Ontology Learning from semi-structured Web Documents

Dissertation

zur Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg

von Dipl. Wirt.-Ing. (FH) Marko Brunzel
geb. am 21. Januar 1977 in Meerane

Gutachter:

Prof. Dr. Steffen Staab
Prof. Dr. Myra Spiliopoulou
Prof. Dr. Andreas Dengel

Magdeburg, den 17. Februar 2010

Contents

List of Figures

List of Tables

List of Algorithms

Abstract

The research field of ontology learning is about acquiring semantic relations among entities to be represented in ontologies. Usually unstructured text documents are used as input data. In the last years large numbers of Web documents have become available. Using the Web as input data for ontology learning eliminates the user from manually assembling a document collection. In this thesis large quantities of Web documents have been used for learning. Web documents are semi-structured; they consist of structured and unstructured ranges. The semi-structure represents added value created by many Web authors which is worth to be used. The aim is to exploit the semi-structure available in Web documents to learn ontology constituents instead of eliminating the semi-structure by conversion to plain text. The ontology constituents to be learned within this thesis are sibling relations, terms and synonyms. Those ontology constituents are important for creating ontologies. The emphasis is on acquiring semantically plausible sibling relations. The core method applied in several approaches is to create paths for the text spans of Web documents according to the structural nesting of structural markup. Text spans with equal paths are grouped as siblings. The obtained structural siblings are afterwards further processed. We learn groups of sibling terms, hierarchies of sibling term groups and sibling term pairs. Our approach is language independent since it relies on structural characteristics of Web documents. Multiword terms which are to be handled are treated in the same way as simple single word terms. This is especially important for languages like English where compound terms are not used to the same extent as in German language. The learned sibling relations are evaluated according to gold standard ontologies. The results show that the quality is higher than what is obtained by prior approaches.

Kurzfassung

Das Forschungsfeld des Ontologielernens beschäftigt sich mit dem Erwerb von semantischen Beziehungen zwischen Entitäten die in Ontologien repräsentiert werden. Unstrukturierte Text-Dokumente dienen hierfür bisher meist als Datenquelle. In letzten Jahren ist sind riesige Mengen an Web-Dokumenten verfügbar geworden. Die Verwendung des Webs als Datenquelle für das Ontologielernen befreit den Anwender davon selbst manuell eine Dokumentensammlung zusammenzustellen. In dieser Arbeit werden große Mengen an Web-Dokumenten als Grundlage für das Lernen verwendet. Web-Dokumente sind semistrukturiert, sie bestehen aus strukturierten und unstrukturierten Bereichen. Die Semistruktur repräsentiert einen von vielen Web-Dokument Autoren manuell geschaffenen Mehrwert, der es wert ist genutzt zu werden. Das Ziel ist es die in Web-Dokumenten enthaltene Semistruktur heranzuziehen um Ontologiebestandteile zu akquirieren, anstatt sie durch Konvertierung zu reinem Text zu beseitigen. Die in dieser Dissertation zu akquirierenden Ontologiebestandteile sind Geschwisterbeziehungen, Begriffe und Synonyme. Die gefundenen Ontologiebestandteile sind wichtig für das Erstellen von Ontologien. Der Schwerpunkt liegt auf dem Erwerb der semantisch plausiblen Geschwisterbeziehungen. Der Kernansatz der in den einzelnen Verfahren verwendet wird ist es, zu den Textabschnitten in den Web-Dokumenten Pfade anhand der Verschachtelung der Strukturauszeichnung zu erstellen. Textabschnitte mit gleichen Pfaden werden als Geschwister gruppiert. Die gefundenen strukturellen Geschwisterbegriffe werden nachfolgend weiterverarbeitet. Es werden Gruppen von Geschwisterbegriffen, Hierarchien von Gruppen von Geschwisterbegriffen und Geschwisterbegriffspaare erlernt. Da dieser Ansatz auf strukturellen Eigenschaften von Web-Dokumenten beruht ist er Sprachunabhängig. Die oft viel schwieriger zu handhabenden Mehrwortbegriffe werden hierbei genauso berücksichtiget wie einfache Wörter. Dies ist besonders wichtig für Sprachen wie die englische Sprache in der zusammengesetzte Wörter nicht so oft verwendet werden wie in der deutschen Sprache. Die erlernten Geschwisterbeziehungen werden anhand von Referenzontologien evaluiert. Die Ergebnisse zeigen dass die Güte höher als bei bisherigen Verfahren ist.

1 Introduction

1.1 Motivation

At the turn oft the millennium the interest in *ontologies* was increased on account of the idea of the semantic Web [Berners-Lee, 1998, Berners-Lee et al., 2001]. Ontologies are shared conceptualizations [Gruber, 1993] for representing domain knowledge. However, ontologies had been rare. The shortage of existing ontologies and the problems that cropped up during the creation of ontologies were referred to as the *knowledge acquisition bottleneck*. *Ontology engineering* is the field concerned with the methods for creating ontologies. The manual creation of ontologies is expensive. The idea was to semi-automatically support the ontology engineer in ontology construction by means of *ontology learning* [Maedche and Staab, 2000, Maedche and Staab, 2001]. The research field of ontology learning comprises methods for acquiring domain models from data.

Several sources of data have been used as input for ontology learning processes. The vast majority of approaches for performing ontology learning are designed for performing *ontology learning from text* documents. Some years ago, text documents had been the predominant source of textual content which was available to domain experts in contrast to the newly grown Web. The Web nowadays provides a huge source of content on nearly every topic one can think of. Subsequently, the Web constitutes a significant source of input data to be used for ontology learning. In this thesis we will use the *Web as input data source* for ontology learning.

1.2 Using the Web for Ontology Learning

There are several arguments for using the Web as data source for ontology learning: (1) basically it is another source of input which should be explored thoroughly, (2) the Web has become the dominant source of digital content and should not be ignored, (3) the Web covers almost all topics and domains one can think of, (4) Web documents are publicly available providing (5) the possibility of getting collections of Web documents automatically which eases the overall learning process, and (6) particular characteristics of Web documents, the semi-structure, bear opportunities. Points 4 to 6 are further explained in the next sections, but before that we have a brief look at the disadvantages of using Web documents for ontology learning.

The quality of Web documents varies to a large extent. Documents in different languages, possibly mixed languages in single documents, misspellings and slang

language use are common. The Web documents are published by unknown entities pursuing different goals, and so a single Web document cannot be regarded as a trustworthy document per se. Web documents do not often adhere to the Web document standards perfectly. Natural language processing focuses on processing only a few hundreds or thousands of documents. Web document collections, on the other hand, can be in the range of millions and billions of documents. This limits the applicability of existing processing techniques with high complexity and increases the demand for more adequate processing techniques. The Web is not static; performing a Web crawl with equal parameters of the Web crawler at a different point of time yields different results. Those sketched drawbacks are the challenges in the approaches to be used on Web documents. A general principle which we rely upon to overcome the above mentioned potential problems is that using large quantities of Web documents is expected to overcome quality problems on fractions of processed Web documents. The Web as a whole reflects a rather reliable source of human knowledge. The Wisdom of the crowds [Surowiecki, 2004] can overcome particular shortcomings. The dynamism is not so much a problem as an advantage. The Web reflects newly covered content and topics.

An advantage of the Web is that the Web documents are publicly available. This is advantageous in a situation where a corporate ontology engineer creates an ontology which is to be shared with other parties afterwards. The ontology engineer can perform the learning process on freely available content, liberated from the necessity that potentially private information is made publicly available in the ontology which should be kept private.

The public availability of Web documents opens another opportunity, the automatic acquisition of Web document collections. Methods for ontology learning from text usually rely on the availability of a local collection of text documents of high quality. As a consequence, the ontology engineer has to provide a document collection of reasonable size and coverage of a domain. The manual assembling of such a document collection is a laborious effort which is not even straightforward; freeing the ontology engineer from the task of providing a document collection decreases the overall amount of human efforts which goes into ontology learning processes.

A general critique on using plain text as input is that it is questionable if plain text is indeed suited for the acquisition of shared conceptualizations. The background knowledge which is made explicit in ontologies is the kind of information which is rarely used in written textual communication because the sender usually assumes that the receiver already has this background (domain) knowledge and, therefore, he usually relies on it without repeating it again [Brewster et al., 2003]. This can be different for Web documents. While creating and publishing Web documents, authors are sometimes willing (or forced) to "go the extra mile", to make information explicit and content easily consumable. This means authors make effort on creating *markup*. The *characteristic of the Web documents* we will rely upon is the *semi-structure* given by the markup. A detailed description of the layer cake depicted in figure 1.1 is given by

Cimiano [Cimiano, 2006]. Next we describe the three constituents to be learned in this thesis and explain why they are important for ontology learning.

1.3 Objectives

The aim of this thesis is to use the semi-structure of large amounts of automatically obtained Web documents to acquire ontology constituents beneficial for ontology engineering. We will acquire *terms*, *synonyms* and *sibling relations*. These three ontology constituents can be located in the *ontology learning layer cake* [Buitelaar et al., 2005, Cimiano, 2006] shown in figure 1.1. The ontology learning layer cake distinguishes different levels representing different types of knowledge which are worth acquiring while performing ontology learning.

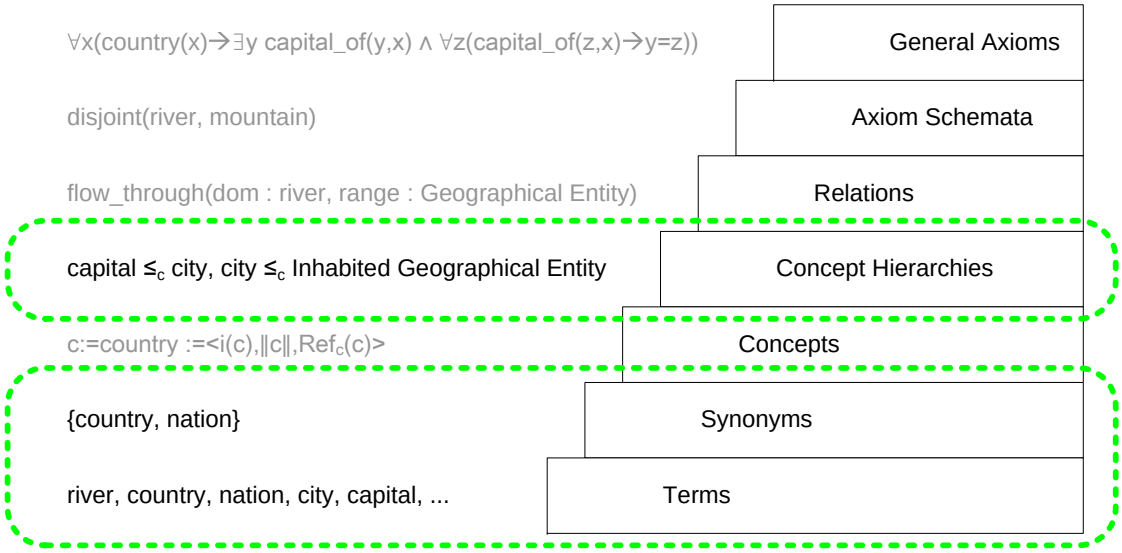


Figure 1.1: Ontology learning layer cake [Cimiano, 2006]. The layers examined in this thesis are highlighted.

Terms: With respect to ontology engineering, terms are the labels of ontological entities such as concepts, instances or relations. They are the signs that depict ontological entities. They are the most basic building ingredients of ontologies and are depicted as layer one of the ontology learning layer cake (figure 1.1). In linguistics, terms form the vocabulary of a domain [Mitkov, 2003]. According to [Bourigault and Jacquemin, 1999], terms correspond to sequences of words, most of the time noun phrases, which are "terminological units". Since terms can contain whitespace, one can further distinguish single word terms and multiword terms. Single word terms are, for example, "ocean" and "water". They do not contain whitespace. The sequence of words "Atlantic ocean" is a multi-term

expression which includes whitespace. Multiword terms, in a similar notion also referred to as multiword expressions [Sag et al., 2002], play a crucial role. The importance of multiword expressions is for example discussed in [Sag et al., 2002]. According to Jackendoff [Jackendoff, 1997, page 156], it is estimated that the number of multiword expressions in a speaker's lexicon is of the same order of magnitude as the number of single words. Jackendoff also notes that this might be even an underestimate, since, for example, 41 percent of the entries in WordNet 1.7 [Fellbaum, 1998] are multiword expressions and that specialized domain vocabularies overwhelmingly consist of multiword expressions. For the English language, multiword expressions constitute a crucial fraction of domain vocabularies. For languages like German, where compounds are heavily used (for example "Tigerhai" for the English "tiger shark"), detecting multiword expressions is less important, but still relevant since there are a number of terms which consist of several words. In general, we conclude that multiword terms are also important for the lexical layer of ontologies.

The field of terminology acquisition investigates methods for acquiring terms from textual content. Even after decades of research, acquiring terms is not easy because the approaches are usually domain and language dependent and require training. In a comparative evaluation of term recognition algorithms, Zhang [Ziqi Zhang and Ciravegna, 2008] notes that there are only 5 approaches which are capable of acquiring single word and multiword expressions at the same time. In this thesis we will illustrate an approach that uses Web documents to obtain terms without the necessity of incorporating training or language specific or domain specific software. The acquired terms include both single word terms and multiword terms.

If the task of acquiring multiword terms is omitted in ontology learning procedures and no vocabulary containing terms is given as input, which is usually the case, the learned concepts and relations have only trivial labels of single words and it is left to the ontology engineer to correct this manually. But even worse, relations between ontology entities labelled with multiword terms are likely to be missed. Since the overall aim of performing ontology learning is to reduce the per entry cost, it is an important goal to acquire and process vocabularies which include multiword terms. All approaches presented in this thesis are capable of handling multiword terms. They can actually handle multiword terms in the same manner as single word terms; no separate processing is necessary.

Synonyms: In ontologies, synonyms are terms which denote the same concept. In linguistics, synonymy [Cruse, 2004, page 154-156] of terms is discussed and several grades are distinguished. Cruse [Cruse, 2004, page 154] distinguishes absolute synonymy, propositional synonymy and near synonymy. The last one, near synonymy, is approached by several methods which try to obtain synonym relations from text. Terms are regarded as near synonyms when they are exchangeable in some contexts. In this thesis we rely on the definition of synonym of Wordnet [Fellbaum, 1998] synsets, where words are regarded as synonyms if they share a

common meaning which can be used as a basis to form a concept relevant for the domain in question.

Knowing that two terms refer to the same concept is important for the ontology engineer for not creating separate concepts which are actually the same. The acquisition of synonyms in ontology learning is reflected by the second layer of the ontology learning layer cake (figure 1.1). The acquisition of synonym candidates is very challenging; there are only a few approaches for doing so. We will show an approach for acquiring synonymous terms from Web documents.

Sibling Relations: Hierarchies of concepts are usually given by hierarchical relations of types such as "is-a" and "has-part". Coming along with such relations are indirect hierarchical relations which are the orthogonal counterparts to direct hierarchical relations. We refer to those relations as sibling relations [Cimiano, 2006, page 109]. The emphasis of this thesis is on acquiring sibling relations which are described in more detail in section 1.4.2 and 1.4.3.

As already stated, in this thesis we describe approaches aiming at obtaining results which belong to three layers of the ontology learning layer cake: terms, synonyms and, with special emphasis, the concept hierarchy layer where the sibling relations reside. But in contrast to many ontology learning approaches, we pursue a different direction while addressing the concept of hierarchy layer. There are numerous methods for the discovery of direct hierarchical relations of subordination. There is less work in discovering concepts that stand in a sibling relation to each other and are the children of a common parent concept. The special emphasis of this thesis on the sibling aspect is reflected by an updated ontology learning layer cake level shown in figure 1.2. In this updated layer cake, the direction of hierarchical and non-hierarchical relations is distinguished and made explicit. In the next section we highlight the importance of sibling relations and provide foundational descriptions.

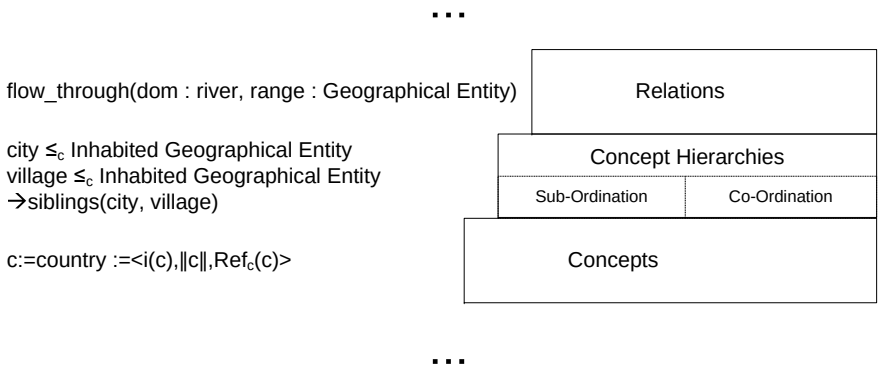


Figure 1.2: Distinguished sub-ordination and co-ordination directions of concept hierarchies within the ontology learning layer cake

1.4 Foundations

In a hierarchy two directions for two different kinds of relations can be distinguished. The most prominent is the *superordination-subordination* [Cruse, 2004, page 134] direction. But accompanying one can observe the orthogonal *coordination* direction. In ontology engineering much emphasis is paid to hierarchical relations of subordination, whereas the superconcept-subconcept relation is the most prominent example of a sub-ordination relation. Orthogonal to superconcept-subconcept are relations between *sibling concepts*, concepts which share a common super-concept. The notion of sibling concepts is mentioned and prevalent in practically oriented instructions on ontology construction such as in [Henze, 2004, Rector et al., 2006, Groh and Toni, 2005] but not within formal ontology engineering methodologies [Sure et al., 2006]. The notion of siblings is used in some approaches for ontology alignment [Ehrig, 2006] and ontology learning [Cimiano and Staab, 2005, Cimiano, 2006].

Next we will explain what we refer to as sibling relations by introductory examples, and then we consider variants of sibling relations which can be defined and described as the limitations of the approaches for finding sibling relations described in this thesis.

1.4.1 Introductory Examples

Figure 1.3 shows an exemplary hierarchy of concepts from the geography domain. The concept `city` and the concept `village` are both inhabited geographic entities. One could also say that `city` and `village` are sibling concepts regarding their common super-concept `Inhabited Geographic Entity`. There are two more sibling concepts `mountain` and `river` as well as `Natural Geographic Entity` and `Inhabited Geographic Entity`.

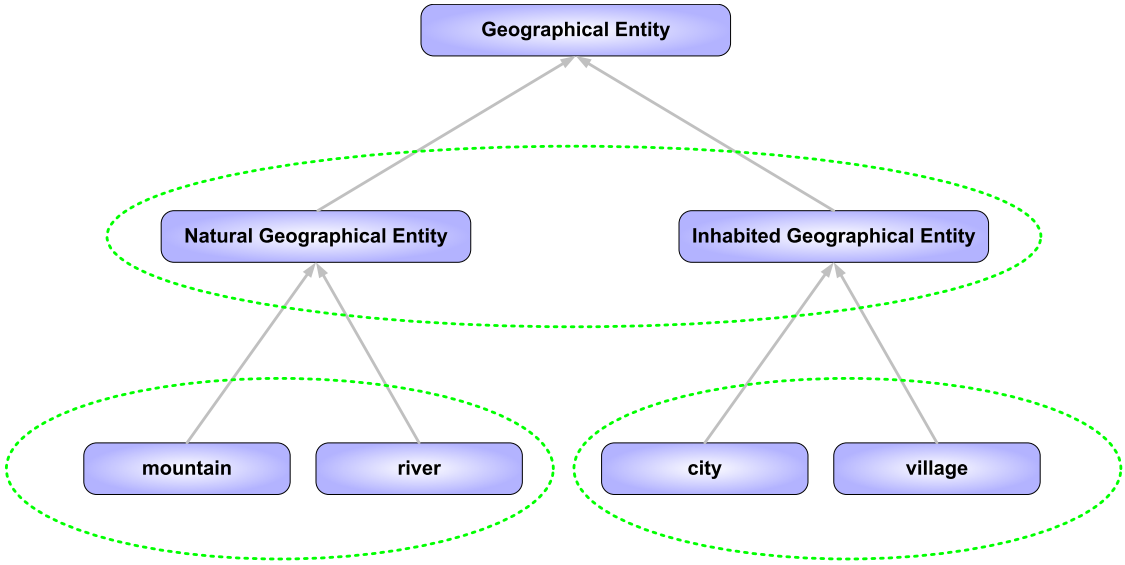


Figure 1.3: Example hierarchy of geographic entities (adopted from [Buitelaar and Cimiano, 2007], shown in appendix A, figure A.1). Sibling concepts are emphasized by doted ellipses.

The aim of this thesis is to find sibling relations. If, for example, the entities depicted by the terms `city, village, Inhabited Geographic Entity, mountain, river, Natural Geographic Entity, Geographic Entity` are given, then the aim will be to find out that between (1) `city` and `village` and (2) `mountain` and `river` and (3) `Natural Geographic Entity` and `Inhabited Geographic Entity` sibling relations exist.

However, ontologies consist not only of hierarchical is-a relations between concepts but are represented by more ontological entities. For example, figure 1.4 shows the exemplary hierarchy of concepts from figure 1.3 and also instances of two concepts. The concept `river` has two instances `Rhein` and `Elbe`, the concept `city` has two instances `Leipzig` and `Dresden`. The instances of a concept are siblings to each other too. The two instances `Rhein` and `Elbe` are sibling instances - regarding their common "type" `river`.

In the next section we discuss several types of sibling relations and describe which of them we actually learn.

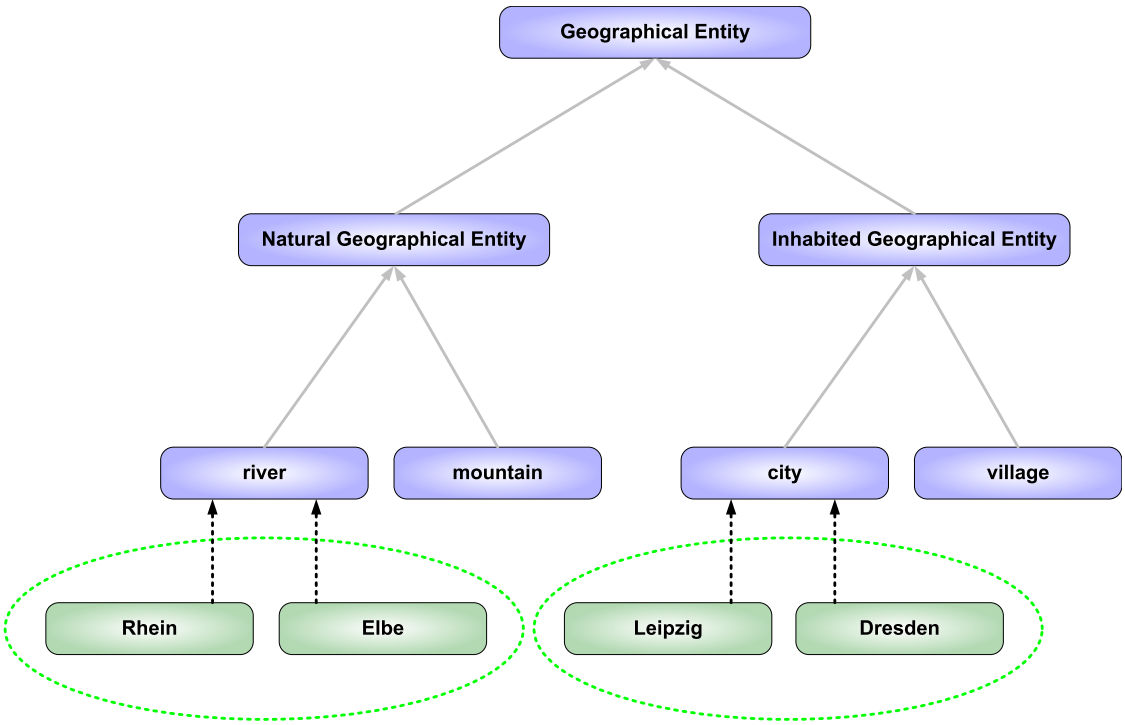


Figure 1.4: Example hierarchy of geographic entities where in addition to the concepts shown in figure 1.3 as blue boxes, instances depicted by green boxes are present.

1.4.2 Notions of Sibling Relations

In the introductory example of the previous section we already described that there are siblings relations among concepts and instances. In an ontological data structure where non-hierarchical relations are also defined, more constellations between entities standing in a sibling relation can be observed.

A comprehensive definition of an ontology structure comprising ontology entities such as concepts, instances, relations, and attributes is given by Cimiano [Cimiano, 2006, page 10 ff]. According to such an ontology structure, different notions of sibling relations among ontological entities can be defined.

1. Concept Siblings: Concepts which are sub-concepts of a common super-concept.

2. Instance Siblings: Instances which are instances of a common concept.

3. Relation Siblings I: Relations standing in sibling relation because they are sub-relations of a common super-relation

4. Relation Siblings II: Concepts (or instances) which are siblings to each other because they are connected by a relation of the same type to another concept

(or instance). For instance `Dresden` might be sibling to `Hamburg` because for both it is stated that the `Elbe` "`flows_through`" those cities.

5. Attribute Siblings: Attributes which are defined for a concept, which have the same "domain".

However, even though such subtle notions of ontology entities standing in sibling relation can be distinguished such a subdivision is not feasible within this thesis for two reasons.

