohmer, *Lessons for the future of semantic desktops learnt from 10 years of experience with the IDELIANCE semantic networks manager*, Proc. of Semantic Desktop Workshop at the ISWC, Galway, Ireland, November 6 (Stefan Decker, Jack Park, Dennis Quan, and Leo Sauermann, eds.), vol. 175, November 2005.
- [Sau03] Leo Sauermann, *The gnosis — using semantic web technologies to build a semantic desktop*, Master's thesis, Technische Universität Wien, December 2003.
- [Sau06] ———, *Pimo - a pim ontology for the semantic desktop*, draft article at <http://www.dfki.uni-kl.de/~sauermann/2006/01-pimo-report/pimOntologyLanguageReport.html>, 2006.
- [SBD05] Leo Sauermann, Ansgar Bernardi, and Andreas Dengel, *Overview and outlook on the semantic desktop*, Proceedings of the 1st Workshop on The Semantic Desktop at the ISWC 2005 Conference (Stefan Decker, Jack Park, Dennis Quan, and Leo Sauermann, eds.), 2005.

- [Sch05] Sven Schwarz, *A context model for personal knowledge management*, Proceedings of the 2nd International Workshop of Modelling and Retrieval of Context (MRC 2005) in conjunction with IJCAI 2005 (Edinburgh), jul 2005, pp. 39–50 (english).
- [Ser06] François-Paul Servant, *Semanlink*, Jena User Conference (2006).
- [SS03] Sven Schwarz and Michael Sintek, *rdf2java*, Tech. report, German Research Center for Artificial Intelligence (DFKI) GmbH; Knowledge Management Group, <http://rdf2java.opendfki.de>, 2003.
- [Tan92] Andrew S. Tanenbaum, *Modern operating systems*, Internals and Design Principles, Prentice-Hall, International, 1992, TAN a 92:1 P-Ex.
- [Ter05] Daniel Terdiman, *Folksonomies tap people power*, Blog entry at <http://www.wired.com/news/technology/0,1282,66456,00.html>, 02 2005.
- [W3C03] W3C, *Some early ideas for html*, online article at <http://www.w3.org/MarkUp/historical>, January 2003.
- [Wal04] Thomas Vander Wal, *Would we create hierarchies in a computing age?*, blog entry at <http://www.vanderwal.net/random/entrysel.php?blog=1598>, December 2004.
- [XC05] Huiyong Xiao and Isabel F. Cruz, *A multi-ontology approach for personal information management*, Proc. of Semantic Desktop Workshop at the ISWC, Galway, Ireland, November 6 (Stefan Decker, Jack Park, Dennis Quan, and Leo Sauermann, eds.), vol. 175, November 2005.