First, from the ontology engineering point of view, it is not clear which of the available ontology entities should actually represent a certain constellation. It depends on concrete ontology engineering design decisions. One might, for example, represent `African Lion` and `Asian Lion` as instances of `Lion` or as subclasses of `Lion`. There are more such design variants where several ontological entity types can be used to model a comparable circumstance. And secondly, a limitation of the approaches we propose is that we cannot provide the parent entities which make up the sibling entities – neither for the listed sibling types in general nor for one of the types such as concept siblings. This means that while one acquires that `Rhein` and `Elbe` are depicting sibling ontology entities, it is not known that they are both a kind of `river`. We do not know what kind of entity the term labels: if it is a concept, an instance, a relation or an attribute. Rough heuristics could be that if the terms are verbs or adjectives, they might rather depict relations and attributes; if they are usually used in uppercase, they are likely to be named entities and thus rather depict instances. But this would not solve the problem to a considerable extent since terms are not always of a single word with a particular POS, let alone the difficulties in obtaining the precise POS. The labelling of the latent parent concept could be approached by incorporating automatic approaches suited for this purpose, but we want do focus on evaluating the quality of the newly proposed methods relying on Web document structure. Furthermore, to decide whether obtained sibling candidates are "parts" of the same entity or if they are sub-concepts is beyond the scope of what the proposed approaches of this thesis can provide. This is often not straightforward for ontology engineers. Ontology learning is known to deliver only rather rough raw results and the ontology engineer is required to add a large part of the engineering efforts by himself. If an approach delivers the hint that there might be a group of entities standing in sibling relation such as `weight`, `height` and `width` he has to decide to represent this according to his representation formalism and is objectives.

Therefore, we restrict our observations within this thesis to *concepts standing in sibling relation* regarding a hierarchical relation. This means that the vocabulary used within the approaches of chapter 4 to 6 are terms which depict the labels of concepts. The quality of the results is judged according to whether the concepts depicted by those terms are standing in a sibling relation according to a common super concept. Thus the measured quality is those of sibling concepts. It has to be borne in mind that if learned candidates are observed as erroneous, they might be valid siblings according to another notion of siblinghood and that for vocabularies

where labels of other entities besides concepts are also present, sibling relations between those other entities are obtained.

1.4.3 Definitions

Next we provide a definition of sibling concepts where concepts stand in a sibling relation because they have a common direct super-concept. First we define the ontology or more exactly a core ontology structure whereupon sibling concepts can be defined.

Definition 1.1 (Core Ontology [Cimiano and Staab, 2005]) *A core ontology is a structure $\mathcal{O} := (C, \leq_C)$ consisting of a set C called concept identifies, a partial order $\leq_C$ on C called concept hierarchy or taxonomy.*

Sibling concepts can then be defined as:

Definition 1.2 (Sibling Concepts) *[Cimiano, 2006, page 109]*

$$Sibling(c, O) := \{c' \mid \exists c'' \; c \prec_C c'' \wedge c' \prec_C c''\} \tag{1.1}$$

$\prec_C$ depicts the immediate predecessor relation. The immediate predecessor relation can be defined as follows:

Definition 1.3 ($\prec_C$ [Cimiano and Staab, 2005]) *$c' \prec_C c$ iff $c' \leq_C c$ and there is no c'' such that $c' \leq_C c''$ and $c'' \leq_C c$.*

About the characteristics of sibling relations it has to be remarked that the sibling relation is a *symmetric* relation. Sibling relations are *not transitive*.

Example 1 *In the examples of the layer cake (and figure 1.3), we can conclude from city $\leq_C$ InhabitedGeographicalEntity and village $\leq_C$ InhabitedGeographicalEntity that there is a sibling relation between city and village. In this case they are siblings on account of being both a InhabitedGeographicalEntity.*

Furthermore, a concept is labelled by signs which we denote as terms t. For our approaches of 4 to 6 we restrict that a concept is labelled by only one term. Furthermore, we ignore polysemy/homonymy and assume that a term refers to only one concept, an assumption which can be made within a narrow domain of interest. Thus a term denotes a concept. The terms which are the labels of two concepts standing in sibling relation are referred to as *sibling terms*. In chapter 8 we also consider the circumstance that concepts can be labelled by more than one term where we consider synonymous terms. We do not refer to terms which are synonyms as sibling terms since we base siblinghood on concepts being siblings.

1.4.4 Sibling Relations beyond Ontologies

The notion of entities standing in sibling relation is prevalent in disciplines of computer science other than ontology engineering. Sibling relations among lexical constructs are known from linguistics. It is important to know since the border between ontology learning and knowledge acquisition for linguistics is often vanished. The lexical hypernym-hyponym (hyponymy: [Cruse, 2004, page 148-150] relation of noun terms provides a relation between the hyponym noun which is more special than the general hypernym. Also between meronyms (parts) of a holonym (whole) (meronymy: [Cruse, 2004, page 150-154]) a subordination relation is observed. Orthogonal to the subordination direction, the co-ordination direction can be observed. Subsequently, *co-hyponyms* [Lyons, 1977] and [Cruse, 2004, page 161]) are hyponyms of a common hypernym. Co-hyponyms are often referred to as "coordinate" [Anderman and Rogers, 1998, page 19-20] especially in the Wordnet [Fellbaum, 1998] terminology. *Co-meronyms* [Cruse, 2004, page 162] refer to the meronyms of a common holonym. In our related work, described in chapter 2, we will, therefore, regard approaches which aim at finding coordinates as related work intending to find sibling relations.

However, sibling relations are also observable in a quite pragmatic way: for many real world circumstances, it is not the super-concept that is of interest but a group of entities which have something in common and which are thus siblings to each other. This is colloquially denoted as categories. Categories are a natural way of observing the world. Humans have a tendency to structure things into categories. In a widespread quiz, young children learn to recognize things which do not belong into a certain group of things. Consequently, categories and categorization are object of research in psychology [Mervis and Rosch, 1981, Markman, 1989, Murray and Reuter, 2005]. Also in more concrete application fields like Geography are categories investigated [Smith and Mark, 1999].

There are numerous examples of categories which are used to structure entities. Categories where sibling characteristics are present are frequently used to structure entities such as products on e-commerce Web sites. Figure 1.5[1] shows exemplary categories from Amazon's hierarchy of categories of book topics. Categories have been an important structuring tool not only in the Web, but also before the emergence of the Web. In library science one could find a hierarchy of topics, the Dewey decimal system, a hierarchical system of categories.

[1] http://www.amazon.com, Screenshot taken on November, 4th 2008

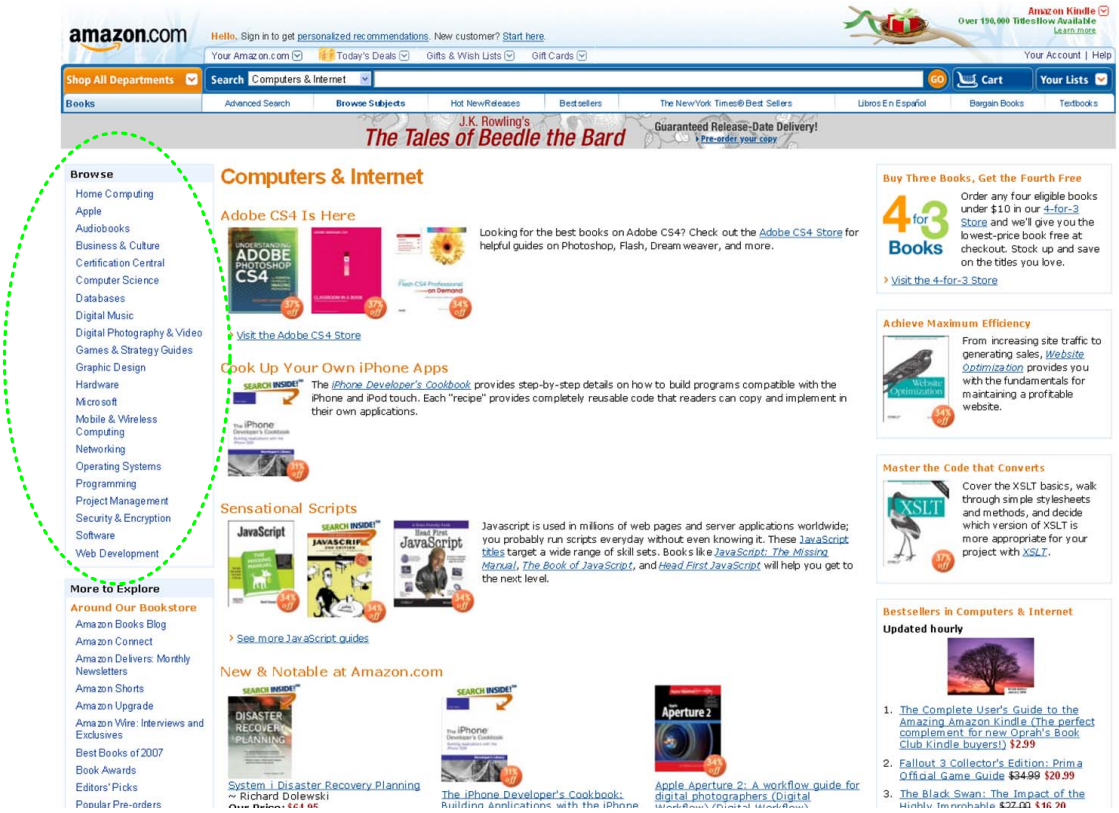


Figure 1.5: Exemplary usage of sibling items on an e-commerce website

1.5 Outline

In this section we give an overview of the following chapters of this thesis. Figure 1.6 gives an overview of the relations of the chapters.

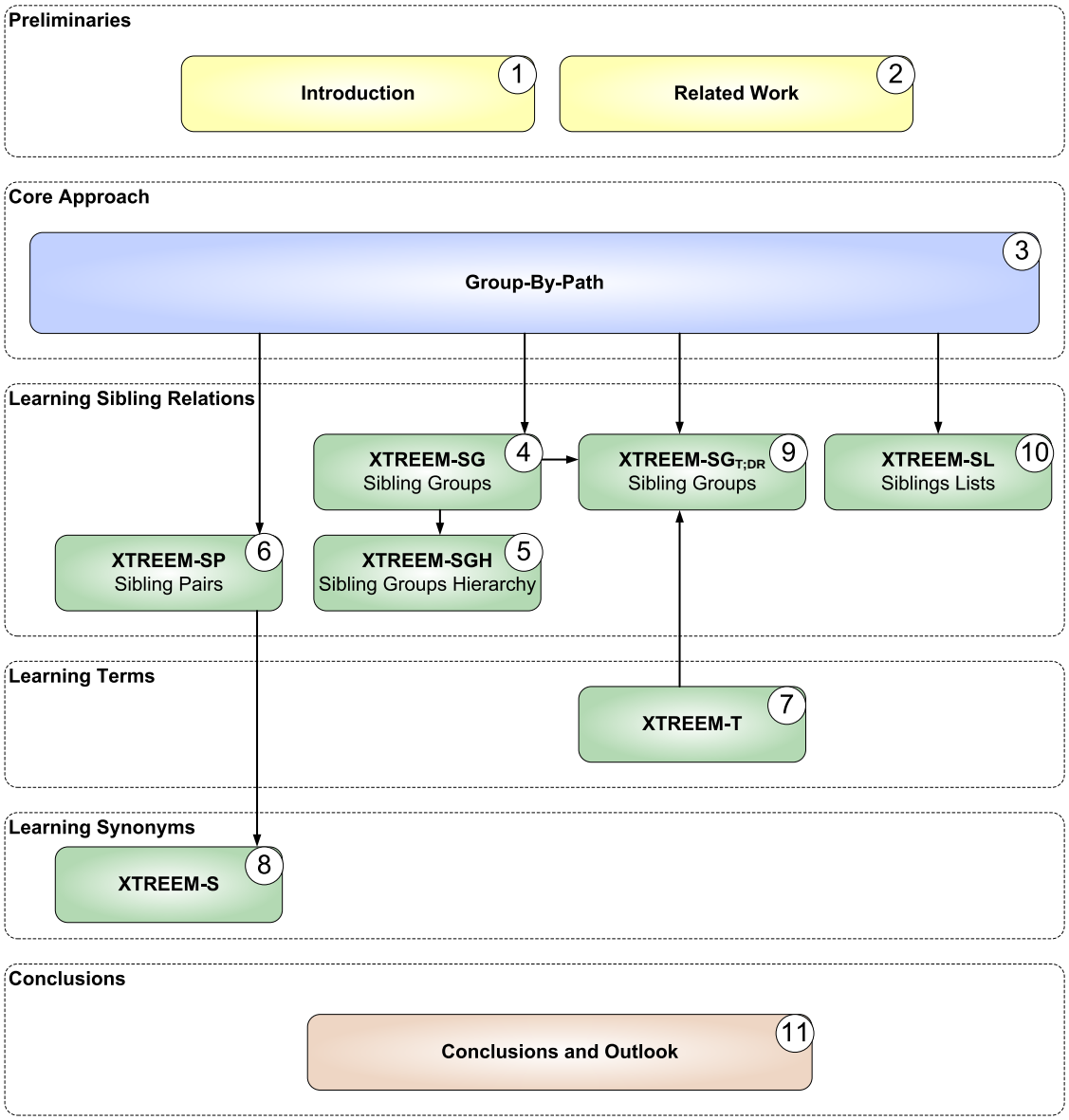


Figure 1.6: Thesis overview. Dependencies between chapters.

- **Chapter 2** provides a description of *Related Work*. The described related work is focused on *ontology learning from Web documents* in general, on approaches exploiting particular characteristics of semi-structured Web documents as well as on approaches for finding sibling relations. Related work on the Group-By-Path operation, will be presented in chapter 3 after the Group-By-Path operation has been described. Chapter 7, 8, 9 and **??** have a separate section on related work.

- **Chapter 3 - Group-By-Path:** In this chapter the core approach used by all solutions presented in the subsequent chapters is described. The Group-By-Path approach is the core method for accessing semi-structured Web documents proposed in this thesis. This Group-By-Path approach enables the acquisition of terms which stand in a sibling relation. In contrast to the established Bag-Of-Words model, the Group-By-Path operation considers the tree structure of semi-structured Web documents. The Group-By-Path operation was initially published in [Brunzel and Spiliopoulou, 2005, Brunzel and Spiliopoulou, 2006a].

The objective of the next 3 chapters, chapters 4 to 6, is the acquisition of sibling relations. Those methods, therefore, belong to layer 4 of the ontology learning layer cake.

- **Chapter 4 - Learning Sibling Groups - XTREEM-SG:** The first, and probably the most important, solution described in this thesis is XTREEM-SG procedure. The XTREEM-SG procedure uses flat clustering to structure given vocabularies into sibling groups while Web documents are used as input. We investigate how variations on input, parameters and gold standard influence the obtained results. By means of a gold standard evaluation we show that the state of the art results have been improved significantly. This chapter is based on work published in [Brunzel and Spiliopoulou, 2006c, Brunzel and Spiliopoulou, 2008].

- **Chapter 5 - Learning Sibling Groups Hierarchies - XTREEM-SGH:** In this chapter we use a different type of clustering techniques for structuring a given vocabulary into a hierarchy of sibling groups. In chapter 4 the ontology engineer has to inspect a potentially large number of clusters. We want to improve this situation by applying hierarchal clustering so that the clusters provide additional structure. We apply Bi-Secting-K-Means on the same dataset as used in chapter 4 and obtain a hierarchy of sibling groups. This chapter is based on work published in [Brunzel, 2007].

- **Chapter 6 - Learning Sibling Pairs - XTREEM-SP:** In this chapter we process a Web document collection with the Group-By-Path approach and perform association mining to find sibling pairs. Similar to the XTREEM-SG approach described in chapter 4, a closed vocabulary is structured

into pairs of sibling terms. The evaluation shows recall and precision curves while comparing the obtained results against reference ontologies. Furthermore, we investigate how variations on input and parameters influence the obtained results. This chapter is based on work published in [Brunzel and Spiliopoulou, 2006b, Brunzel and Spiliopoulou, 2007a].

In the next two chapters, the two basic layers of the layered ontology learning process are tackled, the acquisition of terms and the acquisition of synonyms.

- **Chapter 7 - Extracting a Vocabulary with XTREEM-T:** With the XTREEM-T procedure vocabularies of terms can be acquired from Web document collections. By means of frequency based sorting, the most frequent text spans formed by markup boundaries are supposed to be valid terms which can be considered useful within the domain of interest. In an exemplary manual evaluation we investigated the degree to which this is indeed the case. As such, this chapter gives an impression of what feature space on a Group-By-Path vectorization is likely to be constituted, which is important in real world scenarios where the XTREEM-SG (4), XTREEM-SGH (5), XTREEM-SP(6) or other Group-By-Path based approaches are applied on open vocabularies and not on high quality vocabularies given as input as done within the experiments of the chapters 4 to 6. In general XTREEM-T is not bound to the application of Group-By-Path involving approaches; it can be applied to acquire a vocabulary in scenarios where no Group-By-Path is involved at all. In contrast to established term acquisition methodologies, it is appropriate to be applied on Web documents by default. It belongs to the rare number of approaches for terminology acquisition which are capable of acquiring single-word and multiword terms at the same time. This chapter is based on work described in [Brunzel, 2008].

- **Chapter 8 - Finding Synonyms with XTREEM-S:** By means of the XTREEM-S procedure we aim at identifying synonyms. This approach is related to the XTREEM-SP approach described in chapter 6. For the sake of obtaining synonyms, a further processing iteration for computing associations is conducted, as it is often done for finding synonyms in established methods. But in contrast to established methods for finding synonyms, XTREEM-S relies on a Group-By-Path dataset. For evaluation we use reference synonyms from Wordnet [Fellbaum, 1998]. This chapter is based on work described in [Brunzel, 2008].

In the next chapter also sibling relations are obtained, but in contrast to the approaches of chapter 4 to 6, it is not a given vocabulary that is processed but the open vocabularies.

- **Chapter 9 - Domain Relevance enhanced Term Weighting for Learning Sibling Groups - XTREEM-SG$_{T,DR}$:** The Domain relevance

enhanced term weighting is a term weighting schema which, in addition to the internal term occurrence distribution, includes information about external term occurrence distributions. The domain relevance enhanced term weighting is supposed to yield cluster labels which are constituted by domain relevant terms to a higher extent than without domain relevance enhanced term weighting. We apply the proposed term weighting on a Group-By-Path based dataset. By means of several measures we determine the extent to which the term labelling clusters are characteristic for a domain in comparison to the general language. This chapter is based on work published in [Brunzel and Spiliopoulou, 2007b].

- **Chapter ?? - Web-scale Indexing and Retrieving of Sibling Terms with – XTREEM-SL:** The XTREEM-SL establishes an index over large amounts of sibling groups obtained by applying the Group-By-Path operation. XTREEM-SL is an approach to obtain a list of sibling terms for given input terms in an ad-hoc time frame. Here an open vocabulary is used; no given input vocabulary is required. The XTREEM-SL process consists of two sub-processes and time consuming offline process and the quick retrieval process. We evaluate against reference ontologies and show exemplary results where we contrast the obtained results with another approach.

- **Chapter 11 - Final Conclusions:** In this chapter we summarize the main contributions and conclusions, and provide an outlook on future research.

2 Related Work

There is a plethora of approaches and methods which can be regarded as relevant for the field of ontology learning. Comprehensive overviews and surveys on ontology learning can be found in [Gómez-Pérez and Manzano-Macho, 2003, Shamsfard and Barforoush, 2003, Shamsfard and Barforoush, 2004, Biemann, 2005, Zhou, 2007]. In this chapter we focus on related work on ontology learning from the Web, respective semi-structured Web documents as well as on approaches aiming at finding sibling relations.

One major distinction on ontology learning approaches is the degree of structure that can be assessed on the input data.

There are methods which can be considered as *ontology learning from structure*, also referred to as lifting [Volz et al., 2003]. Approaches for ontology learning from structure use well-structured resources to infer ontological knowledge. Such valuable sources of data which are used for learning are, for instance, database schema [Gottgtroy et al., 2003], XML-DTD's, XML schema [Cruz and Nicolle, 2008] or UML diagrams, knowledge bases [Suryanto and Compton, 2001] and dictionaries [Rigau, 1994, Jannink and Wiederhold, 1999]. However, such structures are rare; they are not available for arbitrary domains and topics. Subsequently, the poor availability of suited data structures to be used drastically limits the applicability to rare cases. But, whenever available, such structures can be reused.

On the other side of the spectrum, regarding the structure among input data, is plain text. Indeed, the research on ontology learning is mainly focused on learning from unstructured plain text. There are three major paradigms for ontology learning from text, *lexico-syntactic patterns* [Hearst, 1992], *Harris' distributional hypothesis* [Harris, 1954] and *term subsumption* [Sanderson and Croft, 1999]. Relevant for obtaining sibling relations are the lexico-syntactic patterns which are described in section 2.3 and Harris' distributional hypothesis used for clustering in general and for finding synonyms in particular, described in chapter 8.

While learning from Web documents, a coarse separation can be undertaken between approaches which use the Web to obtain documents where the available markup-is removed, described in section 2.1, and between approaches which rely on the markup described in section 2.2. In section 2.3, we will focus on approaches aiming at finding sibling relations.

2.1 Learning from the Web

In this section we describe approaches which obtain documents from the Web, as we do within our approaches, but where, in contrast, the Web document markup is not used. Those approaches do not rely on the semi-structure of Web documents but remove the HTML markup. They are in principle not restricted to semi-structured HTML Web documents as, for example, PDF documents available from the Web are also processed. Such approaches usually rely on publicly accessible Web search engines for obtaining references of Web documents. Web search engines provide an index over large amounts of Web documents allowing for two different types of usage. On the one hand, there are approaches using *entire Web documents* and, on the other hand, there are approaches which use only special parts as the *snippets* returned by search engines.

The Web documents are obtained by creating queries which obviously point to documents adhering to particular topics/domains. The Web documents are additionally downloaded from the Web and the HTML markup is stripped. Those documents are then processed by the various methods known for processing textual content such as co-occurrence analysis (for example [Agirre et al., 2000, Faatz and Steinmetz, 2002, Junichiro et al., 2004, Chung et al., 2006]) or natural language parsing (for example [Alani et al., 2003, Navigli, 2005, Kathrin Eichler and Neumann, 2008]). Such approaches additionally incorporate various language specific resources such as stop words, stemmers, sentence splitters or POS taggers. This dependency makes those approaches language dependent and even worse, the text obtained from Web documents is not as pure as those text where, for example, sentence splitters and parsers are typically built for. The number of documents processed by those approaches is also rather low, in the range of dozens to a few thousands, compared to the number of potentially available documents for the domain of interest. Especially the approaches relying on computationally expensive techniques such as deep parsing are problematic since for such approaches even hundreds or thousands of documents are consuming a lot of time. There are attempts to improve this situation by using less complex techniques; for example, only shallow parsing instead of full parsing [Sazedj and Pinto, 2007].

Other approaches only use special Web documents such as news [Sung et al., 2008] or product catalogue web sites [Ye and Chua, 2006, Labský et al., 2005] or Wikipedia [Ruiz-Casado et al., 2006, Herbelot and Copestake, 2006, Suh et al., 2006, Suchanek et al., 2006, Ponzetto and Strube, 2007]. Such approaches are only weakly related since approaches are not designed to work with arbitrary inhomogeneous Web documents as in the approaches described in this thesis but with Web documents where a high regularity can be observed.

The other type of approaches relies on processing only snippets of Web documents. The snippets can be obtained from the Web search engine directly without the need to download entire Web documents. And the amount of text

to be processed is much lower, also allowing for processing with more complex techniques such as parsers [English and Nirenburg, 2007]. For such approaches creating queries which yield suitable snippets as outcome is important. The most often used approaches of this type are the ones which use Hearst patterns [Hearst, 1992]. Such approaches are described in more detail in section 2.3 since those patterns are capable of obtaining sibling relations. By applying such Hearst patterns on the Web, the drawback of Hearst patterns, the low recall can be reduced.

2.2 Learning from HTML Documents

The approaches described in the previous section relied on processing plain text which was obtained by removing available HTML markup. By doing so a potential added value of Web documents was removed as well. In this section, in contrast, we describe approaches which rely on the semi-structure of Web documents. Semi-structured Web documents here refer to HTML documents with its degree of structuring lying in between 'structures' and plain text.

A coarse separation can be made between approaches which use various tags in a generic way, described in section 2.2.1, and approaches which focus on particular HTML building blocks. Major types of HTML building blocks to be exploited are tables, described in section 2.2.2, headings described in section 2.2.3 and lists described in section 2.2.4.

2.2.1 Markup in General

In this section we describe systems which use the Web document markup regardless of the tag function as it is done by our approach too and regardless of the kind of results aimed at. The difference is that the tree structure which our approach relies upon is not used by the methods described in this section.

The approach of Kruschwitz [Kruschwitz, 2001a, Kruschwitz, 2001b] uses markup sections of Web documents to learn a domain model. From the occurrence of a term in several markup sections he concludes that such a term is more important than other terms. The markup used includes `<meta>`, `<head>`, `<title>` or emphasizing tags as `<b>` or `<i>`. Related terms can, for example, be used to refine search queries.

The approaches of Karoui et al [Karoui et al., 2004, Bennacer and Karoui, 2005, Karoui et al., 2007] present an approach where terms are hierarchically clustered according to their context. As context they use regular co-occurrence within a sequence of text but they also consider the co-occurrence of words across the boundaries of HTML tags for several HTML tags where a dependency is observed. For example, they state that there are dependencies like `<h1>` → `<p>`, `<caption>` → `<td>`, `<dt>` → `<dd>`, `<TITLE_URL>` → `headings of a part of document`, `<TITLE_URL>` → "headings of the referenced document", `<TITLE>` → "headings of

the document" and such dependencies as those of two emphasized terms within the same HTML block. The last dependency of emphasized terms within the same HTML block is especially relevant to our work since such terms would also be acquired by our approach. But our approach is not restricted to particular often used together HTML blocks, but uses HTML structuring in a more generic way where dependencies between the nested HTML tags are not required.

Manzano-Macho et al [David Manzano-Macho and Borrajo, 2008] use the co-occurrence of words in `title`, `keywords`, `meta`, `headers` and highlighted information (bold, different type cases) to obtain frequent collocations where the terms appear in the same unit/block. For doing so they use the notion of semantic textual units proposed by [Buyukkokten et al., 2001]. They use as a hint the notion of "in-the same-hierarchy", where words occurring at the same level of indentation or within two consecutive list items are considered. This is related to the way we access Web document structure but we only use entire marked-up text sequences, and not the words constituting the text block.

2.2.2 Tables

Tables are places where information with a high degree of structuring can be found. But tables can contain unstructured information as well and tables are often used for layout purposes, not representing tables suited for extracting knowledge. Subsequently, the extraction of knowledge from tables is not simple. There are a couple of approaches ranging from those which focus on single tables which are displayed in the users browser [Bagni et al., 2007] up to the ones using all tables crawled by a major Web search engine [Cafarella et al., 2008]. The goals here vary, for example, integrating the obtained data [Tijerino et al., 2005], extract F-logic frames [Pivk et al., 2005] or creating an index over large numbers of tables [Cafarella et al., 2008].

While automatically processing large numbers of HTML tables, a problem that emerges is to distinguish between meaningful and decorative tables as, for example, done by Jung and Kwon [Jung and Kwon, 2006]. Meaningful tables include valuable information, in contrast to decorative tables which, for example, split the browser window into a navigational and textual part. Subsequently they try to extract the table head. They observe that decorative tables often contain many links and pictures, many different cell sizes, empty rows or columns, highly customized borders, intermediate cell spans, etc. In contrast to this, meaningful tables often contain textual information and numeric columns or rows. They also observe that missing `<th>`-tags are often compensated by `<b>`- and `<font>`-tags in the first row or column. From their observations Jung and Kwon generate heuristics and apply machine learning techniques to build a table classifier which decides whether a table is meaningful or not and extracts the identified table head for further usage.

Cafarella et al [Cafarella et al., 2008] uses a classifier to obtain 154 million tables that are supposed to contain high quality data from some 14.1 billion tables. They

create corpus wide statistics on co-occurrences of table schema (header) elements. This approach is related since it belongs to the small number of approaches using a large number of Web documents on one the side and, more importantly, the header elements are standing in sibling relation to each other; the header items they track are a considerable subset of the items we process in our approaches.

2.2.3 Headings

Approaches using headings of Web documents are related to our work for the following reasons. First, we consider markup and headings as very informative tags. The second reason, which is related to the first reason, is that the extraction of semantics from headings is a promising task. Further, in such approaches, just as in the case of our approach, headings are used as an entire span of text – in contrast to splitting such sequences into words or terms as done by most other approaches on processing textual content.

Makagonov et al. [Makagonov et al., 2005] present a method which aims at finding subordination relations between topics and subtopics. They exploit the fact that documents are often hierarchically structured and that this can be used to infer subordination relations subsequently. They rely on the circumstance that words occurring in more general titles subordinate the words occurring in the texts described by these titles. For this purpose they use the titles and the main text of the HTML `h1` to `h6` tags that mark the headers, sub-headers, sub-sub-headers, etc. The learned "ontology" is directly reflected by the hierarchy level of the found topics. As an advantage they state that this approach can be used with only a small amount of available data.

Hazman et al [Hazman et al.,] use the headings from a small number of documents. They use the hierarchical structure given by HTML headings for discovering the children of a root concept. From 87 documents they extracted 3191 headings.

2.2.4 Lists

Shinzato and Torisawa [Shinzato and Torisawa, 2004] present an approach which aims at finding hyponym-hypernym relations from Web document collections. Their approach does not primarily intend to extract sibling relations but as an intermediate step they use "hyponym candidates that may have a common hypernym". This could be referred to as coordinates or co-hyponyms. As candidates they use words or phrases that appear as list items of the same list. They use both ordered and unordered HTML lists. Their acquired co-hyponyms are a subset of the siblings we acquire from Web document lists since they use a different notion of deciding of what is included in such a candidate co-hyponym set compared to the approach we will present in chapter 3. They only use list items which are neighbours to each other and belong to one list. They extract list items if the number of list items is at least 4 and less than 20. From 871,000 HTML

documents they extracted 90,200 candidate co-hyponym sets. Their approach in the subsequently steps aims to extract a corresponding hypernym for the co-hyponym sets. Their approach applies a condition where they exclude 70 repeatedly re-occurring list items such as "help" and "links" which they have manually obtained. Such items are regarded as not being semantically related to the other list items. In general this approach can be regarded as the related work which is closest to our approach. They use a large number of Web documents and they use items occurring together in a manner that is exploited in a way related to our Group-By-Path approach. Because of that their approach acquires the subset of sibling terms which occur within HTML lists compared to what we will acquire from Web documents. We required the terms not to occur as close as neighbour HTML list items, nor do we restrict our acquisition to HTML list but acquire terms regardless of the HTML tag/block types.

2.3 Learning Sibling Relations

In this section we focus on approaches appropriate for learning sibling relations regardless of the type of used input information, thus also covering methods using plain text as input. From the methods for ontology learning from text, the Hearst style lexico syntactic patterns [Hearst, 1992], and Harris' distributional hypothesis [Harris, 1954] are the major paradigms used for obtaining sibling relations. They can be applied on plain text as well as on Web documents.

Most of the approaches described later use the linguistically originating expressions such as co-hyponyms or coordinates to refer to term constellations which we refer to as sibling terms depicting sibling entities/concepts.

A frequently used strategy for extracting embedded relations from natural language texts is based on the use of language style patterns. Such patterns are called lexico-syntactic patterns, sometimes also referred to as Hearst patterns [Hearst, 1992]. Such patterns are suited for acquiring sibling relations in the form of co-hyponyms.

Lexico-syntactic patterns make use of Part of Speech (POS) Tagging while focusing on Noun Phrases (NP). Patterns are for example:

1. NP_0 *such as* $\{NP_1,\ NP_2, \ldots (\ and|or)\}\ NP_n$
2. $NP_1\{,\ NP_2,\ NP_3, \ldots\}$ *and other* NP_0

Such patterns match phrases as those shown in the following two examples:

(1) ... *dangerous sharks* **such as** *great white sharks*, *hammerhead sharks* **and** *tiger sharks*...

(2) ... *great white shark*, *hammerhead shark* , *tiger shark* **and other** *dangerous sharks*...

To a reader such expressions imply that NP_0 is a hypernym of NP_i, but accompanying NP_i, a sibling relation, namely co-hyponymy, can be observed. In other words, NP_i depicts s sibling group. In the example one observes the sibling relations: *co-hypernym(great white shark, hammerhead shark)*, *co-hypernym(great white shark, tiger sharks)* and *co-hypernym(tiger sharks, hammerhead shark)*.

Even though Hearst patterns are intriguing by their simplicity and low computational costs, they traditionally suffer from their low coverage even in relatively large text document corpora. Patterns which reliably indicate the relation of interest occur rarely while frequent patterns are not reliable enough. This disadvantage is less serious when applying lexico-syntactic patterns to huge document collections such as the Web as, for example, done by Ciminao et al [Cimiano et al., 2004a] Ciminao and Staab [Cimiano and Staab, 2004] and Sanchez [Sanchez and Moreno, 2006, Sanchez and Moreno, 2008b, Sanchez and Moreno, 2008a, Ruenes, 2007].

Besides the already stated patterns, Hearst presented more patterns as well as a method for finding such patterns, which can increase the overall coverage [Hearst, 1992]. He also postulated that many other lexical relations could be acquired in the same way. The principle of lexico-syntactic patterns has been adopted and refined in many approaches. A major concern here is to learn new patterns as, for instance, performed by Alfonseca and Manandhar [Alfonseca and Manandhar, 2002] and by Morin and Jacquemin [Morin and Jacquemin, 2004]. The adoption of existing generic patterns to domain specific document collections is pursued, among others, in the Caméléon system by Aussenac-Gilles and Jacques [Aussenac-Gilles and Jacques, 2006].

Riloff and Shepherd [Riloff and Shepherd, 1997] presented the idea of using *noun conjunctions* as the way to obtain members of the same semantic category. Their notion of semantic category is what we refer to as sibling group. Noun conjunctions are patterns where nouns are combined by a conjunction word like “or” and “and”. The noun conjunctions are often terms supposed to stand in a sibling relation. Such patterns are for example “cats and dogs”, “tigers and lions”, “summer or winter”. Starting with a term, sentences which contain these terms are identified. Those sentences are then parsed and noun phrases which stand in conjunction are identified. For those candidates they computed a kind of conditional probability about how often a term occurs together with another term in conjunction compared to the overall number of occurrences. This score is used to rank the candidate terms and the top-n related terms are added to the sibling group. By doing so, they iteratively grow a list of given seed terms by terms supposed to belong to that sibling group. This approach also, is, therefore, in line with the family of approaches relying on contextual proximity of co-occurrence, but they limit the context to surrounding words which are supposed to be more semantically related. This approach was for example refined by Roark and Charniak [Roark and Charniak, 1998] while incorporating a statistical parser.

The approach of Riloff and Shepherd [Riloff and Shepherd, 1997] described above is also the basis for Caraballo’s method [Caraballo, 1999]. For each term

Caraballo's method creates a vector where the number of nouns co-occurring within conjunctions is stored. Those vectors are then clustered by means of agglomerative hierarchical clustering. The hypernym found most often is chosen as the label of the cluster. In the next step, intermediate clusters which could not be labelled are eliminated from the cluster hierarchy. The results are manually evaluated by presenting a random choice of clusters and the hypernym cluster label to three human judges.

Ciminao and Staab [Cimiano and Staab, 2005] presented a guided clustering algorithm where terms are clustered together only if they are known to stand in sibling relations. One source for obtaining sibling relations is the application of Caraballo's method. The guided clustering algorithm is a rare example of an ontology learning approach where sibling relations are explicitly obtained, in contrast to approaches which rather rely on the linguistic notion of coordinates.

In a similar but opposed way to noun conjunctions, the Web has also been used to query for patterns where the terms are connected by a disjunction expression. Ohshima et al [Ohshima et al., 2006] search for "X OR" and "OR X" whereas X is the seed term for which conjunctions should be obtained. The terms retrieved as neighbours are likely to stand in a coordination relation to the given term.

Widdows and Dorow [Widdows and Dorow, 2002] describe an algorithm which relies on an association graph. The association graph was generated based on noun conjunctions extracted from text corpora. Their algorithm aims at adding the most similar node to an existing collection of nodes to incrementally build a stable cluster. This algorithm is supposed to be effective by avoiding the introduction of "infections" [Roark and Charniak, 1998]. Infections herein are out-of category words which distort the character of the term list. Infections are caused by spurious occurrences and by ambiguity. Their notion of category is similar to our notion of sibling group; their term list corresponds to a list of terms supposed to stand in sibling relation. Their approach can process a closed vocabulary of single word terms.

Cederberg and Widdows [Cederberg and Widdows, 2003] describe an approach where they find candidates of coordinates by means of lexico syntactic patterns, whereas their "coordinates" correspond to what we call terms standing in sibling relation. The found candidates are processed by Latent Semantic Analysis (LSA) [Deerwester et al., 1990]. By using LSA they can show that they could substantially improve the precision of their method. Their approach is limited to single word terms standing in coordination relation.

We do not particularly aim at finding named entities, but the siblings found by our approaches can be that of named entities of a particular named entity type. For example, our approaches might yield results such as `Bach, Mozart, Beethoven`. As already pointed out, we do not know the "nature" of the sibling relation but in such a case, the nature could be assessed as "`composers of classic music`", which could be regarded as a particular named entity type. As such approaches for obtaining named entities are relevant since the named entity categories can be made very fine finally denoting something as concepts where the named entities are

what in ontology structures are regarded as instances. One can say that among the named entities of the same named entity type also a sibling relation exists. And if the vocabulary given as input to our approaches of chapter 4 to 6 is comprised of terms depicting named entities, then the terms are likely to be grouped according to a named entity type, though the actual type is not revealed. Such an approach is not unrealistic. By doing so one could aim at results which rather belong to the knowledge base part of an ontology also refereed to as *ontology population*.

Next we mention approaches which extract named entities from large amounts of Web documents. This circumstance makes those approaches relevant to the fact that those approaches have been the rare approaches which at the time when our method was initially published used large amounts of Web documents for finding relations. Etzioni et al describe the KnowItAll system [Etzioni et al., 2004, Etzioni et al., 2005]. In the KnowItAll named entities are extracted from large Web document collections by means of lexico syntactic patterns. First they locate lists of instances. Then a wrapper is generated for each list. Those wrappers are used to match further lists. The results are compared to the results obtained from Google Sets. Their approach is dedicated to rather generic categories of named entities like movies and scientists. Another approach for finding named entities on large Web document collections is described by Pasca [Pasca, 2005, Pasca, 2004]. Pasca uses Hearst patterns in combination with POS-tagging to extract named entities from Web documents where the HTML markup is removed. Their approach relies on capitalized nouns as hints for identifying the terms to be considered. This is appropriate for named entities in English, but would not, for instance, work for German where nouns are capitalized in general and this prevents the application of this approach to non-capitalized terms in general. This approach was refined from coarse grained named entity types to more fine grained named entity types in [Pasca, 2008].

Heyer et al [Heyer et al., 2001] and Biemann et al [Biemann et al., 2004a, Biemann et al., 2004b] discover co-hyponymy relations from plain texts with the use of association measures computed from the co-occurrence of words. Their approach relies on the processing of large text document collections with association measures to compute collocations. Among the discovered relations could be found co-hyponym relations. The approach of Biemann et al [Biemann, 2003, Biemann et al., 2004a] also considers so called X-onyms. X-onyms are collocations of a higher order. Collocations of higher orders are collocations found in collocation sets of 1 order less. They state that co-hyponyms have a higher degree of collocation significance. Co-hyponyms for a term are made apparent within a visualization where terms are placed in a Cartesian plane according to the association strength they are related to [Biemann et al., 2004a, Biemann et al., 2004b]. In a particular region co-hyponyms can be observed.

There are two approaches already described in the section about related work using the Web document markup which acquire sibling relations. The approach of Cafarella et al [Cafarella et al., 2008] described in section 2.2.2 acquires the headers of tables from large amounts of Web documents. This approach is related since the

header elements of a table can be regarded as standing in a sibling relation. Many of the header elements they acquire are likely to be acquired by our approaches too. The approach of Shinzato and Torisawa [Shinzato and Torisawa, 2004] described in 2.2.4 intends to find hyponym-hypernym relations from Web document collections. This approach acquires sibling relations in an intermediate step too. They group list items which belong to one HTML list. Our approach will also acquire those list items but in a more broad sense. In the following chapter we describe our core approach.

3 Group-By-Path

In this chapter we present the Group-By-Path approach for obtaining text sequences which are siblings due to structural regularities within Web documents. The assumption is that structural regularities within Web documents are indicative of "semantics" and that large numbers of "structural sibling text sequences" can be processed to make semantic sibling relations apparent. The validation of this hypothesis is the objective of the following chapters. The Group-By-Path approach depicts the core principle for several processes to find semantic sibling relations from large numbers of Web documents. In this chapter we present the core approach centred at obtaining structural siblings from one Web document.

First we briefly describe the foundations of Web documents and describe which Web documents we actually refer to. Then we describe the structural regularities we rely upon and describe how the Group-By-Path approach obtains sibling text sequences. Afterwards we describe related work.

3.1 Web Document Structures

The emergence of the World Wide Web (WWW) [Berners-Lee et al., 1992], was facilitated by the creation of the HyperText Transfer Protocol (HTTP) and HyperText Markup Language (HTML) [Raggett et al., 1997]. HTML is a standard for representing Hypertext documents. HTML enables one to markup textual contents by so called "tags". HTML documents are also often called semi-structured documents [Abiteboul, 1997] because they consist of a mix of unstructured texts and structures created by HTML tags. For instance, HTML tags can be used to highlight important phrases as shown in figure 3.1. Tags are also used for indicating headings as shown in figure 3.2.

```
...
<p>... In the following section you can find a description
of oceans, the <strong>Atlantic Ocean</strong>, the
<strong>Pacific Ocean</strong>, the <strong>Indian
Ocean</strong>, ...
</p>
...
```

Figure 3.1: Highlighted terms in an exemplary HTML Web document

```
...
<h2>Great White Shark</h2>
<p>The Great ...</p>
<h2>Hammerhead Shark</h2>
<p>The head ...</p>
...
```

Figure 3.2: Headings in an exemplary HTML Web document

HTML was meant to provide a document representation so that documents can be easily created by authors and are visually appealing to readers browsing the linked Web documents. Figure 3.3 shows a Web document rendered in a Web browser. The markup is hidden from the user; for example, a table defined by the HTML table modelled, which is created by hierarchical nested tags, is rendered as a 2 dimensional visual structure. Tables are also used for the general page layout. The different degrees of headings are usually causing a different visual appearance indicated by different font sizes.

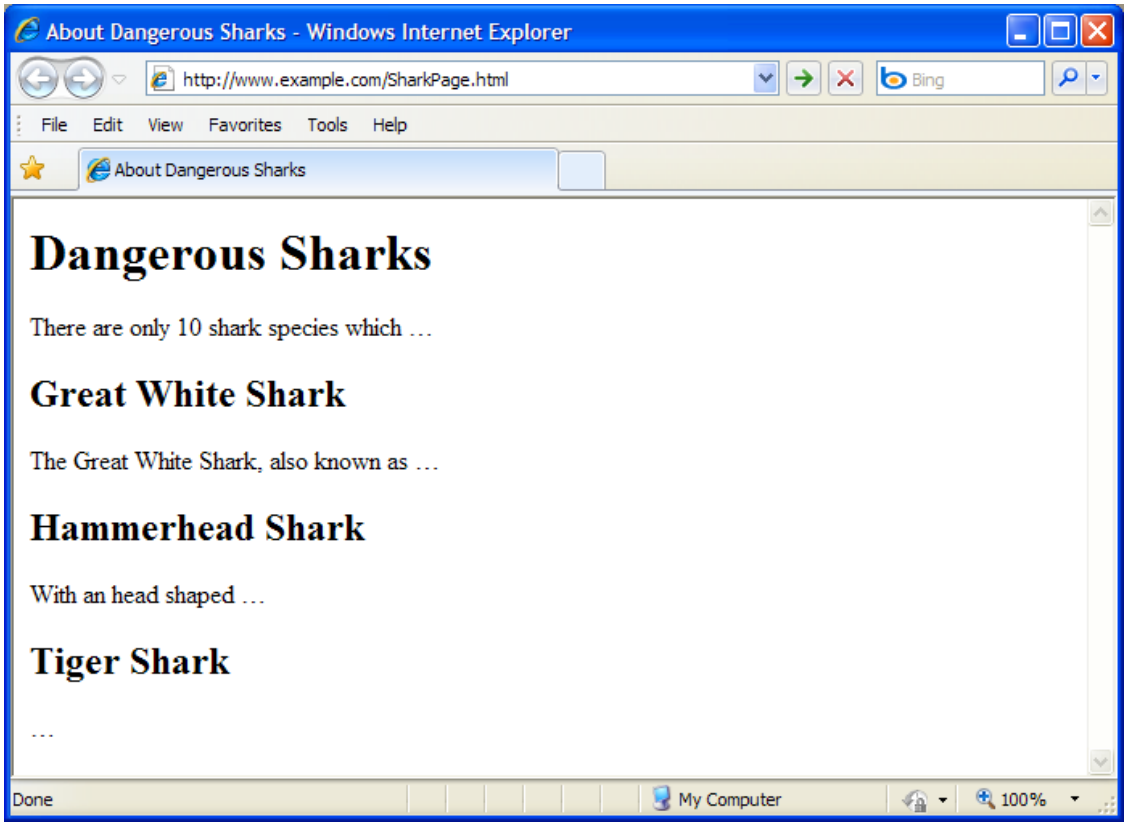


Figure 3.3: Web document rendered in a Web browser

By means of the various HTML tags it is possible to create documents which are to be read by humans. While creating Web documents, authors place pieces of text at the same structural level within Web documents for various purposes, such as providing navigation menus and so on. A human who reads a displayed Web document can often recognize several textual passages which stand on the same "level" and which could be captured in an ontology since they depict useful ontological entities. In particular we refer to items which are plausible sibling items, available from Web documents and worthy to be captured in an ontology. For example, while reading the Web document displayed in figure 3.3, an ontology engineer creating an ontology on a domain where `sharks` are relevant (for example diving) he could get inspired to model a concept `shark` with the sub-concepts `Great White Shark`, `Hammerhead Shark` and `Tiger Shark`. Alternatively he might decide to create `Great White Shark`, `Hammerhead Shark` and `Tiger Shark` as instances of `shark`. We aim at facilitating the acquisition of terms standing in a sibling relation and which in an ontology would be labels of sibling entities such as sibling concepts, sibling relations or sibling instances. We do so by exploiting structural regularities within Web documents.

For our goal of finding sibling text terms by grouping texts according to structural regularities we rely on the tree structure of Web documents. This basic requirement is fulfilled by documents adhering to the XML standard [Bray et al., 1998]. Hence, within this thesis, a *Web document d* is a semi-structured document according to the XML standard. But the notion of XML documents is rather general, while in speaking of Web documents a more specific notion of Web documents is typically referred to, (X)HTML documents. Originally HTML did not enforce a strict tree structure. The eXtensible HyperText Markup Language (XHTML) [Pemberton et al., 2000] is an XML dialect, wherein the former HTML standard has been adopted to meet the XML requirements. Traditional legacy HTML documents can mostly be automatically converted to XHTML documents, as it is also done by popular Web browsers. Hence they are subsumed by our notion of Web documents where a tree structure can be assumed.

For our following descriptions we rely on the terminology of the Document Object Model (DOM) [Vidur Apparao, 1998], a widespread platform- and language-neutral interface that allows representing Web documents as a hierarchy of nested nodes. Two kinds of tree nodes are important for our purpose, (1) *element nodes* and (2) *text nodes*. Element nodes have a *tag name* which, for example, depicts the HTML tags such as `title`, `p`, `h1` and so on. We do not rely on a fixed set of tag names but on an open set of tag names as allowed by the XML standard.

The other kind of nodes we rely on are text nodes. Text nodes depict the textual part of Web documents, the sequence of characters. We refer to the textual content of text nodes as *text span*. Text nodes are leafs in the tree and cannot contain child nodes. Text nodes have to be normalized, meaning that adjacent text nodes are merged into one text node, which is the default behaviour until a DOM is changed.

Figure 3.4 depicts an excerpt of the source code of the example Web document shown in figure 3.3. Figure 3.5 shows the tree structure of the example Web document of figure 3.4.

```
<html>
      <head>
            <title>About Dangerous Sharks</title>
      </head>
      <body>
            <h1>Dangerous Sharks</h1>
            <p>There are some shark species …
               <h2>Great White Shark</h2>
               <p>The Great White Shark, also known as …</p>
               <h2>Hammerhead Shark</h2>
               <p>The head shaped …</p>
               <h2>Tiger Shark</h2>
            …
            </p>
      </body>
</html>
```

Figure 3.4: Source code of a Web document

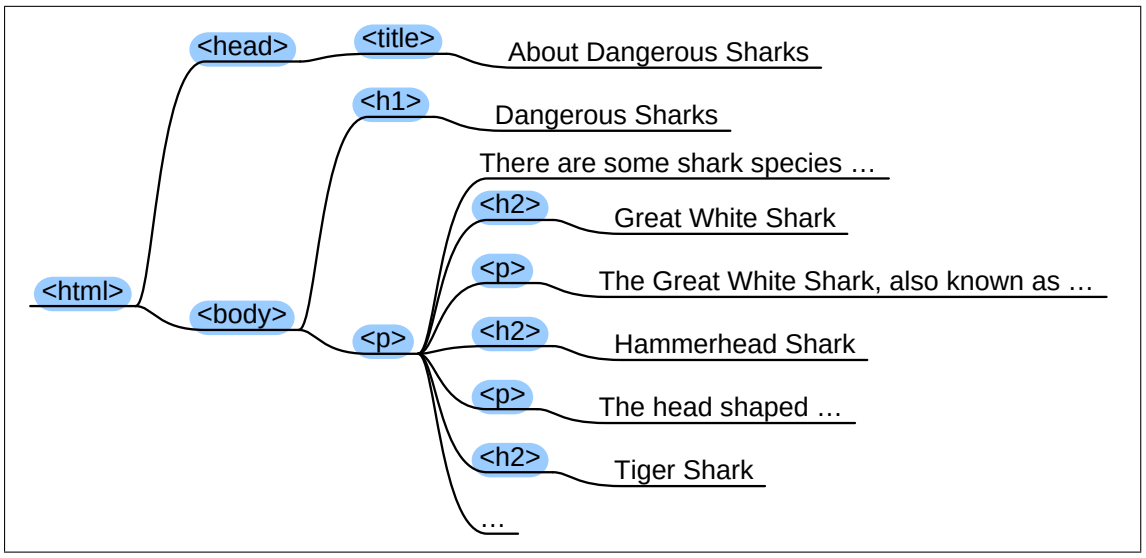


Figure 3.5: Tree structure of a Web document

Next we define the structural regularity which we will use to group text sequences together, the so called *tagpath*.

Let d be a Web document, let c be a text node, and e the text span of a text node c in d. A "tagpath" p in d is a sequence of tags (element tag names) m of the path of tree nodes leading from the root node in d to the text node c in d. Hence, p has the form $p =< m_1, m_2, \ldots, m_v >$, where m_i are tag names of element nodes.

We use the notation (p, e) to indicate that e is the text span to which p leads. It should be noted that by using only the tag names, a tagpath emerges from individual paths of nodes. This is an important difference from the vast majority of approaches that also use the structure of Web documents which we will present as related work in section 3.4.

Moreover, a Web document d constitutes a collection of pairs of the form (p, e), where p is a tagpath and e is the text span at its end. This is exemplary shown in figure 3.6. The tagpaths and text spans of the Web document excerpt in figure 3.4 are shown in figure 3.6. In line 4, the tagpath `<html><body><p><h2>` leads to the text span `Great White Shark`. In the next line, the tagpath `<html><body><p>` leads to a long text span depicting a paragraph of text.

```
<html><head><title>About Dangerous Sharks
<html><body><h1>Dangerous Sharks
<html><body><p>There are some shark species …
<html><body><p><h2>Great White Shark
<html><body><p><p>The Great White Shark, also known as …
<html><body><p><h2>Hammerhead Shark
<html><body><p><p>The head shaped …
<html><body><p><h2>Tiger Shark
…
```

Figure 3.6: A Web document with its tagpaths and text spans

Summary: In this section we described that we understand a Web document as an XML document where a tree structure can be observed. Thus for every sequence of textual content which we denote as text span a tagpath can be created. In the next section we describe how the tagpaths are used to extract sibling text spans.

3.2 Group-By-Path Algorithm

Next we present the Group-By-Path approach. The aim of the Group-By-Path approach is to group text spans which adhere to a common structural regularity namely the tagpath introduced in the previous section. Those text spans are, therefore, sibling text spans according to a structural regularity, the tagpath.

The expectation is that by obtaining such text spans standing in a sibling relation we will extract also siblings according to the notion of siblings described in chapter 1 section 1.4, siblings which are plausible with respect to ontologies.

The Group-By-Path algorithm, depicted below as Algorithm 3.1, takes as input a Web document d. Essentially, d will be represented as a collection of tagpaths with associated text spans, which we denote as Y_d. For the document d, or equivalently for the set Y_d, Group-By-Path identifies and groups identical tagpaths: all distinct tagpaths in d constitute a set Z_d (line 3). For each tagpath $p \in Z_d$, Group-By-Path finds all occurrences of p. For each such occurrence, it adds into a set B_p the text span e, to which p leads (line 5). This set B_p is the *text span set* of the tagpath p. So, for the document d, Group-By-Path forms and returns the union of all text span sets $A_d = \cup_{p \in Z_d}\{B_p\}$ (lines 6 and 8).

Algorithm 3.1: The Group-By-Path Algorithm

1 **Input:** Web document d
2 **Output:** A multiset A_d, containing the text span set of each tagpath $p \in Z_d$
1: $A_d = \emptyset$
2: map d to the set Y_d of (p, e)-pairs, where p is a tagpath and e is its target text span
3: let Z_d be the set of tagpaths in d, that is $Z_d := \{p|(p, e) \in Y_d\}$
4: **for all** $p \in Z_d$ **do**
5: $B_p = \{e|(p, e) \in Y_d\}$
6: $A_d = A_d \cup \{B_p\}$
7: **end for**
8: **return** A_d

```
<html><head><title>About Dangerous Sharks
<html><body><h1>Dangerous Sharks
<html><body><p>There are some shark species …
<html><body><p><h2>Great White Shark
<html><body><p><p>The Great White Shark, also known as …
<html><body><p><h2>Hammerhead Shark
<html><body><p><p>The head shaped …
<html><body><p><h2>Tiger Shark
…
```

Figure 3.7: Grouping of text spans with the same preceding tagpath

This operation is performed for all tagpaths of one Web document. As a result a collection of text span sets is obtained. Consequently, it is possible that more than one group of tagpath siblings can be obtained from one Web document.

When we apply Group-By-Path on the tagpaths of the Web document in figure 3.6, text spans of the same colour are grouped together. The tagpath <html><body><p><h2> groups together the text spans Great White Shark, Hammerhead Shark and Tiger Shark. Subsequently we acquire the text span set {Great White Shark, Hammerhead Shark, Tiger Shark} for the tagpath <html><body><p><h2>. The text spans appearing after the occurrences of the tagpath <html><body><p><p> build another text span set of text spans The Great White Shark, also known as ..., The head shaped ....

As we can see in the examples, multiword terms are included. Or more exactly, by default, text spans, however long they might be, are not dismissed. Text spans are also not truncated; they are used in their entire length. Text spans are just grouped according to their tagpath. The Group-By-Path algorithm works solely on text spans which can be sequences of text of arbitrary length thus comprising single characters, words, multiwords, phrases, sentences, paragraphs and other granularities of plain text. The transition from text spans to terms can be done after Group-By-Path was applied. As we will see in later chapters, the long text spans which are rather paragraphs or phrases than terms will be filtered out by subsequent processing steps. This can be done in different manners. First, one can restrict text spans to be only text spans which are a subset of a given vocabulary of terms as applied in chapters 4, 5, 6 and 8. Second, while the Group-By-Path approach is conducted on large numbers of Web documents, one can settle on the hypothetical circumstance that text spans which are frequent correspond to terms and can be imagined as proper labels of ontological entities. This hypothetical circumstance that text spans which are frequent correspond to terms will be investigated in chapter 7. It is stressed here that the capability of processing multiword expressions in the same way as simple words is an important advantage of this approach. This holds true especially for languages like English which, unlike German, do not tend to create compounds.

Summary: According to the Group-By-Path algorithm, the tags are used regardless of the purpose that tags are meant to fulfil within a Web document. Tags are used to infer siblings if text spans have a certain tagpath in common. Only the structure imposed by the tags causes the grouping of text spans. The grouping according to tagpaths is nevertheless suspected to be indicatory of semantic sibling relations - as we will investigate in the following chapters.

3.3 Real World Example and Application Outlook

Next we want to observe a real world example, where the Group-By-Path approach was conducted on a real world web document. Figure 3.8 shows the Web document[1] rendered. Figure 3.9 shows the list of tagpaths and text spans retrieved from this page. Figure 3.10 shows all tagpaths with more than one sibling text span.

[1] http://www.seasky.org/reeflife/sea2i.html, Screenshot taken on June, 23th 2008

From the 9 text span sets with a cardinality of at least 2, there are two sets which apparently can be beneficial for an ontology engineer capturing knowledge in a particular domain the set {(carcharodon carcharias), (galeocerdo cuvier), (sphyrna lewini), (prionace glauca), (carcharhinus amblyrhynchos)} and the set {great white shark, tiger shark, scalloped hammerhead shark, blue shark, gray reef shark}. An ontology engineer using these results can create appropriate subconcepts or instances according to his objectives. But it is only hypothetical to use the results obtained from Web document since such a result alone, without further processing, is problematic due to several reasons which we will discuss next. Afterwards we will propose a solution to those problems.

Figure 3.8: A exemplary real world Web document (http://www.seasky.org/reeflife/sea2i.html)

<HTML><HEAD><TITLE> sharks and rays page 1 - coral reef life on sea and sky
<HTML><BODY><TABLE><TR><TD><DIV><TABLE><TR><TD><DIV><TABLE><TR><TD>
<HTML><BODY><TABLE><TR><TD><DIV><TABLE><TR><TD><DIV><TABLE><TR><TD><FORM><TABLE><TR><TD><LABEL> enter your search terms
<HTML><BODY><TABLE><TR><TD><DIV><TABLE><TR><TD><DIV><TABLE><TR><TD><FORM><TABLE><TR><TD><LABEL> submit search form
<HTML><BODY><TABLE><TR><TD><DIV><TABLE><TR><TD><DIV><TABLE><TR><TD><FORM><TABLE><TR><TD>
<HTML><BODY><TABLE><TR><TD><DIV><TABLE><TR><TD><DIV><TABLE><TR><TD><FORM><TABLE><TR><TD><TABLE><TR><TD><LABEL><FONT> web
<HTML><BODY><TABLE><TR><TD><DIV><TABLE><TR><TD><DIV><TABLE><TR><TD><FORM><TABLE><TR><TD><TABLE><TR><TD><LABEL><FONT> www.seasky.org
<HTML><BODY><TABLE><TR><TD><DIV><TABLE><TR><TD><TABLE><TR><TD>
<HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><TABLE><TR><TD>
<HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><P><FONT><STRONG> sharks & rays
<HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><DIV><TABLE><TR><TD><P><FONT><STRONG> page 1
<HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><P><FONT> there is perhaps no other animal on earth that evokes more fear in the mind of man than the shark. they are viewed as vicious man-eaters and are slaughtered the world over in an attempt to make the seas safe. but of the hundreds of different species of sharks in the ocean, only a small handful pose any threat to man. humans do not appear to be on the menu for sharks. it is thought that most shark attacks are a case of mistaken identity. a diver in a wet suit looks a lot like a sea lion, a favorite food for some of the larger sharks. the fact is that more people are killed by lightning each year then by sharks. public fear and ignorance of these magnificent animals has led to many species being hunted and killed in large numbers. they have almost disappeared in some parts of the world. sharks one of evolution's most perfect creations. many of them have not changed in millions of years. they have evolved into the perfect hunting machine. sharks are part of a family of fishes known as cartilage fishes. they have no bones in their bodies, only cartilage, like the soft flexible tissue in the end of your nose. rays are also a member of this group. in fact, rays are actually nothing more than flattened out sharks. they are the "birds" of the sea. they can be seen flying gracefully through the water as effortlessly as a bird flies through the air. some rays are capable if inflicting painful stings with their tails. others, such as the giant manta, can grow to enormous proportions but are completely harmless to man. below is a listing of some of the more common sharks and rays found in the world's coral reefs.
<HTML><BODY><TABLE><TR><TD><DIV><DIV><TABLE><TR><TD><FONT><STRONG> great white shark
<HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><DIV><TABLE><TR><TD><FONT><FONT> (carcharodon carcharias)
<HTML><BODY><TABLE><TR><TD><DIV><DIV><TABLE><TR><TD><P><FONT> few animals inspire more fear in the mind of man than the great white shark. it is an aggressive and ruthless hunter, and in many ways, is the ideal predator. this shark has been known to grow to over 25 feet in length. the great white is very common in the pacific ocean. they will eat almost anything, but prefer to dine on sea lions and other marine mammals. unfortunately, human ignorance and fear has contributed to the decline of this magnificent animal in the wild.
<HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><DIV><TABLE><TR><TD><FONT><FONT><STRONG> tiger shark
<HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><DIV><TABLE><TR><TD><FONT><FONT><FONT> (galeocerdo cuvier)
<HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><DIV><TABLE><TR><TD><P><FONT> the tiger shark is another ferocious predator. it is second only to the great white in its size and reputation as a killer. tigers can grow to a length of over 16 feet. this shark has a big head, blunt snout, and gets its name from the stripe marks on its body. this dangerous shark will eat almost anything, and has been known to attack humans.
<HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><DIV><TABLE><TR><TD><FONT><FONT><STRONG> scalloped hammerhead shark
<HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><DIV><TABLE><TR><TD><FONT><FONT><FONT> (sphyrna lewini)
<HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><DIV><TABLE><TR><TD><P><FONT> the scalloped hammerhead is just one of several species of sharks that are characterized by a large hammer-shaped head. the shark's eyes are located on either end of this wing-like structure. this shark grows to a length of 14 feet, and feeds mainly on small fish and invertebrates. it is an aggressive species and has been known to attack humans.
<HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><DIV><TABLE><TR><TD><FONT><STRONG> blue shark
<HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><DIV><TABLE><TR><TD><FONT><FONT> (prionace glauca)
<HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><DIV><TABLE><TR><TD><P><FONT> the blue shark is a slender species that gets its name from the bright blue color of its tail and fins. it can be identified by its long, thin body and long, conical snout. the blue is one of the most common sharks in the sea, and is found in many parts of the world. they are often seen swimming lazily at the surface, but have also been seen at depths of over 1600 feet.
<HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><DIV><TABLE><TR><TD><FONT><FONT><STRONG> gray reef shark
<HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><DIV><TABLE><TR><TD><FONT><FONT> (carcharhinus amblyrhynchos)
<HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><DIV><TABLE><TR><TD><P><FONT> the gray reef shark is one of the major predators on the coral reef. its highly streamlined body allows it a great deal of speed and maneuverability in the water. the gray reef is a very aggressive species, and is commonly seen in the classic "feeding frenzy" film footage. this shark can be identified by the black markings on its pectoral and tail fins.
<HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><P><FONT><B> previous page |
<HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><P><FONT><B><A> next page
<HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><P><FONT><B> | go to page 1
<HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><P><FONT><B><A> 2

Figure 3.9: Tagpaths and text spans from Web document

```
Set Number: 1
PagePath: <HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><P><FONT><B><A>

Set member: "next page"
Set member: "2"
Set member: "3"

Set Number: 2
PagePath: <HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><P><FONT><B>

Set member: "previous page |"
Set member: "| go to page 1"

Set Number: 3
PagePath:
<HTML><BODY><TABLE><TR><TD><DIV><TABLE><TR><TD><TABLE><TR><TD><DIV><TABLE><TR><TD><FORM><TABLE><TR><TD><LAB
EL>

Set member: "enter your search terms"
Set member: "submit search form"

Set Number: 4
PagePath: <HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><DIV><TABLE><TR><TD><FONT><FONT><FONT>

Set member: "(carcharodon carcharias)"
Set member: "(galeocerdo cuvier)"
Set member: "(sphyrna lewini)"
Set member: "(prionace glauca)"
Set member: "(carcharhinus amblyrhynchos)"

Set Number: 5
PagePath:
<HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><P><TABLE><TR><TD><FORM><TABLE><TR><TD><TABLE><TR><TD><LABEL
><FONT>

Set member: "web"
Set member: "www.seasky.org"

Set Number: 6
PagePath: <HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><P><TABLE><TR><TD><FORM><TABLE><TR><TD><LABEL>

Set member: "enter your search terms"
Set member: "submit search form"

Set Number: 7
PagePath: <HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><DIV><TABLE><TR><TD><FONT><FONT><STRONG>

Set member: "great white shark"
Set member: "tiger shark"
Set member: "scalloped hammerhead shark"
Set member: "blue shark"
Set member: "gray reef shark"

Set Number: 8
PagePath: <HTML><BODY><TABLE><TR><TD><TABLE><TR><TD><DIV><DIV><TABLE><TR><TD><P><FONT>

Set member: "few animals inspire more fear in the mind of man than the great white shark. it is an
aggressive and ruthless hunter, and in many ways, is the ideal predator. this shark has been known to grow
to over 25 feet in length. the great white is very common in the pacific ocean. they will eat almost
anything, but prefer to dine on sea lions and other marine mammals. unfortunately, human ignorance and fear
has contributed to the decline of this magnificent animal in the wild."
Set member: "the tiger shark is another ferocious predator. it is second only to the great white in its
size and reputation as a killer. tigers can grow to a length of over 16 feet. this shark has a big head,
blunt snout, and gets its name from the stripe marks on its body. this dangerous shark will eat almost
anything, and has been known to attack humans."
Set member: "the scalloped hammerhead is just one of several species of sharks that are characterized by a
large hammer-shaped head. the shark's eyes are located on either end of this wing-like structure. this
shark grows to a length of 14 feet, and feeds mainly on small fish and invertebrates. it is an aggressive
species and has been known to attack humans."
Set member: "the blue shark is a slender species that gets its name from the bright blue color of its tail
and fins. it can be identified by its long, thin body and long, conical snout. the blue is one of the most
common sharks in the sea, and is found in many parts of the world. they are often seen swimming lazily at
the surface, but have also been seen at depths of over 1600 feet."
Set member: "the gray reef shark is one of the major predators on the coral reef. its highly streamlined
body allows it a great deal of speed and maneuverability in the water. the gray reef is a very aggressive
species, and is commonly seen in the classic "feeding frenzy" film footage. this shark can be identified by
the black markings on its pectoral and tail fins."

Set Number: 9
PagePath:
<HTML><BODY><TABLE><TR><TD><DIV><TABLE><TR><TD><TABLE><TR><TD><DIV><TABLE><TR><TD><FORM><TABLE><TR><TD><TAB
LE><TR><TD><LABEL><FONT>

Set member: "web"
Set member: "www.seasky.org"
```

Figure 3.10: Text spans from Web document grouped according to tagpaths

First, identifying even a single Web document with a good sibling text span group is not straightforward. In contrast, acquiring many Web documents for a topic can be done by performing focused Web crawls.

Second, not all found text span sets are plausible, as being semantic siblings which are likely to be labels of ontological sibling entities; only 2 out of 9 sets are good candidates. Observing the raw results and identifying plausible sibling groups would not be feasible for large quantity of such results.

Third, in this particular case, the hammerhead shark is referred to in a more special case, as "`Scalloped Hammerhead Shark`" compared to our simplified example of figure 3.3 where a "`Hammerhead Shark`" was referred to. Authors use different "granularities" among their presentation of sibling items. Also an ontology engineer has to decide whether "`Hammerhead Shark`" or "`Scalloped Hammerhead Shark`" or both should become entities of the ontology. This depends on the domain to be represented in the ontology and the ontology objectives in general.

The forth and perhaps a more problematic reason is the errors where text spans are captured in on set due to a common tagpath but where the text spans are not plausible siblings. For example, if we imagine that the text spans which are now contained in set number 6 have the same tagpath as the text spans from set number 8, a set "`blue shark`", "`enter your search terms`", "`gray reef shark`", "`great white shark`", "`scalloped hammerhead shark`", "`submit search form`","`tiger shark`" would be created where the coherence of being semantically plausible siblings is not given for all set members. It can happen that tagpaths do not distinguish well among sibling text spans and a sibling text span group which as a whole is not plausible is created. The found raw results are not reliable; they lack a "statistical" validation.

The solution to overcome the problems described above is to involve the Group-By-Path algorithm on large amounts of Web documents, obtain large amounts of sibling groups and to apply appropriate techniques for acquiring the underlying "patterns". In concrete this means that we will apply clustering and association detection to reduce the number of obtained sibling groups to a number which could be studied by the ontology engineer. The following chapters describe solutions where large amounts of Web documents are processed with Group-By-Path and subsequently the large amount of "raw" text span sets are processed to obtain results which are suited to be presented to an ontology engineer.

3.4 Related Work

In this section we describe work which is related to the Group-By-Path approach because these techniques also rely on structural characteristics of XML documents. These approaches are not required to be from the field of ontology learning but are applied for various purposes.

First we describe wrapper 3.4.1 which also exploit the structural regularities of Web documents for extracting information. Then we will describe approaches

related to the sibling aspect of XPaths. Afterwards we briefly address XML document similarity and XML document clustering. Then we will describe approaches which are related because they also use paths within HTML/XML documents but in a different way and for different purposes compared to Group-By-Path.

3.4.1 Wrapper

Wrapper inductions systems to be applied on semi-structured documents are the pendent to information extraction systems to be applied on plain text. Both apply pattern matching based on extraction rules. A recent survey on the major Web data extraction approaches also commonly referred to as wrappers is given by Kayed and Shaalan [Kayed and Shaalan, 2006]. A first application of wrapper learning appeared in [Doorenbos et al., 1997] where an agent for querying online stores for known product names and detection of regularities was described. Later work on wrappers generalized and formalized this idea [Kushmerick et al., 1997, Muslea et al., 2001]. Wrapper induction systems are related as there are also approaches using the tree structure [Muslea et al., 1999] of Web Documents or even tree paths [Cohen and Fan, 2000, Cohen et al., 2002]. There are also Wrapper systems which extract sibling "items" such as movies, books and their associated properties. In contrast, we do not extract the properties of "complex items" consisting of many together belonging attributes, but only sibling text items. Alvarez et al [Alvarez et al., 2008], in a study published later than our approach, describe an approach which also uses what we call tagpaths for extracting records from Web documents. Their approach requires, in contrast to former alternatives, only single documents adhering to a template. But wrappers in general are more focused regarding the homogeneity of Web documents they are to be applied upon compared to our Group-By-Path approach. According to Sarawagi [Sarawagi, 2002], HTML wrappers can be distinguished into approaches which operate on (1) record-level, where usually boundaries are to be detected, (2) page-level, which extract all data from one page and (3) site-level, which can cope with information scattered across several pages. But they are focused on more specific regularities than Group-By-Path. Group-By-Path can be applied on Web documents from arbitrary sources/authors. Group-By-Path is not limited to Web documents that have been created based on the same template. This is related to the observation that wrapper induction system tends to be focused on obtaining results which are not as heavily processed afterwards as we do with the Group-By-Path method. Group-By-Path is usually applied on large amounts of Web documents from which large amounts of raw candidate information is acquired. This raw information needs to be processed appropriately as demonstrated in the following chapters to make the potentially large amounts of information presentable to humans.

Wrappers usually rely on annotated training data; in contrast, our approach does not rely on annotated training examples. This is a serious problem and a

disadvantage of most wrapper systems since the creation of annotated examples is laborious and secondly because changes on the structure of Web documents which practically occur, tend to draw the learned rules useless. The type of structural regularity which is used by the Group-By-Path operation, equal tagpaths, is restricted to one Web Document. Therefore Group-By-Path can cope with changing Web document structures, as it can cope with arbitrary inhomogeneous Web document structures in general.

3.4.2 XPath - Siblings

After the creation of XML related standards and the emergence of larger amounts of content stored in XML documents, dedicated XML processing approaches also have been investigated. The XPath [Clark and DeRose, 1999] standard is a language for selecting nodes from an XML document. The XPath language is based on a tree representation of the XML document and provides the ability to navigate in the tree and to select nodes by a variety of criteria.

In the XPath standard also a notion of siblings exists. In contrast to our Group-By-Path approach, siblings can be arbitrary tree nodes, regardless of if they are leaf nodes or non-leaf nodes and regardless of if they are elements, texts or attributes. In contrast, Group-By-Path is only centred on textual leaf nodes as siblings. The notion of siblings in XPath is based on a common parent node but not a path of tags. Therefore, siblings are restricted to those that occur close to each other. In contrast, by using the tagpaths, text spans are captured as siblings by Group-By-Path regardless of how far away they occur within a document tree. In XPath there are the so called "following-sibling axis" which refers to all the siblings that follow a node. The preceding-sibling axis analogously refers to all preceding siblings of a node. The differentiation between following and preceding siblings is also different from that of our Group-By-Path style of siblings, where the order of siblings is not captured and preserved.

In summary, the Group-By-Path tagpaths siblings are different from XPath siblings:

- Group-By-Path concentrates on textual leafs nodes; it does not consider the siblinghood of non-leaf nodes.

- Group-By-Path does not concentrate on siblings of the same parent; siblings can be the children of different parents nodes.

The above described notion of siblings within the XPath language is also reflected in research of creating *indexes on XML repositories* [Cooper et al., 2001][Gou and Chirkova, 2007] [Krátký and Baca, 2006]. There the aim is to develop high-performance techniques to query large XML data repositories efficiently. Also special attention is paid to create sibling indexes [Cho, 2005] but in the XPath sibling notion described before which is different from Group-By-Path.

3.4.3 XML Document Similarity

For computing *similarity among XML documents* there are several approaches [Zhang et al., 2003] [Buttler, 2004]. The used similarity measures try to consider the characteristics of the structure within XML documents [Leung et al., 2005] [Nierman and Jagadish, 2002]. The difficulty lies in finding ways of performing the comparison in an efficient way [Long et al., 2005]. But those similarity measures are different from the way we use similarity. Those methods compare the similarity of Web documents to each other. We are only interested in using equal tagpaths to find sibling text spans within one document. With exception of [Iyer and Simovici, 2007] neither the tagpaths nor their associated text spans are used to perform a processing where Web documents are compared with each other. Iyer and Simovici [Iyer and Simovici, 2007] present a system where documents are contrasted by comparing the corresponding multisets of tagpaths. They use only the occurrence frequencies of tagpaths but not the text which is associated with the tagpaths as it is the objective of Group-By-Path.

The above mentioned similarity metrics can be used to perform processing where documents are compared with another, for example, clustering. *Clustering of XML documents* is performed in [Costa et al., 2004, Dalamagas et al., 2004, Dalamagas et al., 2006, Vuong et al., 2006, Choi et al., 2007]. The results obtained by Group-By-Path can be clustered as well, but not for the purpose of clustering documents but for clustering sets of sibling text spans for obtaining sibling relations as we will show in the following chapters.

3.4.4 Further Path based Approaches

Mukherjee, et al [Mukherjee et al., 2003] describe an approach which relies on their observation that semantically related items in HTML documents exhibit spatial locality. They (re)discover parts of the implicit schema in HTML documents which have been created in a template driven manner. First they create so called "root-to-leaf paths" for all leaf nodes. They include attribute values for the creation of the root-to-leaf paths; therefore, their paths have a different constitution from our Group-By-Path paths. They only regard paths as equivalent when they are equal, as we do with Group-By-Path. In a series of "root-to-leaf paths" they perform sequence mining for discovering "partitions". A partition corresponds to a sibling group obtained by Group-By-Path. According to found partitions they alter the document tree. They apply heuristics to label the partitions. As a result they obtain a labelled tree for a Web document. They have applied their approach to some selected Web documents. The application of their approach is only reasonable on Web documents where one can expect a sufficient quality to yield useful results. The documents have to be rather manually selected. In contrast, Group-By-Path works on large amounts of automatically crawled Web documents without any restrictions. To obtain results where a domain is covered to a larger extent one has to apply their approach on many documents where subsequently

many result trees are obtained. Each single result reflects only the schema of a certain page but not a shared common understanding of sibling items. This is different from the intended application of Group-By-Path where the potentially large number of obtained structural siblings are processed to obtain a consolidated result. Applying sequence mining on each document can be a computationally expensive operation. In contrast, our approach is computationally cheaper since it obtains siblings from one Web document. Their approach is interesting regarding their heuristics of obtaining labels for partitions (sibling text span sets). This can be a useful enhancement for the XTREEM-SG approach described in chapter 4 where such alternative cluster labels could be obtained.

Liu, Keong and Lim [Liu et al., 2004] describe a system for extracting the structure of websites. They extract the underlying hyperlink structure that is used to organize the content pages in a given website. Their SEW algorithm examines groups of hyperlinks; he identifies the navigation links that point to pages in the next level in the website structure. The algorithm starts from the homepage of a website and discovers the links to the Web documents in the next level in a top-down manner. The algorithm solely focuses on `A` tags, the representation of links with HTML. They collect all links in the order in which they occur within the document. Then links are to be "clustered". They call a function "GROUPBYPATH" which divides the nodes into groups where nodes in each group have the same path to the root node in the DOM tree. Their notion of paths is different. Their paths are restricted to paths pointing to a hyperlink. Their approach is focused to paths separating links stored in `href` attributes whereas the Group-By-Path operation works on all text nodes, regardless of the surrounding tag. The path is created only up to the `A` element node, not including the `A` element node as the last path component. The value to which their path points is the `href` attribute of the `A` tag which actually represents the link. In contrast, our Group-By-Path tagpath points to a span of text.

Chung, Gertz and Sundarsan [Chung et al., 2002] describe an approach where HTML documents are transformed into XML documents whereupon a "majority schema" is derived that describes common structures among the documents in the form of a DTD. They treat the Web documents as a collection of tagpaths. From those tagpaths they create an occurrence frequency statistic which is used to derive their "majority schema" which is supposed to be valid for most of the processed documents.

3.5 Summary

In this chapter we describe the so called Group-By-Path approach which relies on regularities within XML Web documents for obtaining sequences of text which are siblings. The structural regularity which we defined is the so called tagpath, the series of tags which point to the textual content of Web documents. The tagpaths are used to group the textual content into sets of text spans. The text spans of

those sets are foremost siblings due to structure. The hypothesis is that the siblings are not limited to structural coherence, but that they tend to be semantically founded. In the subsequent chapters we describe solutions where the Group-By-Path approach is applied on large volumes of Web documents and where we perform experiments to investigate whether the obtained structurally motivated siblings can be used to obtain semantically plausible siblings.

The Group-By-Path operation relies on the structuring within Web documents for extracting semantics. It is, therefore, *language and domain independent*. An important characteristic is the capability of Group-By-Path to process multiword terms in the same way as single word terms. Group-By-Path does not rely on any language specific notion of words/terms; it groups sequences of text of arbitrary length.

4 Learning Sibling Groups - XTREEM-SG

In chapter 3 we describe the Group-By-Path approach which enables obtaining groups of sibling text spans from Web documents. This operation will lift groups of sibling text spans which are far from perfect if they are viewed in isolation. The idea is to overcome this "weakness" by applying Group-By-Path on medium and large Web document collections to obtain large amounts of sibling text span groups which afterwards are processed to reveal shared patterns while particular variations are dimmed. To do so, we apply clustering, a form of unsupervised learning. Clustering is applied to extract a reasonable number of patterns from large quantities of input data.

XTREEM-SG stands for Sibling Groups discovery with the XTREEM (Xhtml TREE Mining) approach. The XTREEM-SG process aims at structuring a given vocabulary into semantically motivated sibling groups. The processing is done by extracting sibling groups according to the Group-By-Path operation from medium to large semi-structured Web document collections.

In the next section we describe the XTREEM-SG procedure which consists of 6 steps. Afterwards, in section 4.2, we explain the evaluation which is performed within the experiments described in section 4.3. In the evaluation we aim to investigate if the text span sets obtained by Group-By-Path indeed capture semantically plausible sibling relations to a large extent.

4.1 XTREEM-SG Procedure

The overall XTREEM-SG procedure is depicted in figure 4.1. In the next sections we describe the 6 single steps.

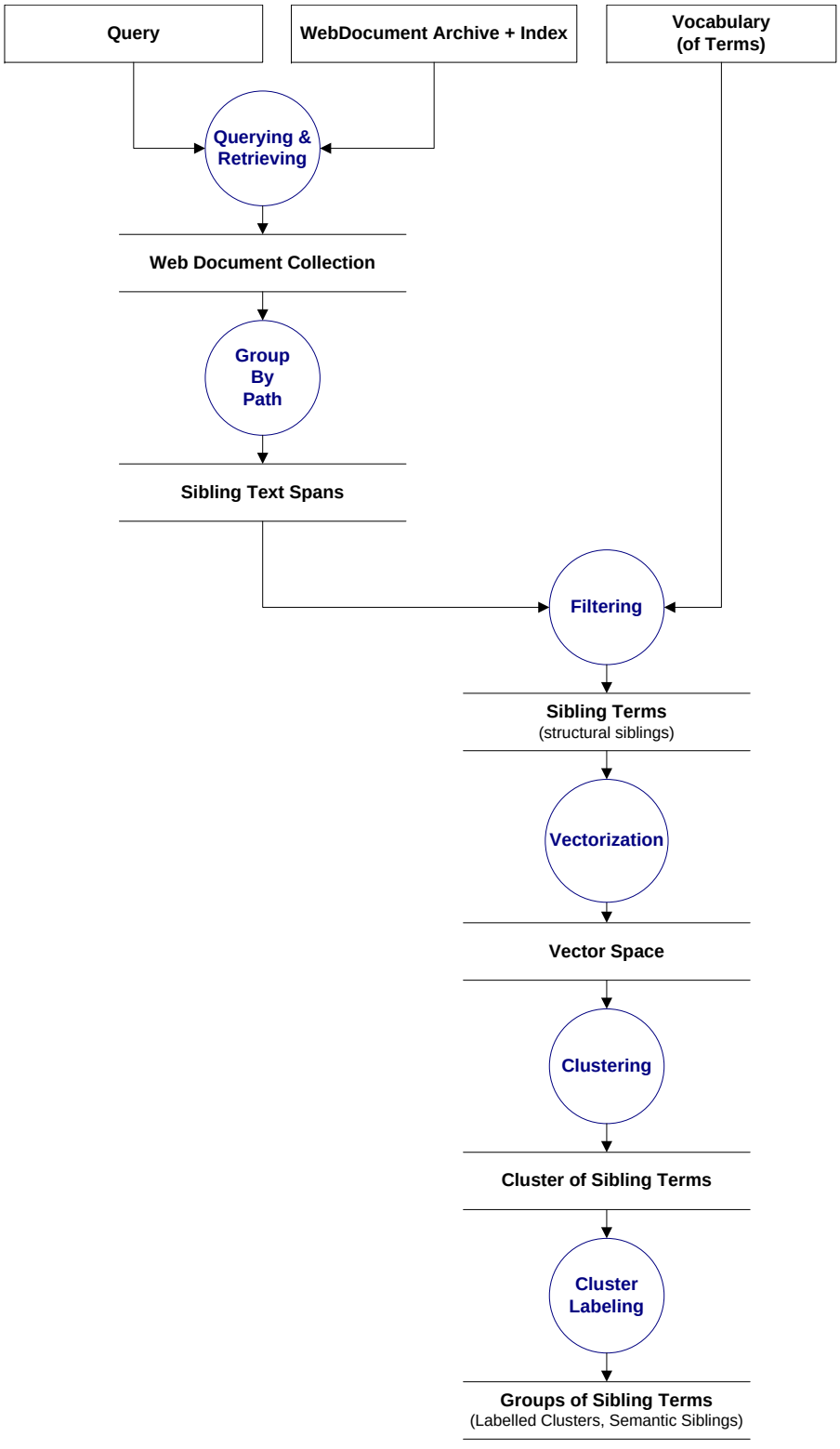


Figure 4.1: Dataflow diagram of the XTREEM-SG procedure

4.1.1 Step 1 - Querying & Retrieving:

The aim of the first step in the XTREMM-SG procedure is to establish a collection of Web documents which are used to find the sibling relations. Except in the rare case that a Web document collection is already locally available, Web documents have to be obtained from the World Wide Web. *Web crawling* is the process in which pages are gathered from the Web. Since the availability of Web documents is a necessity, we will first briefly describe Web crawling to obtain the input for the processing step 1.

Web Crawling: The procedures and systems presented in this thesis rely on Web documents which are to be processed. Web documents are a necessary input. Here the crawl should efficiently retrieve as many useful Web pages as possible. Early descriptions of Web crawlers are given by [Burner, 1997, Brin and Page, 1998, Cho et al., 1998, Najork and Heydon, 2001, Najork and Heydon, 2002]. Overview descriptions on Web crawling can be found in [Chakrabarti, 2002, Markov and Larose, 2007, Manning et al., 2008].

The Web search engines have crawlers running which in certain time intervals perform crawls. Some crawlers try to crawl as much as they can whereas other crawlers limit their scope according to certain criteria. Since it is not always feasible to retrieve all pages available, but only those pages fulfilling certain criteria, there is the notion of so-called *focused crawlers* [Chakrabarti et al., 1999]. Focused crawlers are supposed to prioritize their crawling on Web documents which adhere to, a priori, given topics. While fetching pages, the coherence of the fetched pages to the given topics is determined and the crawler will base his decision of which page to take next accordingly. In order to obtain a Web document collection to be used for ontology learning, ontology focused crawlers [Ehrig and Maedche, 2003] can play an important role. When ontology enhancement is to be performed, the already available ontology can also be used to obtain a Web document collection.

The Web documents used for the experiments of this thesis have been crawled with the Nutch [Cafarella and Cutting, 2004, Rohit Khare, 2004, Cutting, 2005] crawler. An alternative is to use the Web Service as for example provided by Amazon search service[1] where for a fee of 150$ 10 million results could be retrieved or of 80legs [2] where one million pages crawled account $2. This is a comparatively low fee compared to the efforts involved in running an own Web crawler.

The XTREEM-SG process takes as input a query Q that reflects the domain of interest, that is the application domain for which sibling relations should be identified. This query which is used to create the Web document collection is expected to comprise the domain of interest. It should be generic enough to

[1] http://aws.amazon.com/alexawebsearch/, available from 2006 to 2008
[2] http://www.80legs.com

ensure a reasonable coverage over the domain, for example, "touris*" for everything associated with tourists and tourism.

XTREEM-SG issues this query towards an *Archive+Index*. The results of Web crawls can be made available in an Archive+Index. The Web documents which have been stored in an archive for fast local access are indexed so that, for a given query, the documents matching that query can be delivered. The result of the query execution is a Web document collection $W = \{d_1, \ldots, d_s\}$.

The accessed archive should be of adequate size and diversity to ensure manifold occurrences of the desired concepts. The Web itself does satisfy this requirement for almost every domain; so an extensive Web crawl can establish a collection of adequate coverage. Recall is more important than precision at this stage.

4.1.2 Step 2 - Group-By-Path:

Each document $d \in W$ is processed by the Group-By-Path algorithm described in chapter 3. This algorithm takes as input Web documents in XHTML format. The transformation of HTML documents to XHTML is done as well. An XHTML document can be observed as a tree structure. Group-By-Path extracts each path leading from the root tag to a leaf tag and the corresponding "text span" pointed to by this path. Next, text spans appearing at the end of identical paths are grouped together to form text span sets. Hence, Group-By-Path maps each document d to a set of paths and associated text span sets. In subsequent steps, the text span sets of all documents will constitute the data space while a selection of text spans from the text span sets will form the feature space: terms that appear together in many text span sets are candidates as siblings.

Recapitulating our earlier example of figure 3.7, the text spans `Great White Shark` and `Hammerhead Shark` appear at the end of identical paths. Thus, the path `<html><body><h2>` is associated with the text span set `{Great White Shark, Hammerhead Shark}`. As one can see from the example, the text spans are multiword expressions or even larger phrases which may extend up to paragraphs. In the next step we describe how text spans of such varying lengths are dealt with.

Actually, only a small fraction of the Web document content is used. Here precision is more important than recall.

By applying Group-By-Path on all documents in the Web document collection W, we obtain all candidates for sibling groups discovery. They constitute the multiset of text span sets $\mathcal{A}_\mathcal{W} = \cup_{d \in W} A_d$.

4.1.3 Step 3 - Filtering:

Before proceeding with data mining for sibling groups discovery, we perform two filtering steps upon the collection of text span sets ($\mathcal{A}_\mathcal{W}$) obtained with Group-By-Path. First, if there is an a priori known domain-specific vocabulary V, we use it to eliminate text spans from $\mathcal{A}_\mathcal{W}$, which do not appear in V. This allows us to remove text spans that may be irrelevant to the domain. However, this filtering operation

should be treated with caution: If V does not cover the whole domain, using it may lead to the elimination of terms which are of potential relevance. In this study, we assume that the vocabulary exists and it contains the terms for which we want to find sibling relations, as part of the ontology engineering process.

Text spans constituted by long sequences of words are unlikely to be valid term expressions. These long text spans are usually infrequent. Since we are interested in text spans that are valid term expressions, long sequences of words which are unlikely to be valid term expressions are filtered out automatically by, for example, frequency based feature space building approaches [Mladenic, 1998, Liu and Yu, 2005]. In chapter 7 a procedure and experiments for obtaining vocabularies by means of Web document markup-up are presented. The results presented there give an impression of what an automatically obtained feature space can look like. In chapter **??** we also describe parameters which are suited to filter text spans beyond text span occurrence frequency.

After the filtering in accordance with a given vocabulary only text spans corresponding to terms remain; thus we can refer to those sets as term sets. Furthermore, we remove term sets that have only one member, since such sets do not contribute to the discovery of sibling relations. We use the retained text spans as (single-word or multiword) terms.

4.1.4 Step 4 - Vectorization:

In conventional document mining, analysis is performed upon *document vectors* comprising the terms occurring in each document *or* upon *term vectors* comprising the documents where each term occurs. For the first option, the terms constitute the feature space, upon which a document is described as a vector. For the second option, the feature space consists of the documents and a term is a vector. These two options are important for subsequent steps; so we explain them here with an example.

For XTREEM-SG, a document has been mapped into its tagpaths. So, the equivalents of the two options are (a) a vectorization of tagpaths in the feature space of terms and (b) a vectorization of terms over the feature space of tagpaths. In the example of figure 4.2, they correspond to (a) using the columns as feature space and the rows as vectors vs. (b) transposing the matrix and vectorising the columns over the rows. We use both vectorizations for the subsequent clustering task and use the terms *tagpath vectorization*, resp. *tagpath clustering*, and *term vectorization*, resp. *term clustering* for them.

4.1.5 Step 5 - Clustering

XTREEM-SG performs clustering for sibling groups discovery, offering the option of either term clustering [Grefenstette, 1994, Faure and N'edellec, 1998, Gamallo et al., 2005] or tagpath clustering.

	great white shark	hammerhead shark	tiger shark	batoidea	pinniped	orca	…
DocumentA<html><body><h2>	**1**	**1**	**1**	**0**	**0**	**0**	**…**
DocumentB<html><body><table><h1>	1	0	1	0	0	0	…
DocumentC<html><body><p>…	0	0	0	1	1	0	…
…	…	…	…	…	…	…	…

Figure 4.2: Exemplary fragment of a Group-By-Path vectorization

A cluster of term vectors consists of terms that co-occur in many tagpaths. Hence, the members of such a cluster constitute a sibling group. We obviously assume that there are no clusters with only one member. If the clustering algorithm allows for such clusters, then one-member clusters have to be ignored.

The characteristics of tagpath clustering are different from the characteristics of a term clustering. A cluster of tagpath vectors consists of tagpaths that have many terms in common. These terms are not members of the cluster, they are rather the dimensions/features, which *characterize* the cluster. These terms constitute the *label* of the cluster, which is built in the last step of XTREEM-SG. Clusters with only one member are possible and can be labelled whereas the label directly reflects the corresponding term set. But since the number of tagpaths to be clustered is large compared to the number of clusters, clusters with only one member are less problematic since they only rarely occur.

Tagpath clustering seems less straightforward with respect to sibling group discovery, because it demands cluster labelling. Term clustering, although more intuitive, has a serious disadvantage: a term, being a vector, can belong to only one cluster[3] and so can appear in at most one sibling group. Since a term may have multiple meanings or participate in multiple sibling relations in different contexts,

[3]We assume exclusive clustering.

forcing the term to belong to exactly one cluster is undesirable. Tagpath clustering solves this problem by allowing the same term to appear in multiple labels.

Details of the implementation: For both tagpath clustering and term clustering we have chosen the K-Means algorithm [Lloyd, 1957, MacQueen, 1967, Hartigan and Wong, 1979]. K-Means is a widespread, easy to implement algorithm. As pointed out by [Steinbach et al., 2000], it has many limitations, but it is still appropriate as a first choice for experiments. We have also experimented with the more robust Bi-Secting-K-Means variant described by [Steinbach et al., 2000], but have obtained results of lower quality as described in chapter 5 and in [Brunzel, 2007]. Determining the number of clusters K is a major challenge. In our experiments, we have varied the value of K and studied the influence of large values upon the algorithm's performance. As distance function cosine distance was used. We used the implementation of K-Means from WEKA [Witten and Frank, 2005].

4.1.6 Step 6 - Cluster Labelling

A major difference between term clustering and tagpath clustering is in how the clusters which are generated are labelled. For term clustering, there is a straightforward labelling strategy: a cluster is labelled by all terms which have been assigned to a cluster. For tagpath clustering more complex strategies are required.

The label of a tagpath cluster C is the group of terms that appear frequently in the members of the cluster, subject to a threshold τ. In particular, for a term/feature $f \in F$ where F is the feature space, we define the *in-cluster-support* [Schaal et al., 2005] of f in C, $ics(f, C)$, as the number of members of C that contain f normalized by the cardinality of C. Then, the *label* of C is the set of features, whose in-cluster-support in C exceeds a predefined lower boundary value τ, that is, $L(C) = \{f \in F | ics(f, C) \geq \tau\}$. Each label that contains at least two terms constitutes a sibling group. One-term labels are ignored.

Remark: Alternatively to the above described application of labelling according to a cut-of threshold, it can be sufficient that the terms are ordered by their in-cluster-support into a ranked list. The application of a labelling cut-of threshold is required for the automatic comparison of crisp cluster labels with a reference where also crisp term sets are provided as for the evaluation of clusters performed within this thesis. In practical scenarios, where a human user observes the clustering results, the user can stop reading down the ranked list when the results no longer meet his requirements - similar to the usage of results from (Web) search engines. This can, for example, be seen in the screenshot of figure 5.4, shown in chapter 5. The left side contains a table where a row depicts a cluster. While a cluster is selected, the details of the cluster are shown on the right side. A bar chart diagram depicts the features ordered according to their within cluster support.

Choosing a frequency threshold τ is not straightforward and, therefore, has to be determined by applying several thresholds and choosing those with the best results. If this is not applicable, it can (as the number of clusters to be generated) be chosen based an educated guesses or by a sound heuristically inspired number.

4.2 Evaluation Methodology

In this section we describe the evaluation method and the evaluation criteria, the evaluation reference, inputs, what we vary on the processing procedure, and used and varied parameters.

There are two major types of evaluation which we consider for the evaluation of results obtained by the proposed procedures, human expert evaluation and gold standard evaluation. The human expert evaluations are expensive. Gold standard evaluations which can be done automatically are the usual approach. We will compare the automatically obtained results against semantic sibling relations from a gold standard reference. The measured quality is not easily comparable (over different references), but it can help to show tendencies.

Our evaluation objective is to study how well the method performs on structuring a given vocabulary into sibling groups. We evaluate results that deliver both the vocabulary (the terms) and the sibling relations among them against gold standards. Our goal is to find those sibling relations. The criteria too judge how well the structuring into sibling groups was accomplished is the *F-Measure on average sibling overlap* (FMASO) which was proposed in [Cimiano and Staab, 2005] for the evaluation of sibling relations. The FMASO is described in the next section, section 4.2.1. We will compare the results obtained by our method against the references by means of this measure.

It is to be emphasized here that the objective of the evaluation is not the reconstruction of the complete hierarchy, comprising the naming or detection of the super-concept for each sibling group. In fact, XTREEM-SG is meant to discover sibling groups for which the super-concept may or may not be a priori known.

4.2.1 Evaluation Criteria: Sibling Group Overlap

Each of the gold standard ontologies contains a set of reference sibling groups. XTREEM-SG, as well as the procedures for Bag-Of-Words and MarkUp, delivers their own sets of candidate sibling groups. Intuitively, one would compare each candidate sibling group against each reference sibling group, select the best match and then count the number of common members between the candidate group and the reference group. Candidate sibling groups without match would be regarded as false positives. Reference sibling groups without match would also contribute to the error. However, selecting a *single* "best match" is neither straightforward nor is it always appropriate.

Example 2 *In an ontology on tourism (or geography), all towns of the world are siblings under the concept "town of the world". Within this enormous reference sibling group, there are still many subsets of siblings that are conceptually closer to each other. Among them, one may consider (a) all towns in the same country, (b) all towns along the same river, (c) all towns having an airport, (d) all towns close to the* same *airport, (e) all towns with more than 1 million inhabitants, (f) all capital cities, etc. It is quite probable (from some supportive documents) that two towns are siblings according to one or more of the specific relations above. It is much less probable that 3, 5 or 10 towns are siblings as "towns of the world". At the same time, finding that London and Tokyo are siblings for the relations (c), (e) and (f) is perhaps of more interest than finding out that Amsterdam, Cerbere, Hammerfest, Heraklion and Kyoto are all towns of the world and thus siblings.*

This extreme example highlights a situation that is not uncommon in hand-crafted ontologies, namely, that not all concepts are refined in the same level of detail. Hence, it may happen that some concepts are very abstract and have a lot of children that are not really very related to each other (for example the towns of the world in Example 2), while other concepts are refined in more detail.

For our evaluation we therefore need a measure of the *contribution* of each candidate sibling group to each reference sibling group. We use the "F-Measure on Average Sibling Overlap" (FMASO) proposed by [Cimiano and Staab, 2005].

Definition 4.1 (FMASO) *Let A and B are two sets of sibling groups. Typically, one of them, say A, will be the set of reference sibling groups, while the other, B, will contain the candidate sibling groups. For a reference sibling group $x \in A$ and a candidate sibling group $y \in B$, we compute the "relative overlap" between x and y as the number of common terms in the two groups divided by the number of distinct terms in the groups: $\frac{|x \cap y|}{|x \cup y|}$. This set overlap is also known as* Jaccard coefficient.

For each reference sibling group $x \in A$ we select the candidate sibling group $x' \in B$ that has the maximum relative overlap with x. This is the "sibling overlap" for x towards B: $SO(x, B) = \max_{y \in B} \frac{|x \cap y|}{|x \cup y|}$. Then, we compute the average of these values over the sibling groups in A as the "average sibling overlap" of A towards B:

$$ASO(A, B) = \frac{1}{|A|} \sum_{x \in A} \max_{y \in B} \frac{|x \cap y|}{|x \cup y|} \tag{4.1}$$

The average sibling overlap of B towards A is computed similarly as $ASO(B, A)$. Then, the "F-Measure on the average sibling overlap" FMASO combines the values of both functions as:

$$FMASO = \frac{2 \cdot ASO(A, B) \cdot ASO(B, A)}{ASO(A, B) + ASO(B, A)}$$

The FMASO measure partially deals with the problem highlighted in Example 2 by considering partial matches also between reference sibling groups and discovered

sibling groups. Hence, the FMASO values for the mining methods will be more than zero, even if the ontology contains large groups of loosely related siblings, none of which can be found in the document collection as a whole.

The problem is not completely alleviated, though. If the reference ontology contains large sibling groups that cannot be reconstructed, then they still influence the values of the average sibling overlap.

A further unresolved issue in our evaluation concerns the treatment of terms that participate in multiple sibling groups. First, a term may have more than one meaning (in our Example 2 above, there is one town Paris in France and one in Texas). Second, there may be sibling groups of different semantics; in Example 2, the terms/towns London and Tokyo are siblings under concepts (c), (e) and (f). One of our reference ontologies (GSO1) does not support multiple inheritance; so terms may co-occur in only one group. This means that some of the false positives are not really false; rather, the ontologies are too restrictive with respect to reality. We point to this issue, but we cannot provide a remedy for it.

4.2.2 Evaluation Reference

The evaluation is performed towards two gold standard ontologies (GSO) from the tourism domain. Both ontologies have been created by experienced ontology engineers. As GSO1 we refer to the "Tourism GSO"[4], described in [Cimiano, 2006, pages 79 and 80]. This ontology contains 293 concepts grouped into 45 sibling sets. As GSO2 we refer to a second ontology from the tourism domain. This ontology is described in [Cimiano, 2006, pages 80 and 81] as "pruned version of the $O_{Tourism}$ ontology". This "Getess annotation ontology"[5] contains 693 concepts grouped into 90 sibling sets.

4.2.3 Inputs

There are three Inputs to the XTREEM-SG procedure and these are described below.

Archive+Index: We have performed a topic focused Web crawl on "tourism" related documents. With an n-gram based language recognizer[6] non-English documents have been filtered out. The overall size of the "tourism" document collection is about 9.5 million Web documents. The Web documents have been converted to XHTML. The documents are indexed, so that for a given query a Web document collection can be retrieved.

Queries: For our experiments we consider three document collections which result from querying the Archive+Index. The document collection gathers all those

[4]http://www.aifb.uni-karlsruhe.de/WBS/pci/TourismGoldStandard.isa
[5]http://www.aifb.uni-karlsruhe.de/WBS/pci/getess_tourism_annotation.daml
[6]http://lucene.apache.org/nutch/apidocs/org/apache/nutch/analysis/lang/LanguageIdentifier.html

documents adhering to $Query1$ - "touris*", $Query2$ - "accommodation" and by the whole topic focused Web document collection reflected by $Query3$ - "*". Those variations are the subject of experiment 3.

Vocabulary: The GSOs described above have a lexical layer. Each concept is labelled with a term. These terms constitute the vocabulary whereupon sibling relations are discovered.

4.2.4 Variations on Procedure and Parameters

In the following we describe those processing variants, processing alternatives and parameters which we vary during our experiments.

Document Representation Method: For the evaluation of the Group-By-Path subprocedure we will contrast our Group-By-Path (GBP) method with the traditional Bag-Of-Words vector space model and against the exclusive usage of MarkUp (MU). The Bag-Of-Words is the widely established method of processing of textual data, while MarkUp is used to contrast an approach using text spans also but without tagpaths. The variation of these influences is the subject of experiment 1 and experiment 2.

Clustering Direction: As described in section 4.1.4 to 5.2, there are two clustering directions: tagpath clustering and term clustering. We will apply and contrast both types of clustering directions. Experiment 1 to experiment 8 are aimed at performing tagpath clustering. Experiment 9 to experiment 11 are aimed at performing term clustering. Experiment 9 contrasts tagpath clustering to term clustering.

Number of Clusters: Each dataset (vectorization) is processed by a K-Means clustering with different numbers of clusters to be generated, ranging from rather small to rather big numbers of clusters. For K in tagpath clustering we used the values 50, 100, 150, 200, 250, 500, 750 and 1000. These numbers encircle the range of numbers of clusters which are appropriate to be shown to a human ontology engineer. This variation is made on all experiments involving tagpath clustering. For term clustering, the clustering is performed with values of K ranging from 10 to 350 in steps of 10. For this type of clustering K should be smaller than the number of terms; otherwise no grouping would be enforced.

Cluster Labelling Threshold: The generated tagpath clusters are post-processed by applying the support threshold cluster labelling strategy described in section 5.2. The support threshold is varied from 0.1 to 0.9 in steps of 0.1. The variation of this influence is the subject of experiment 3.

Minimum Feature Support Threshold: In our experiments we found that some of the terms of the vocabulary are never or very rarely found on relatively big Web document collections. For example, one reference contains the errors "Kindergarden" instead of the correct English "Kindergarten". To eliminate the influence of errors in the reference, we also vary the minimum feature support. The support is given by the frequency of the features (terms) in the overall text of the Web document collection. We used minimum support thresholds from 0 (all features are used, nothing is pruned) to 100000 (0, 1, 10, 100, 1000, 10000, 100000). When the support is varied, only those features of the vectorization and of the reference fulfilling these criteria are incorporated into the evaluation. The variation of the minimum feature support threshold is the object of experiment 6.

Number of Clustered Instances - Sampling: Processing large datasets in a high dimensional vector space, as it is the case for our datasets with clustering, can be computationally expensive. Since we are not directly interested in having each instance assigned to a cluster but on the cluster labels depicting sibling groups, applying *sampling* [Cochran, 1977] is a alternative worth considering. By means of sampling we will limit the amount of data to be clustered. Reducing the amount of data to be clustered proportionally reduces the required time of the clustering. We, therefore, used only a fraction of the entire dataset. In the experiments 7 and 10 we will investigate the stability of the resulting FMASO while comparing the entire dataset and the samples of the dataset.

There are various sampling strategies for various purposes; for example, undersampling, oversampling or random sampling. Selecting the instances randomly is a common method to obtain a sample which is a representative subset of the entire dataset [Manku et al., 1999]. We apply simple random sampling where each instance has the same probability to be chosen.

Small numbers of instances to be processed can also be the result of smaller Web crawls. The Web crawls we used have been obtained in rather long lasting processes for a broad domain which might not always be possible. Being capable of obtaining meaningful sibling groups from small Web crawls is an advantage in cases where only "smaller" Web crawls are feasible. Smaller Web crawls can practically occur because the domain is of rather limited size regarding the number of obtainable Web documents or where the time to obtain the Web document collection is rather limited.

It is not known in advance if the size of our used Web crawls is sufficient. In experiments 7 and 10 we will, therefore, process reduced datasets obtained by sampling. There are two ways to obtain samples of Group-By-Path data. One is to use a sample of the available Web documents; the other is to obtain a sample from the overall available raw text span sets. The first variant is more direct regarding conclusions of the process Web document size. But because of limited resources of processing time we choose the second variant where samples from the once acquired entire dataset are used instead of running the entire XTREEM-SG

procedure multiple times. Our findings will thus only allow for limited conclusions regarding the stability of results regarding varying Web document collections.

For tagpath clustering we generated samples of 10000, 25000 and 50000 instances. For each sample size we obtained 10 random samples. For term clustering we created a sample of 500, 1000, 5000, 10000, 50000 and 100000 instances.

Processing Algorithm - Frequent Itemset Mining in Comparison to Clustering: The filtering step of the XTREEM-SG procedure delivers a collection of termsets for the Web document collection. As already described, the termsets are vectorized and by means of clustering sibling terms groups are to be discovered. As an alternative, instead of finding patterns with clustering, patterns can as well be found by frequent itemset mining. The termsets which are vectorized for clustering can alternatively be treated as itemsets. Upon those itemsets, frequent itemset discovery using the a priori algorithm proposed in [Agrawal et al., 1993] can be performed. Here, the "items" are the terms and an "itemset" is a subset of a termset. The frequency and support refer to the number of occurrences of a term in the data. Frequently co-occurring terms constitute a sibling group. By doing so also sibling groups are obtained. They can be evaluated by means of the FMASO in the same way as groups obtained by clustering. For frequent itemset mining the algorithm from the XELOPES[7] library was used. Upon the a priori algorithm for finding frequent itemsets a support threshold can be set. In our experiments we varied the support by evaluating the top-n frequent itemsets with the highest support. For n we chose values of 10, 20, 50, 100 and 150, beginning with n=200 in steps of 100 up to n=5000.

4.3 Experiments

In the following we will show the results obtained from the experiments. Table 4.1 shows the number of documents which adhere to a certain query. This corresponds to the size of the Web document collection which is processed by the following processing steps.

Table 4.1: Number of Web documents returned by the Web Archiv+Index for the queries used in the evaluation experiments

Document Collection	Query Phrase	Number of Documents
1	"touris*"	1,468,279
2	"accommodation"	1,612,108
3	"*"	9,437,703

[7] http://www.prudsys.com/Software/Algorithmen/Xelopes/

4.3.1 Experiment 1: Sibling Relations from Group-By-Path in contrast to alternative Methods

In our first experiment we want to investigate how many sibling relations are captured by the Group-By-Path (GBP) method in contrast to alternative methods. As alternative methods can be regarded as the traditional Bag-Of-Words vector space model and the exclusive use of MarkUp (MU). To do so, the Group-By-Path step of the XTEEM-SG procedure was changed to Bag-Of-Words and MarkUp. We evaluated the collections of sibling sets for the following constellations of query ($Query1$, $Query2$, and $Query3$); document representation method (Bag-Of-Words, Group-By-Path and MarkUp) against the reference sibling sets (GSO1 and GSO2) of two gold standard ontologies. Since the two ontologies have different numbers of terms, each constellation of Web document collection results in a different number of vectors after the vectorization.

In this experiment we want to examine the raw, unprocessed term groups found by Group-By-Path. Each obtained sibling group is directly treated as a result sibling group. This corresponds to clustering, where each instance becomes a separate cluster. Actually this cannot be called clustering at all. One cannot expect a user to inspect such large numbers of groups; this represents a kind of baseline.

Table 4.2: Results of FMASO for different constellations of references, queries and document representation methods. The resulting sibling groups are separated according to their cardinality. Empty sets (no match with given vocabulary, cardinality=0) or single element sets (single match with given vocabulary, cardinality=1) are not processed since at least cardinality 2 is necessary to infer a sibling relation among the set member elements.

Reference	Document Collection	Method	Number of Sibling Term Sets			FMASO
			Cardinality=0	Cardinality=1	Cardinality>1	
GSO1	1	BOW	18,012	29,104	1,421,163	0.206
		GBP	12,589,016	817,289	222,037	0.247
		MU	794,325	343,891	323,428	0.235
	2	GBP	12,712,295	1,034,741	293,225	0.252
	3	GBP	63,049,135	3,485,782	924,045	0.256
GSO2	1	BOW	19,399	18,494	1,430,386	0.160
		GBP	12,478,364	831,969	318,009	0.208
		MU	753,657	332,973	375,014	0.199
	2	GBP	12,677,515	988,944	373,802	0.196
	3	GBP	62,572,763	3,559,356	1,326,843	0.229

The last column of table 4.2 shows the measured FMASO. From these results it can be seen that for GSO1 the FMASO is higher for all constellations where the Group-By-Path method was involved (0.247, 0.252, and 0.256) compared to the alternative methods (0.235, 0.206). Though it was never claimed that the traditional Bag-Of-Words method is strong on capturing sibling relations it resulted

in the weakest results on capturing sibling relations. For GSO2 the result of *Query*1 and MarkUp is slightly better than the result of *Query*2 and Group-By-Path, though for the same Query, Group-By-Path performs again best. Bag-Of-Words again performs worst.

The circumstance that Group-By-Path less sets with a cardinality of at least two are obtained is due to the fact that tagpaths are the most restrictive criteria for obtaining related terms compared to the two alternatives. This restriction of requiring terms to be placed at the same structural characteristic of Web documents yields the bias towards sibling relations.

Conclusion: The term groups generated by the document representation according to the Group-By-Path method reveal a stronger sibling relation characteristic than the traditional Bag-Of-Words vector space model. Though it was never claimed that Bag-Of-Words has significant sibling relation characteristics, it can be concluded that the Group-By-Path method does not capture sibling relations by chance; the path information of Web document structure can be used to obtain semantic sibling relations.

4.3.2 Experiment 2: Sibling Relations from Labelled Clusters

In addition to the raw sibling sets evaluated in experiment 1 a K-Means clustering was performed for *Query*1 and the document representation methods (Bag-Of-Words, MarkUp, and Group-By-Path). The cluster labelling threshold was set to τ=0.2. The threshold of τ=0.2 yielded the best results in experiment 3 where the cluster labelling threshold is varied for GSO1. Alternatively, we could have chosen τ=0.3 which yielded the best results for GSO2. But since the FMASO value for GSO2 is generally worse we choose τ=0.2 which yielded the best results.

Figure 4.3 shows that the Group-By-Path approach performs better when the sibling sets are clustered. There is a general trend towards better results when higher numbers of clusters are generated, as shown in experiment 4. The difference between MarkUp and Group-By-Path seems marginal. A possible explanation for this circumstance is that when termsets are created with MarkUp, those termsets have a big overlap with the termsets created by Group-By-Path since they originate from the same Web markup created text spans, caused by the rather small vocabularies used, that only allow for a fraction of the terms occurring in the Web document collections. Here, the insensitivity of the FMASO may also be responsible for the low measured difference whereas sibling relations not stated by the reference are regarded as false to the same extent as they are not truly sibling related nominations. This could only be solved by a human expert evaluation. In experiments judged by a human expert, one can see that the siblings yielded by Group-By-Path are more plausible compared to siblings yielded by MarkUp.

Conclusion: These results are compatible with the conclusions of experiment 1 and verify our hypothesis that Group-By-Path performs well on capturing sibling relations.

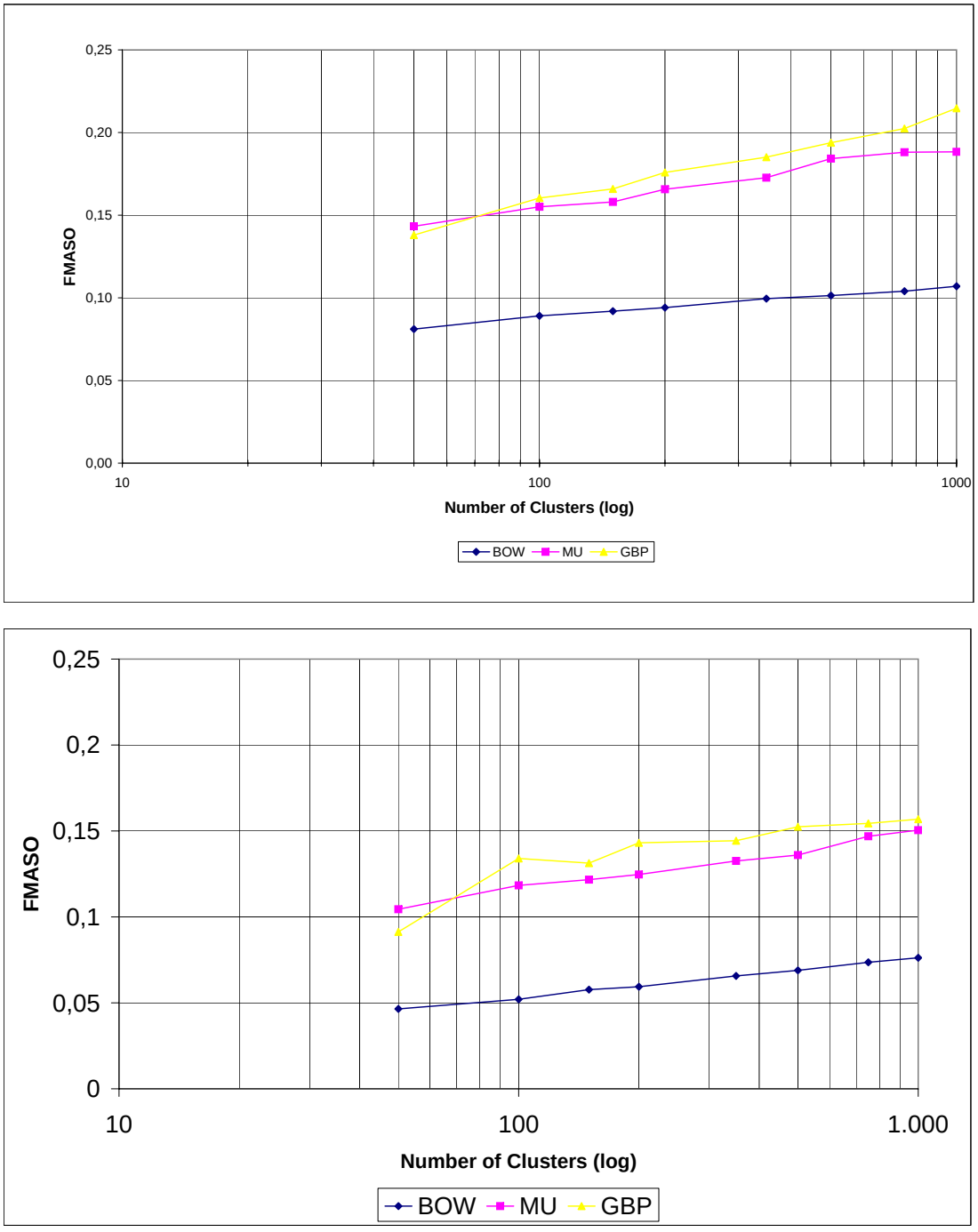


Figure 4.3: FMASO for different K and for different document representation methods (query1,τ=0.2) for (a) GSO1 and (b) GSO2

4.3.3 Experiment 3: Varying the Cluster Labelling Threshold

For *Query*1 in combination with Group-By-Path we varied the cluster labelling support threshold from τ=0.1 to τ=0.9 in steps of 0.1 resulting in the following chart of Figure 4.4. The best results have been obtained on the biggest used number of clusters (K=1000) in combination with a cluster labelling strategy using a support threshold of τ=0.2, resulting in an FMASO of 21.47 percent (figure 4.4, upper diagram). The results on GSO2 (figure 4.4, lower diagram) are (again) worse than the results for GSO1. The best FMASO of 15.88% for GSO2 is obtained on K=1000 and τ=0.3. The second reference ontology is more than twice as big as the first one; so structuring the vocabulary into sibling groups may be more difficult. We suspect that this has to do with the large size of the ontology. There are many terms, but not all sibling relations which can be found in the world are explicit in the reference.

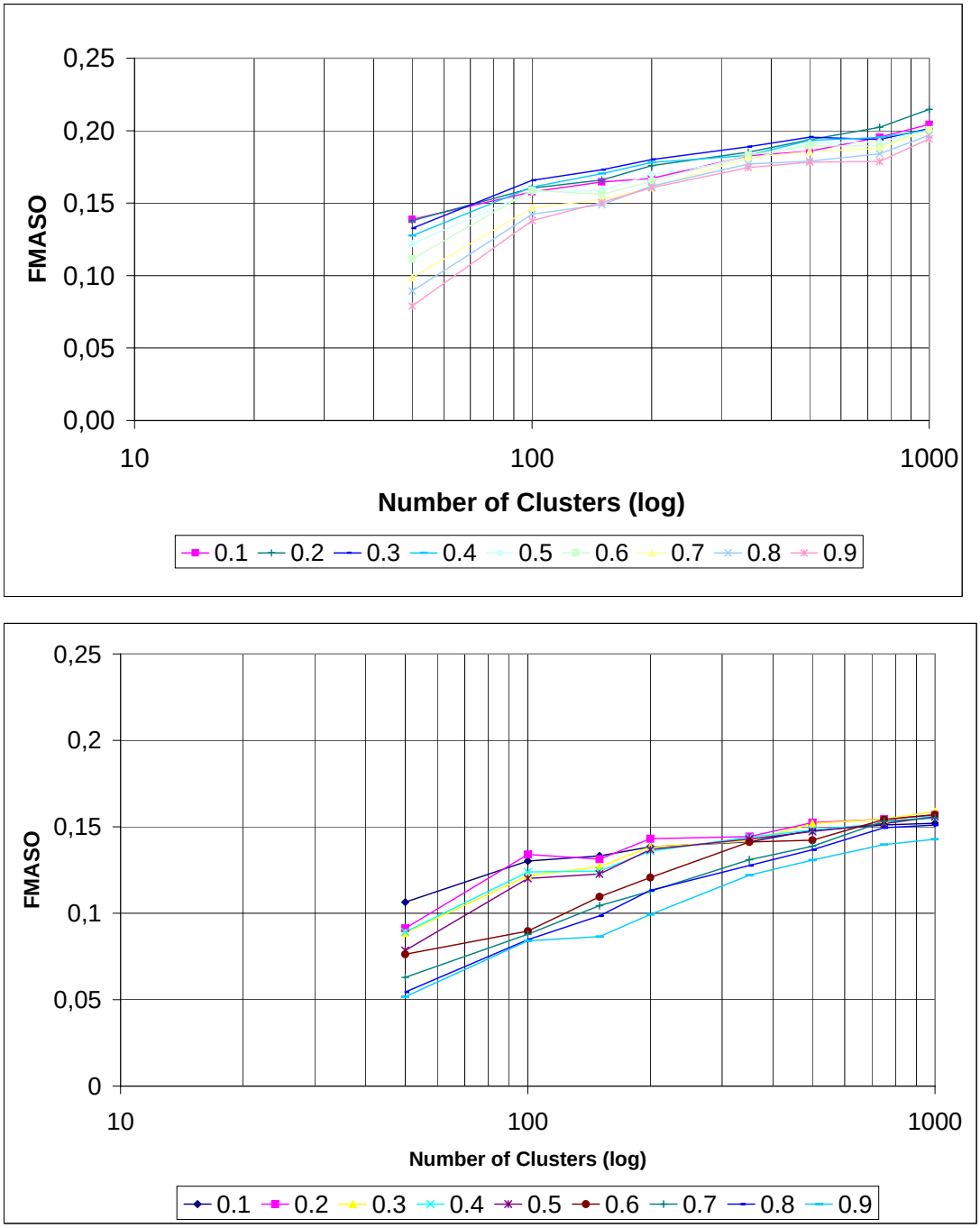


Figure 4.4: FMASO for different K and for different τ (query1) for (a) GSO1 and (b) GSO2

4.3.4 Experiment 4: Varying the Number of Clusters

As already shown in experiment 2 and in experiment 3, with an increasing number of clusters generated, the F-Measure on average sibling overlap increases too, but there is a saturation point (the number of clusters is logarithmically scaled). One interpretation of this is that the FMASO measure at it is does not punish weak overlaps enough; it seems to have a bias towards large numbers of compared sets. On the other hand, one can explain this situation by the circumstance that clustering is sometimes also regarded as a compression method. This behaviour can be observed for tagpath clustering too. With a decreased information amount, the observed quality goes down; as it is, for instance, for image compression. This loss of information can be regarded as acceptable, because only by doing so can the amount of data/information be reduced to an amount suitable for human consumption. The loss of information which we observed is not mandatory; one can also imagine cases where only by joining acquired sibling groups can good matches with a reference be established, but this behaviour was not observed by the combination of XTREEM-SG generated clusters evaluated towards the two reference ontologies.

For tagpath clustering the increasing number of clusters has the drawback that the overall number of terms a result is constituted of and which is compared to the reference, increases too. This overall number of terms is the information an ontology engineer is required to inspect. For automatic evaluation this is not a problem, but if a human would inspect the generated data, this is relevant since large amounts of information to be inspected decrease the potential benefits of ontology learning regarding lowering the per entries costs on ontology creation. We want to measure the number of terms a human ontology engineer would have to observe by using our results. First we count the number of terms/features appearing in the cluster labels for all the clusters of a clustering. This sum of terms/features in the cluster labels over all the clusters of a clustering we refer to as *Sum of Features in Cluster Labels* (SOFICL). This number is an indicator of how much information was produced by the process and represents a notion of precision in absolute numbers. The aim is to minimize this measure; optimal would be if SOFICL corresponds to the number of terms which form the reference sibling groups. The number of distinct features/terms used for cluster labelling we will refer to as *Number of distinct Features in Cluster Labels* (NODFICL). NODFICL captures the number of unique terms which are present in the results. This number states how many of the vocabulary terms are indeed structured and presented as the results by the clustering and cluster labelling. NODFICL represents a notion of recall in absolute numbers.

As figure 4.5 and figure 4.6 show, the values of SOFICL and NODFICL are correlated. With an increasing K (and decreasing τ) more terms are used for labelling in the sum (SOFICL) but also more distinct terms (NODFICL) are incorporated into the labelling. An increasing NODFICL means that a bigger share of the vocabulary is indeed incorporated into the results. The lower right corner of Figure 4.5 shows that for high numbers of clusters, 160-180 out of 293 of the

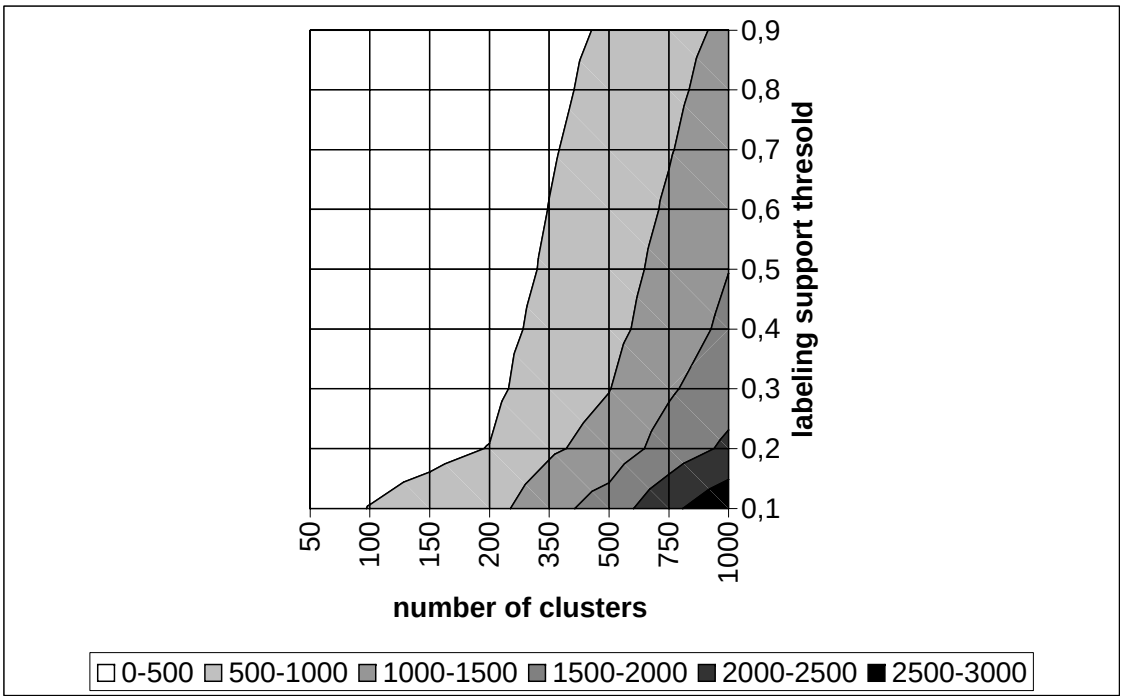


Figure 4.5: SOFICL for different K and τ (query1) for GSO1

features are used for cluster labelling. The circumstance that on lower numbers of cluster many terms/features are not used for labelling may be caused by the different support the features have within the vectorization. The low frequent features may never have the chance to be frequent enough for cluster labelling. But this is rather a problem for automatic evaluation. In practical settings; one can present a ranked list of features for a cluster to the user, who is free to choose less frequent features. For example, figure 5.4 (chapter 5) shows a screenshot of GUI where clusters can be sorted according to various cluster metrics. For the selected cluster a list of frequent features is shown. The user is free to interpret the within cluster support according to his objectives. Even a feature with low support can represent plausible siblings.

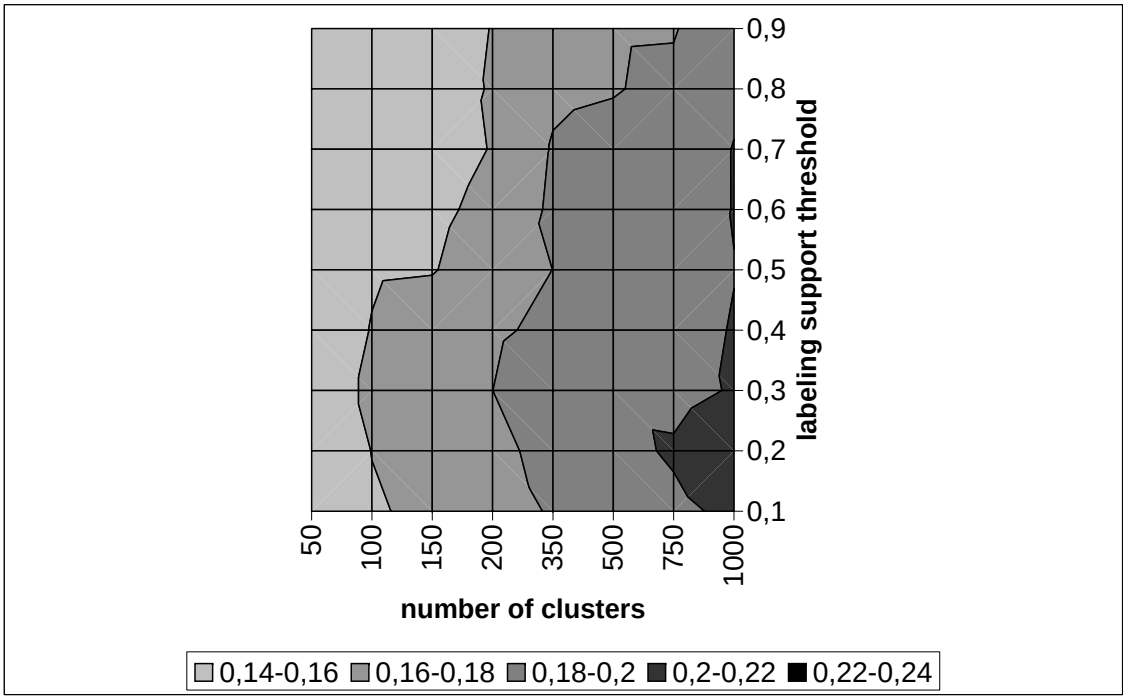


Figure 4.6: NODFICL for different K and τ (query1) for GSO1

4.3.5 Experiment 5: Varying the Topic Bias

Now we will investigate the influence of the processed Web document collection. Since the Web document collection is given by a query, we will apply the XTREEM-SG procedure for $Query1$ ("touris*"), $Query2$ ("accommodation") and $Query3$ ("*"; whole topic focused Web crawl). The queries hereby define the "exact topic", though in general those topics belong to the tourism domain.

The results depicted by Figure 4.7 show that there are no big differences in the results measured by the FMASO regarding the choice of a domain constituting query for GSO1. This is in so far a positive finding, that the domain expert should only roughly state which topic he is interested in. While doing so, minor variations do not lead to significantly worse or . The results are quite stable. For GSO2 $Query1$("touris*") and $Query3$ (big tourism focused Web crawl) yielded the best results. An explanation for this may be that $Query1$ and $Query3$, which are broader than $Query2$ ("accommodation"), encircle more sibling characteristics which have also been encoded in the GSO2.

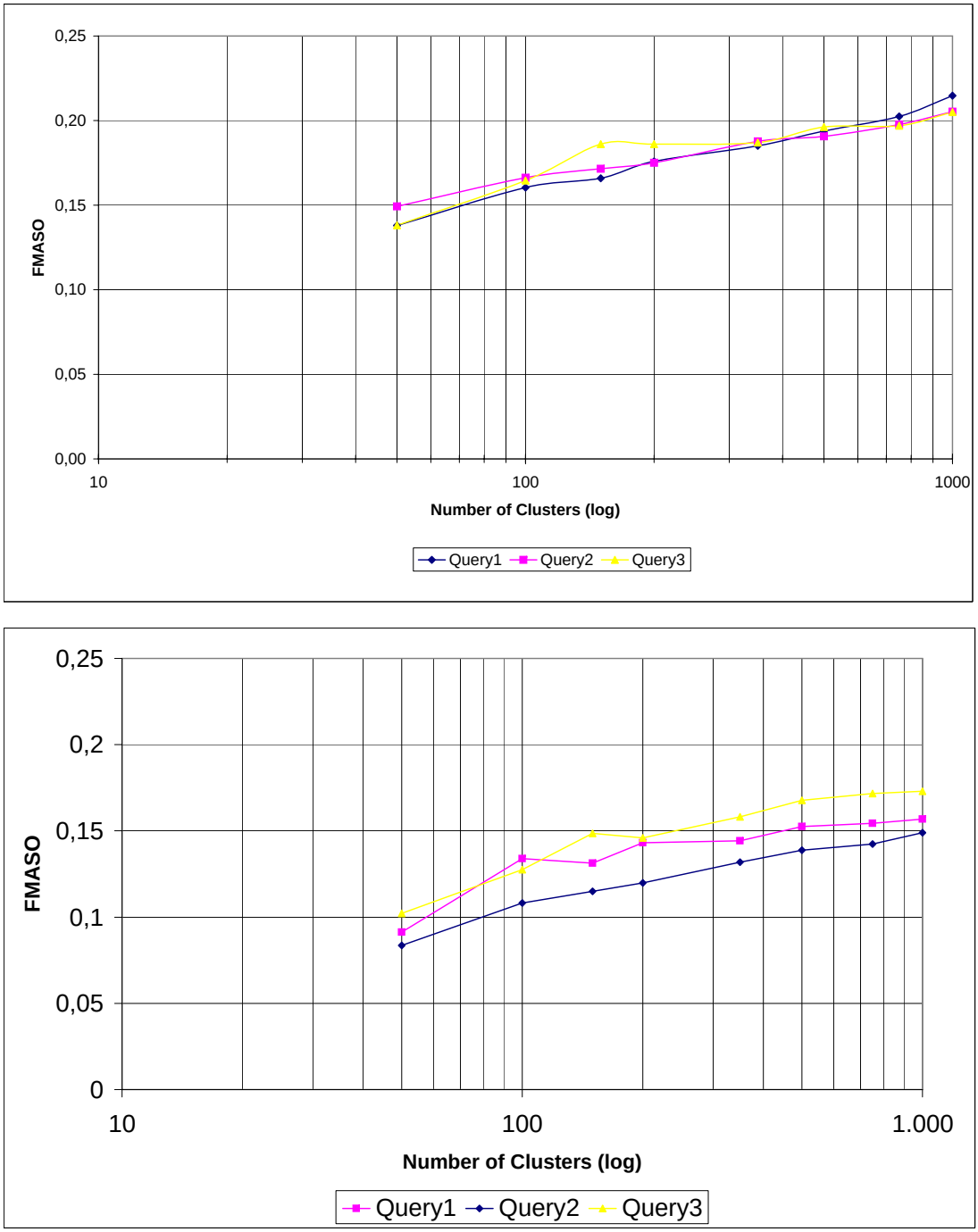


Figure 4.7: FMASO for different K and for different queries (τ=0.2) for (a) GSO1 and (b) GSO2

4.3.6 Experiment 6: Variations on the Minimum Support

For *Query*1 we set a threshold on the minimum support of terms in the Web document collection. This means that terms/features which are rather weakly supported are increasingly ignored, both for cluster labelling as well as in the reference sets. We varied the support threshold considering the values 1, 10, 100, 1000, 10000 and 100000 (absolute numbers of documents).

Figure 4.8 shows that while observing only frequent terms, better results on FMASO are shown. With a minimum support errors in the reference are removed. This is relevant to the extent that the relatively low FMASO values given by our approach and by other approaches on ontology learning are also caused by "not perfect gold standards"; the parts of the reference which are supported by large quantities of data are found reasonably well.

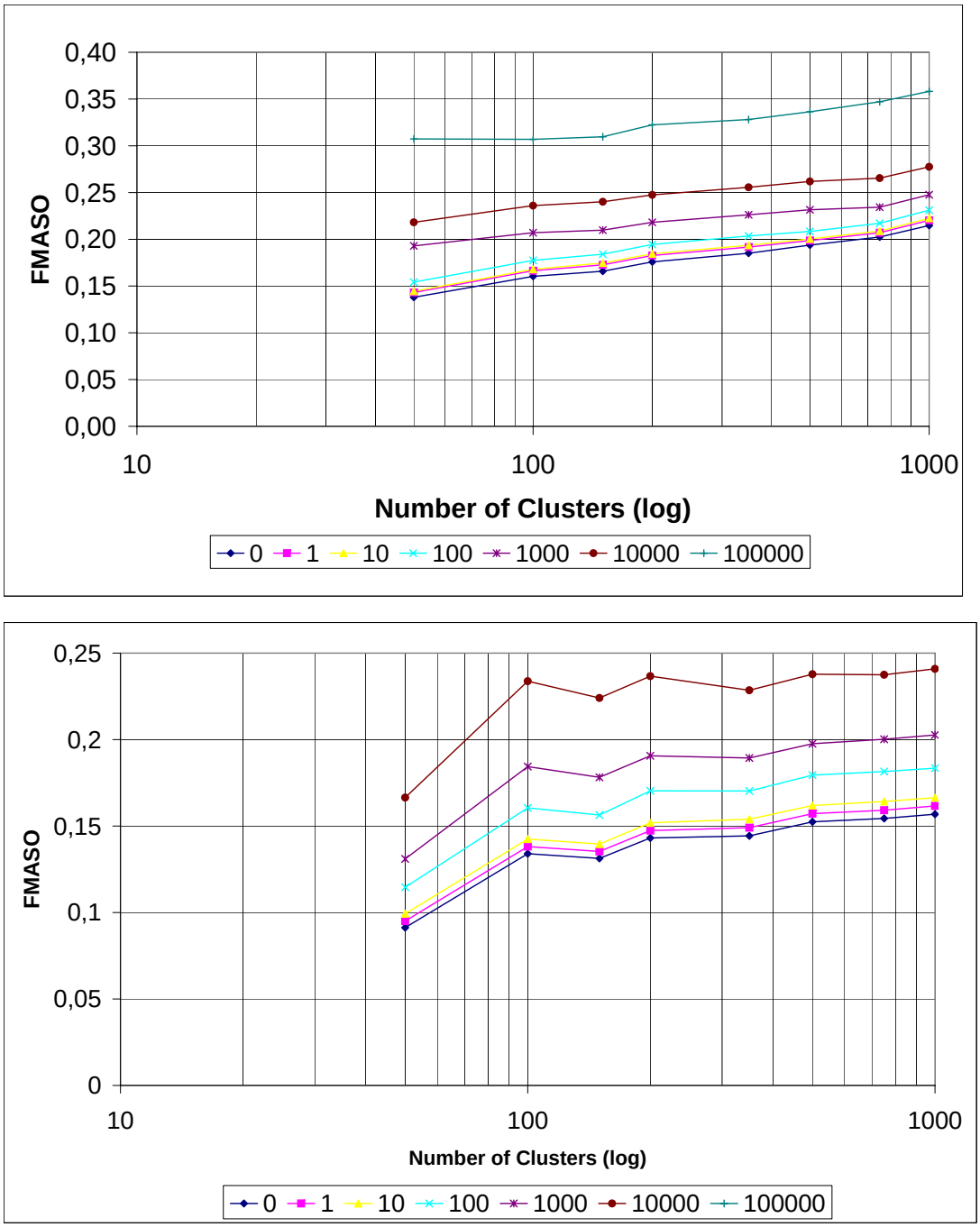


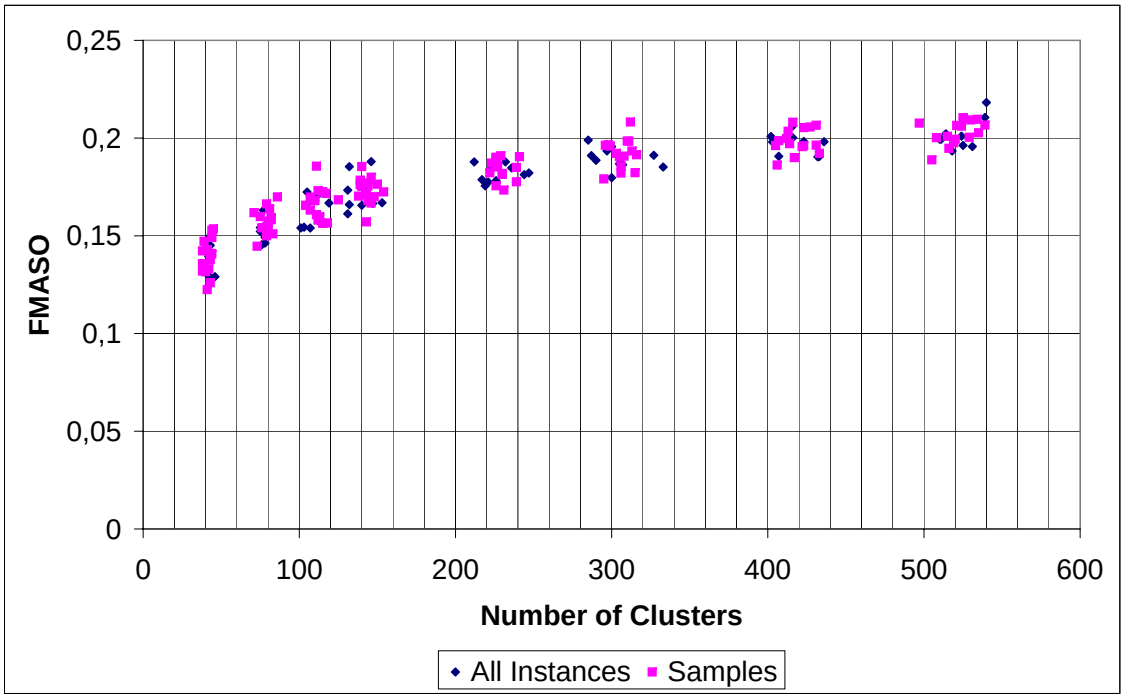
Figure 4.8: FMASO for different frequency support levels (query1, τ=0.2) for (a) GSO1 and (b) GSO2

4.3.7 Experiment 7: Sampling on Tagpath Clustering

By means of sampling the amount of data to be clustered and thus the required time of the clustering can be reduced. In this experiment we cluster only a fraction of the entire dataset to determine how sensitive the achieved results are regarding the size of the dataset. By varying the size of the dataset to be clustered we want to observe if more data is better and to which extent the measured quality changes.

As can be seen from Figure 4.9, the resulting FMASO values of the samples disperse more than the result values of the entire dataset. For high numbers of clusters for GSO2 the samples yield results which are a little bit worse than those of the entire dataset. This behaviour is not surprising since bigger datasets lead to more stable outcomes.

Conclusion: From the observations we conclude that a large dataset obtained from a large Web document collection delivers more sustainable results. But we also conclude that the potentially achievable improvements by using even bigger Web document collections are limited. Even though the used Web document collections comprise up to 10 million Web documents it would be possible to obtain Web crawls bigger by orders of magnitude. Obtaining such large Web crawls is very resource intensive and the relative stability of the achieved results questions the necessary effort towards performing bigger crawls.



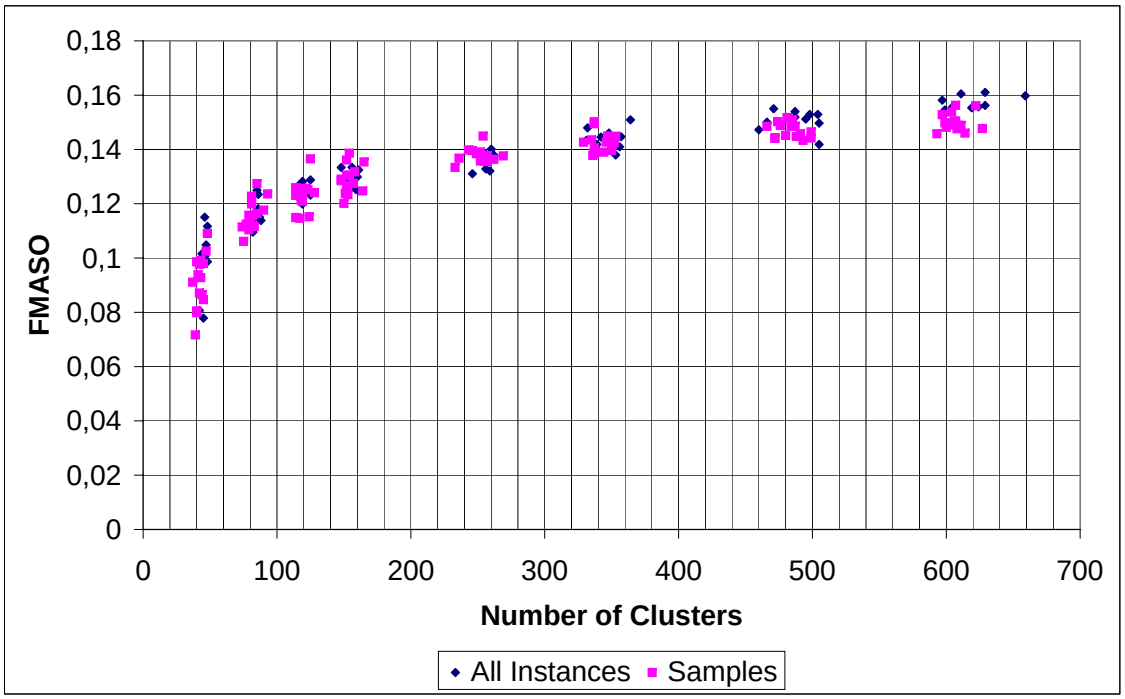


Figure 4.9: Sampling for tagpath clustering for (a) GSO1 and (b) GSO2)

4.3.8 Experiment 8: Frequent Itemsets in Comparison to Clusters

In this experiment we contrast the results obtained by performing frequent itemset mining to the results obtained by performing a K-Means tagpath clustering. The application of frequent itemset mining on Group-By-Path text span sets depicts a separate process. But since the results are also groups of terms supposed to stand in a sibling relation and can thus be compared to the results obtained by clustering we do not present it as a self-standing process. The processing was performed upon a Web document collection obtained by $Query1$ - $touris*$. As shown in table 6.2, 222037 itemsets for GSO1 and 318009 itemsets for GSO2 are the input for the processing.

The frequent itemsets which have been computed from the input itemsets have been ordered according to their support. For the top-n frequent itemsets with the highest support, the FMASO was computed. For n we chose values of 10, 20, 50, 100 and 150, beginning with n=200 in steps of 100 up to n=5000.

In figure 4.10, it can be seen that the frequent itemsets yield generally worse results than K-Means based clusters. Therefore, we can conclude that computing frequent itemsets is not an appropriate approach for finding sibling groups based on Group-By-Path raw sibling sets.

A possible explanation for this circumstance is that the top-n most frequent itemsets are mostly weakly varying variants of the same "sibling constellation". While performing clustering, all instances are assigned to clusters or even get the chance to form a cluster. Therefore, even comparatively rare sibling constellations can be found – in contrast to the support based frequent itemsets.

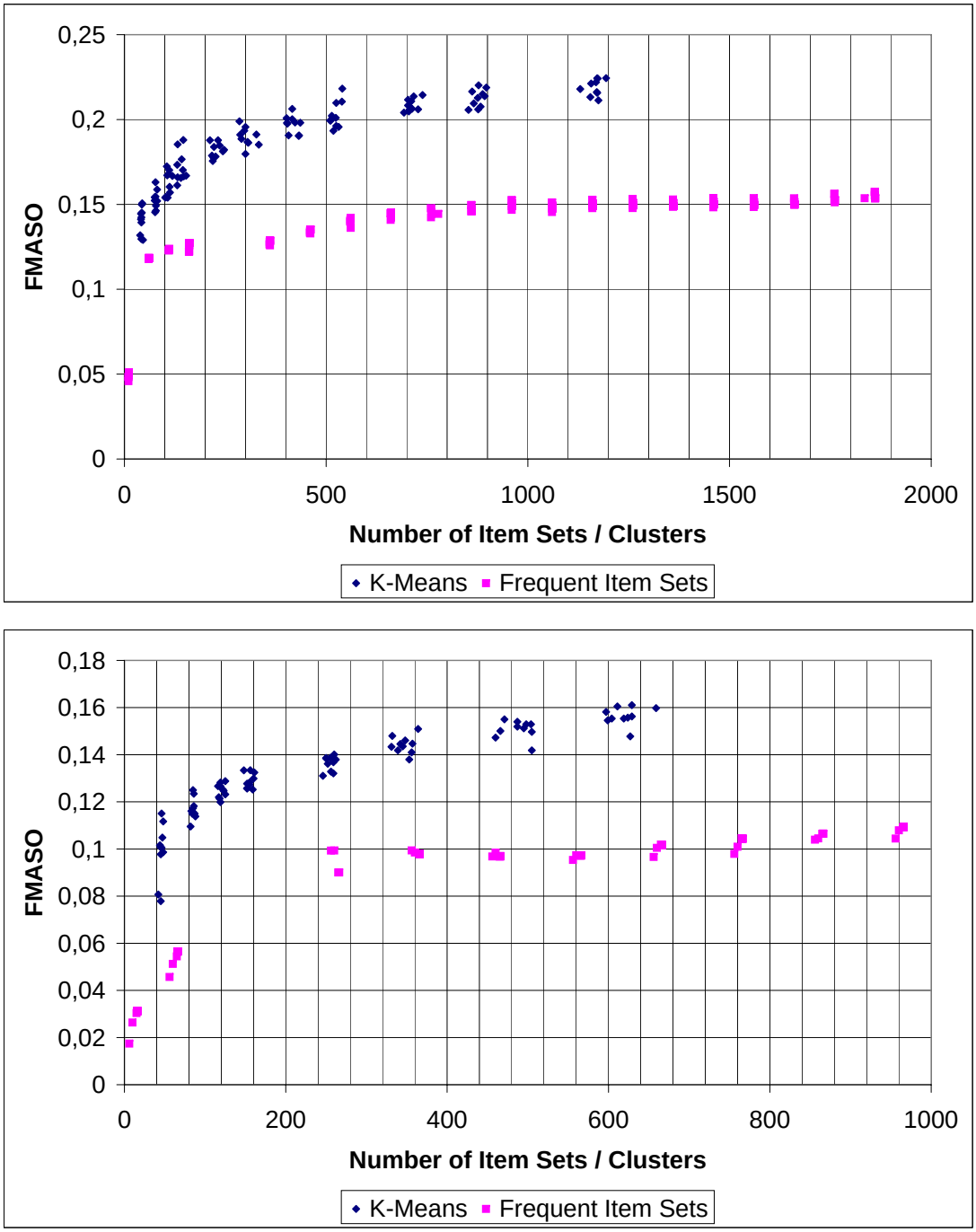


Figure 4.10: Comparison of frequent itemsets and K-Means generated cluster labels for (a) GSO1 and (b) GSO2)

4.3.9 Experiment 9: Tagpath Clustering in Comparison to Term Clustering

As described in section 4.1.4, there are two directions in which clustering can be performed. The clusterings of the previous experiments are tagpath clusterings. In this experiment we will contrast the results obtained by a term clustering with those of the previous tagpath clustering. For this experiment the dataset obtained by *Query*1 was used. For term clustering, the clustering is performed with values of K ranging from 10 to 350 in steps of 10. We do this since it is not clear in advance how many clusters should be generated. Since the clusterings also depend upon the initially chosen centroids, the clustering was performed ten times for each K with different random seed centroids.

Figure 4.11 shows the results of that experiment. The term clustering based sibling groups are superior to those obtained by tagpath clustering regarding the FMASO. For GSO1, term clustering enabled the improvement from 21.47% FMASO to 22.93% FMASO. While this improvement in quality was achieved, only 64 instead of 545 clusters had to be inspected. While doing this, only 253 terms (SOFICL) had to be inspected, compared to 2093 in the case of tagpath clustering. For GSO2 the improvement is even bigger: term clustering enabled the improvement of 15.88% FMASO to 19.59% FMASO. For this improvement the number of clusters to be inspected was reduced from 627 to 119. The corresponding number of terms (SOFICL) was reduced from 2249 terms to 627 terms.

It seems that the large number of tagpath clusters does not contribute towards discovering all reference groups. The tagpath clusters are to a high degree variants of similar sibling constellations. In contrast, term clusters are true partitions regarding the desired results. Each term can belong only to one cluster. Even rare terms get the chance to form meaningful clusters; the overall clustering is not dominated by variants of similar sibling groups.

Conclusion: The variant of XTREEM-SG using term clustering yielded better results than the variant using tagpath clustering.

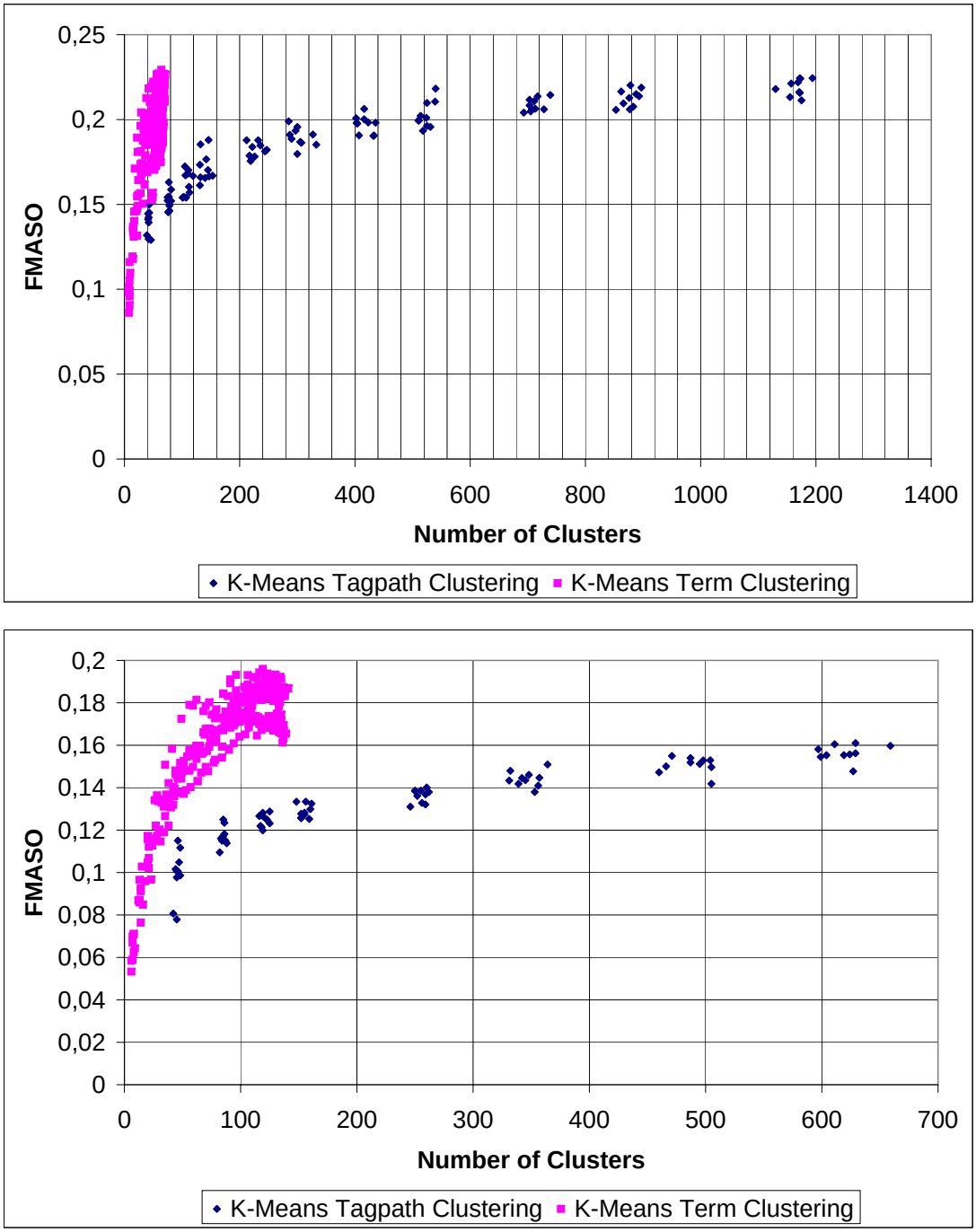


Figure 4.11: Comparison of K-Means tagpath clustering to term clustering for (a) GSO1 and (b) GSO2)

4.3.10 Experiment 10: Sampling on Term Clustering

In experiment 7 we observed the stability of the FMASO results on tagpath clustering. In this experiment we observe the stability of term clustering on samples of the dataset. Actually, we used only 500, 1000, 5000, 10000, 50000 and 100000 instances of the dataset. The number of clusters to be generated was set to K=100. Due to limited resources the experiment was limited to one random sample which allows only for a rough conclusion. It has to be borne in mind that to obtain more sustainable findings the experiment should be performed more systematically, considering more than one random sample for each sample size.

Figure 4.12 shows that with an increasing number of the clustered instances, the results yield a better FMASO. This is achieved while more clusters consisting of more than one term are created. On small numbers of instances the inhomogeneity can be assumed to be too big to allow for partitionings that reflect meaningful sibling relations. By observing the increase of achieved quality by higher numbers of incorporated instances, one can conclude that it is better to use the entire available data if there are no time constraints.

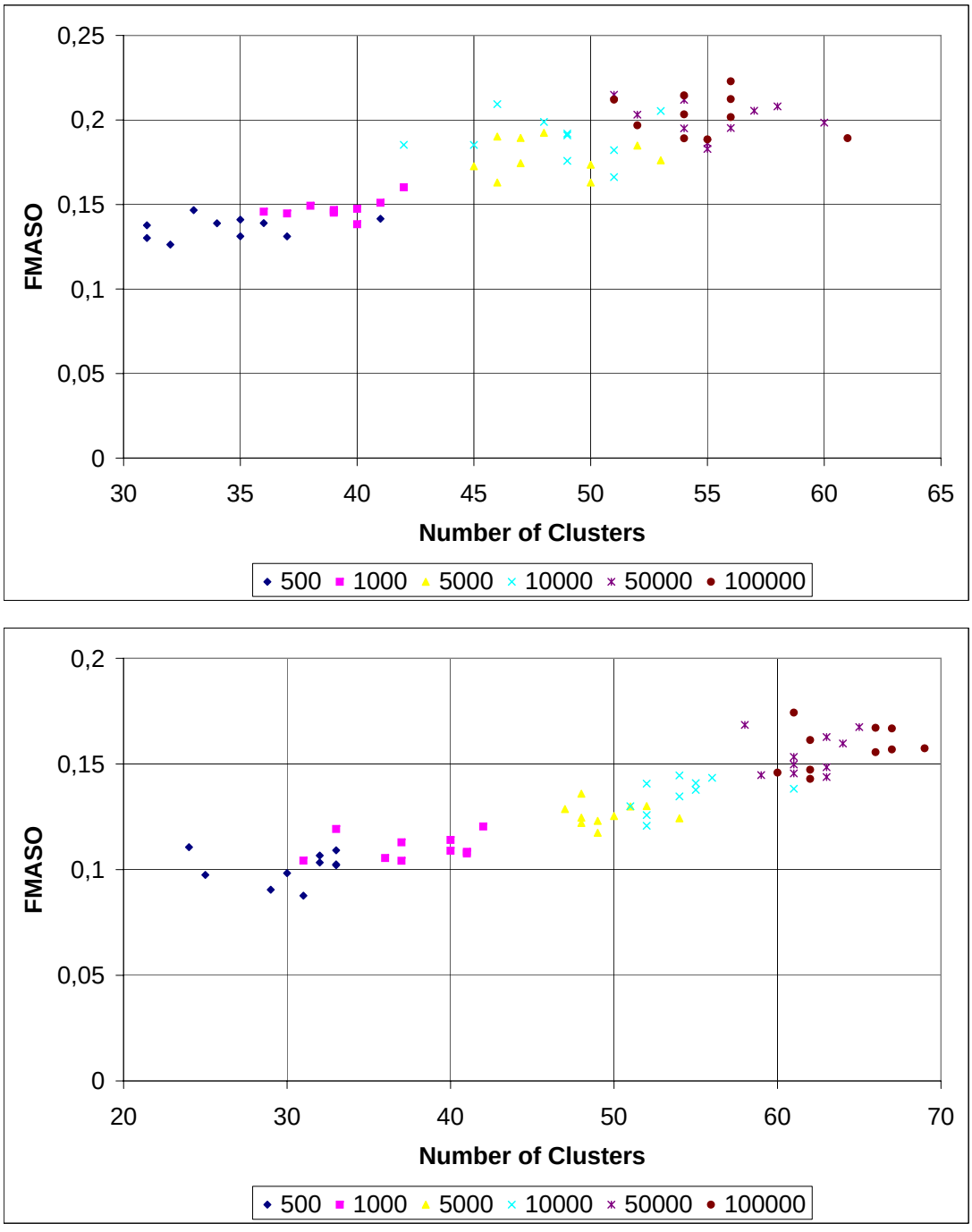


Figure 4.12: Sampling on term clustering for (a) GSO1 and (b) GSO2)

4.3.11 Results from Term Clustering

In the previous experiments we observed the results by means of the FAMSO, a value which is extracted from the particular findings and allows for quantified comparisons. In this section we show exemplary results obtained by term clustering to give an idea of how the results actually look like. Figure 4.13 shows clusters generated for the vocabulary of GSO1. Figure 4.14 and figure 4.15 clusters generated for the vocabulary of GSO2. The 283 terms of the vocabulary for GSO1 have been grouped into 100 clusters. The last column of figure 4.13 shows one large cluster of terms which cannot be clustered well. In the lower right corner there are 32 clusters with only one term. Those clusters are not helpful either to derive sibling relations. There are 63 clusters with more than one term, excepting the big noise cluster. The corresponding sibling groups from the reference ontology GSO1 are shown in the appendix B in figure B.1.

The 693 terms of the vocabulary for GSO2 have been clustered into 200 clusters. The last column of figure 4.15 shows one large cluster of terms which cannot be clustered well. In the lower middle of figure 4.15, there are 55 clusters with only one term. Those clusters are also not helpful to derive sibling relations. There are 119 clusters with more than one term, with the exception of the big noise cluster. The corresponding sibling groups from the reference ontology GSO2 are shown in the appendix B in figure B.2 and figure B.3.

We can see that in both results there are many clusters where plausible siblings can be recognized. On the other hand, there is still much room for optimization, since many outliers also can be found.

bus	balcony	early_season	ausblick
cabaret	iron	period	ausblickturm
car	kitchenette	driving_license	basketball_ground
rental	playground	id_card	bow_shooting_installation
beauty_farm	radio	buffet	chimney_room
beer_garden	shuttle_service	bull_fight	city_port
cure	tv	excursion	concert_house
short_trip	information	cross_country_ski_run	conference_folder
skittle_alley	sport	off_season	crazy_golf_course
gym	tourist	ski_run	cultural_installation
swimming_pool	bank	beach_volleyball_field	educational_journey
climbing_wall	library	nature_reserve	electronic_device
football_field	museum	date	fitness_course
pilgrimage	organization	event	hoarse_carriage
holiday_time	tourism_center	time	holiday_appartment
shooting_gallery	category	booking	holiday_equipment
single_room	region	daytime	ice_hall
visum	afternoon	holiday	living_thing
rowing_boat	day	meal	local_recreation_area
sailing_boat	morning	person	main_meal
starting_point	night	trip	masseur
contract	barbecue_area	ballroom	material_thing
day_trip	conference_room	disco	moor_bath
journey	fitness_room	massage	moor_therapy
first_class_hotel	bill	sight	night_cafe
middle_class_hotel	carriage	volleyball_field	non_material_thing
park	root	caravan	non_private_accommodation_equipment
situation	animation	diving_station	partially_material_thing
tent	musical_theatre	ruin	personal_thing
harbour	indoor_swimming_pool	airport	qualitative_time_concept
view	room	day_tour	recreational_installation
basilica	sauna	club	riding_crop
opera_house	service	equipment	spatial_concept
badminton	solarium	jazz_club	sport_installation
golf	concert	port	squash_field
kayak	musical	sport_equipment	sun_studio
swimming	sport_event	sport_shop	theatre_house
tennis	promenade	autumn	time_interval
open_air_theatre	thermal_spring	spring	touristic_installation
steam_bath	agreement	email	town_sightseeing_tour
turkish_bath	appartement	telephone	transport_vehicle
cinema	hotel	fitness_studio	trimmdichpfad
football	inn	sports_hall	volleyball_ground
theatre	motel	ball	wellness_installation
badminton_court	pension	cheque	wine_tavern
billiard_room	sea_view	fango	route_description
address	terrace	hair_dresser	adventure_holiday
city	washing_machine	parking_lot	accommodation_equipment
country	banquet	public_holiday	beach_view
town	bicycle	accommodation	art_exhibition
squash_court	business_event	camping	house_description
weights_room	conference	culture	kiosk
elevator	congress	gallery	sport_holiday
hair_dryer	cultural_event	money	cafe
lounge	exhibition	price_list	mud_therapy
minibar	seminar	shop	thing
panorama	drier	area	sanatorium
trail_map	golf_course	currency	place
beach	table_tennis_table	passport	tour_operator
boat	tennis_court	shopping_center	football_game
casino	walking_trail	videorecorder	fishing_equipment
castle	farm	human_activity	winter
group	holiday_village	vehicle	health_club
law	thermal_bath	animal	seminar_house
pub	youth_hostel	plant	bed
ship	ferry	station	sledge
shower	menu	presentation	change_office
wedding	aerobic	bowling_alley	heritage_town
whirlpool	billiard	room_equipment	bar
yacht	brochure	party	summer
yacht_port	kindergarden	city_wall	

Figure 4.13: Resulting clusters from term clustering for GSO1

monastery	casino	instruction	beach_view	air
old_town	dune	water	north_shore	care
city_museum	sand	facial_therapy	pedestrian_area	cat
culture	parachuting	massage	promenade	church
night_life	water_sport	shooting_range	south_shore	communication
shopping	backwater	buy	avenue	farm
sightseeing	bicycle_tour	coast	coastal_resort	father
talk	horse_tour	recreate	fish	frontier
menu	journey	sea	fishing_equipment	level
mouse	tour	district	fishing_rod	mother
dog	tour_guide	walking_trail	nature	offer
dog_care	booking	badminton	attic	opera
horse	reservation	bowling	diving_station	pigeon
ox	summer_holidays	cleaning	metropolis	place
picnic	eating	fitness_studio	banquet	produce
race	harbour_city	handball	conference	walk
rat	panorama_view	ice_hockey	congress	fresh_water
afternoon	spare_time	squash	exhibition	saltwater
day	cliff	table_tennis	guided_tour	surfboard
morning	hike	baby	night	hand_care
week	hiker	club	park	pork
beer_garden	racket	group	seminar	city
skittle_alley	soloist	shore	sleep	country
spit_of_land	table_tennis_table	camping	workshop	state
cross_country_skiing	tennis_ball	downhill_ski	youth_hostel	time_interval
ice_skating	tennis_racket	hiking	aunt	town
snowboard	aerobic	kiosk	bat	back_massage
snowboarding	billiard	open_air_theater	daughter	body_massage
area	billiard_equipment	tennis_lessons	grandchild	face_massage
floor	cycler	wellness_offer	grandfather	water_gymnastics
bird	exchange_office	autumn	regimen	agent
human_activity	fitness_training	spring	sibling	camp
saltwater_fish	concert	summer	son	cross_country_ski_run
tree	dancing	winter	uncle	ski_run
vesper	event	matinee	mono_ski	cinema
agency	festival	moor	vista_point	theatre
drive	musical	invoice	water_ski	hairdresser
family	business_people	standardization	ballet	secretary_service
tourist	dog_service	basketball	dolphin	sport_shop
organic_food	sports_facilities	cycling	elephant	early_season
table_tennis_ball	visiting	diving	giraffe	main_season
vegetarian_food	working	football	mall	off_season
cosmetic_care	day_time	golf	monkey	season
foot_care	island	sailing	ball	easter_holiday
manicure	region	soccer	boy	holiday
hair_cut	traveling	swimming	female	ski
nail_design	service	tennis	girl	excursion
fango	hill	volleyball	kin	wedding
fitness_room	midday	barbecue	male	east_shore
indoor_swimming_pool	national_park	cleaning_service	person	east_side
mini_golf_area	nature_reserve	garden	relax_weekend	north_side
sauna	art_gallery	sea_view	teenager	south_side
solarium	christmas_special	tennis_court	accommodation	west_side
steam_bath	golf_course	town_centre	culture_tourism	cultural_activity
surf_school	museum	climbing_wall	shop	guest
ground_floor	weekend_special	horse_riding_lessons	tourist_information	village
sport_offer	whirl_pool	horse_riding_school	traveling_by_air	acquaintance_week
upper_floor	breakfast_buffet	cultural_institution	baby_sitter_service	action_affecting_an_object
bungee_jumping	dinner_buffet	gallery	billiard_table	airpark_guest
city_trip	surfboard_rental	market_place	disco	awaking_service
fishing	badminton_court	ball_game	sports_hall	base_ball_bat
hunting	football_pitch	ball_room	act	basket_ball_game
professional_sportsman	squash_court	billiard_ball	gourmet	basketball_ground
breakfast	beauty_day	art	driver	beach_chair_rental
brunch	make_up	performance	musician	beach_promenade
buffet	permanent_make_up	cafe	art_exhibition	riding
fast_food	adventure	hair_dresser	pageant	circus
sport_equipment	pilgrimage	harbour_area	panorama	zoo

Figure 4.14: Resulting clusters from term clustering for GSO2 - part 1 of 2

day_trip	bank	trip_for_singles	beach_volleyball_ground	harbour_round_trip
digestive	embankment	type_advice	beauty_relax_weekend	healthiness_holiday
heat	fortress	vanishing_cream_pack	beauty_temple	healthiness_tourism
cabaret	organizer	view_to_the_eastern_sea	bicycle_renting_agency	heath_tour
mini_golf	holiday_village	visagist	billiard_queue	holiday_arrangement
playground	relaxing_holiday	vista_tower	brenn_ball	horse_renting_agency
thermal_bath	side	volleyball_ground	charter_excursion	horse_riding_offer
action	top_hotel	water_sports_institution	chimney_room	jazz_breakfast
animal	yacht_port	water_sports_offerings	city_forest	keep_fit_course
donkey	yacht_rental	weekend_arrangement	colour_light_therapy	landscape_protection_area
mammal	bay	weekend_quest	conference_guest	light_diet
plant	cape	wellness_institution	cosmetic_therapy	living_creature
formula_one	adult	whit_holidays	countable_concept	mass_concept
overnight_stay	child	whit_sun	creativity_holiday	master_hairdresser
short_trip	pensioner	wine_tavern	cure_city	material_thing
boat_rental	cycling_tour	working_person	dancing_tee	moor_bath
catering	weekend_trip	horse_riding_yard	default_root_concept	moor_therapy
intangible	city_harbour	informing	deluxe_journey	museum_guided_tour
ruin	harbour	island_round_trip	device_state_change	motor_bicycle_renting_agency
canyon	view	basilica	dog_doctor	night_cafe
temple	advent	castle_complex	dog_hair_cutter	overnight_stay_possibility
puppet_theatre	christmas	cathedral	dog_psychologist	partially_material_thing
symposium	easter	city_wall	dog_race	permanent_eyebrow_make_up
brother	holiday_place	national_art_gallery	educational_holiday	permanent_guest
cultural_event	dance	oratory	educational_journey	permanent_lid_make_up
gala	dancing_night	gym	end_of_year	permanent_lip_make_up
apartment	music	health_club	event_offerings	principal_meal
castle	sport	swimming_pool	event_trip	produce_information_carrier
guest_house	contract	climbing	excursion_goal	purge_day
hunting_castle	employee	hang_gliding	eyebrow_correction	putting_the_shot_ball
motel	gala_dinner	skiing	eyelashes_correction	qualitative_time_concept
pension	organization	car_rental	family_celebration	recreation_area_close_to_a_town
sea_side	program	hotel_garden	fango_application	recreation_location
sightseeing_flight	registration	shopping_tourism	fango_therapy	recreational_institution
sightseeing_tour	bargain	car_race	festival_house	recreational_offer
table_tennis_racket	relaxing	holiday_maker	fish_rod_rental	regimen_offer
continent	sight	horse_race	flat_offer	regimen_organization
street	time	grandparents	forest_border	renting_agency
actor	midnight	grilling	formula_one_tour	reproducing_service
face_mask	snack	inner_city	free_way	romantic_day
forest	thing	chess_tournament	gala_menu	rural_district
mountain	winter_garden	exchanging_money	gliding_field	sailing_boat_rental
peninsula	aroma_bath	chicken	graduation_travel	salt_backwater
river	arrangement	music_house	hand_peeling	salt_bath
valley	bow_shooting_range	base_ball	handball_game	sea_territory
desert	business_dinner	jazz_club	travel_organizer	self_employed_person
lake	business_event	cosmetician	trimmdichpfad	service
airport	canoe_tour	peeling	muenster	several_days_trip
sports_holiday	city_centre	craft_work	arrival	shoeblack_service
train_station	city_guided_tour	open_air_bath	summer_season	short_trip_tourist
suburb	concert_house	work_of_art	ski_lift	skiball
adventure_holiday	congress_city	kursaal	holiday_time	skin_diagnosis
family_holiday	cultural_offerings	bullfight	west_shore	spare_time_possibility
holiday_home	educational_event	weekend	boat_round_trip	spatial_concept
holiday_special	experience_gastronomy	turkish_bath	shuttle_service	sport_trip
billiard_room	hanseatic_city	beauty_farm	appetizer	sports_institution
sanatorium	lake_side	forest_side	catering_company	steep_bank
sledge	lunch_packet	round_trip	middle_class_hotel	stone_grave
therm	outskirts	kindergarden	ice_hall	stone_shore
bowling_alley	road_lane	breakfast_service	balloon_trip	sweet_water_fish
library	seminar_house	sportsman	animation	testing_week
camping_ground	short_holiday	skate	child_care	theater_arrangement
cruise	soccer_game	grandma	water_hiking	theater_house
hotel	watching_tv	exhibition_opening	golf_tournament	theater_weekend
pub	dinner	fisticuffs	sun_studio	tightening_therapy
winter_holiday	farewell_dinner	club_holiday	nature_experience	tour_offer
beach	lunch	grill_place	sports_event	town_border
crane	provider	business_traveler	shoulder_season	traffic_route
stork	snow			

Figure 4.15: Resulting clusters from term clustering for GSO2 - part 2 of 2

4.4 Conclusion

We have presented XTREEM-SG, a method for the discovery of semantic sibling relations among terms on the basis of structural conventions in Web documents. XTREEM-SG processes Web documents collected from the WWW and thus eliminates the need for a well-prepared document corpus. Furthermore, it does not rely on linguistic pre-processing or natural language processing resources. So, XTREEM-SG is much less demanding of human resources. Since it does rely on Web document structure, the algorithm is language and domain independent. A further important advantage of this algorithm is that here we can process multiword terms in the same way as words.

The application of Group-By-Path with K-Means clustering reduced the initial candidate sets substantially by retaining most of the quality measured by the F-Measure on average sibling overlap. In [Cimiano and Staab, 2005] is described, that Cimiano and Staab have obtained average sibling overlap F-Measures from 12.40% to 14.18% on the tourism GSO (GSO1). With these results, they realized a significant improvement in contrast to Caraballo's method [Caraballo, 1999], which gave a sibling overlap F-Measure of 8.96%. We can get good results in this evaluation measure too. Our best result using tagpath clustering gives an F-Measure on average sibling overlap of 21.47% using a K-Means clustering with 1000 clusters and labelling the clusters by using all features which have a support within the cluster of 20%. For term clustering an F-Measure on average sibling overlap of 22.93% was achieved while producing 64 sibling groups. These are significant improvements and confirm that the XTREEM-SG approach delivers good results for mining sibling relations.

The amount of clusters influences the abstraction forced on the constitution of the resulting sibling groups. For real world settings, the expert may decide to handle the trade-off between achievable quality and the amount of generated information according to his objectives of how detailed the generated results should be. This gold standard evaluation does not capture this aspect, but this can be seen by manually inspecting the results. Staab and Hotho reported [Staab and Hotho, 2005] that the results of their approach get better judgements by a posteriori evaluations by domain experts where the results are regarded as good and helpful. The same holds true for the results obtained with XTREEM-SG. Many of the obtained sibling groups depict a plausible sibling constellation. On the other hand, this is not astonishing as the results are based on many thousands, often hand crafted, manifestations of sibling items on the WWW.

5 Learning Sibling Groups Hierarchies - XTREEM-SGH

In chapter 4, we have shown that the XTREEM-SG method is capable of finding sibling groups that are semantically plausible with higher quality than possible in previous methods. This was done by a flat K-Means clustering. While clusterings have been performed, a trade-off between achieved quality and the amount of information to be inspected became apparent. The more clusters have been generated, the better the overall quality measured by the FMASO became. The more fine granular clusters were created, the more plausible sibling groups could be obtained but for the price that larger amounts of information have to be inspected. The number of clusters to be generated, which is not known but underlies the trade-off described above, has to be stated before the clustering is performed. Once the results are generated and the ontology engineer recognizes that the granularity of the results is too low or to high, a new clustering has to be performed. And the granularity which is appropriate to the ontology design objectives might vary largely across the concepts depicted by the results. In this chapter we will discuss how to improve this situation by performing a hierarchical clustering. Clusters depicting sibling groups will be arranged in a hierarchy where the human ontology engineer can browse sibling groups according to his desired granularity.

Cluster hierarchies provide views on the analyzed dataset at different levels of abstraction. The varying granularity is said to be ideal for exploration and visualization [Zhao and Karypis, 2002]. The availability of sub-clusters can be beneficial since some of the domains for which data is analysed also rely on hierarchical structures such as biological taxonomies (phylogenetic trees) [Duda et al., 2000]. In our application field of ontology engineering, human expert needs only to inspect the hierarchy down to his desired granularity on structuring. To achieve such a cluster hierarchy as an outcome, we propose and describe a procedure where Bi-Secting-K-Means style clustering is applied upon a Group-By-Path dataset. This method is called XTREEM-SGH (XTREEM for Sibling Groups Hierarchies).

First we describe the considerations which lead to the design of the XTREEM-SGH procedure, then we will contrast the quality of results obtained by a hierarchical clustering method, against the quality obtained by a flat K-Means clustering already presented in chapter 4.

5.1 Hierarchical clustering for Sibling Groups Hierarchies

The aim of the XTREEM-SGH approach is to build a hierarchy of clusters depicting sibling groups so that the user is not forced to decide for a certain number of clusters in advance. To produce a hierarchical clustering, two major types of clustering are to be considered. On the one side there is the bottom-up, agglomerative hierarchical clustering [Sneath and Sokal, 1973, King, 1967],and, on the other side, there are the divisive hierarchical clustering methods [Jain and Dubes, 1988].

In hierarchical agglomerative clustering the cluster hierarchy is built in bottom-up fashion. First the instances form separate clusters. Then, in each iteration, the two most similar clusters are merged. There exist different variants of hierarchical agglomerative clusterings which differ on the used cluster-to-cluster distance function, for instance *single link* [Sibson, 1973], *average link* [Voorhees, 1986] and *complete link* [Defays, 1977]. A major drawback of hierarchical agglomerative clustering is its complexity of $O(n^2)$ with single link or worse of $O(n^2 \log n)$ with complete linkage [Murtagh, 1983], which makes its application inappropriate on large datasets. The other types of hierarchical clustering algorithms are the divisive hierarchical clustering algorithms which perform a top-down procedure. First, all instances are together in one cluster. Then this cluster is iteratively split. Bi-Secting-K-Means [Steinbach et al., 2000] is a popular representative of a divisive hierarchical clustering, often used in the clustering of textual documents. The complexity of Bi-Secting-K-Means is $O(n \log n)$ for the variant where a complete hierarchy is created.

5.1.1 Hierarchical Term Clustering

As already described in chapter 4, a clustering for sibling relations can be conducted as a tagpath clustering where labelled clusters are obtained or as a term clustering, where the terms are constituting clusters. The best results regarding sibling relations have been achieved by means of term clustering. Our first experiment is, therefore, to apply agglomerative hierarchical clustering and Bi-Secting-K-Means for term clustering. This means that terms represented by vector of occurrence in sibling sets are clustered, yielding a binary tree where terms are finally the leafs of the tree. We also apply Bi-Secting-K-Means in a manner to produce a complete hierarchy, without stating a (not even roughly) known number of clusters in advance. Bi-Secting-K-Means can be applied to yield a fixed number of clusters. But in our scenario the number of clusters is not even roughly known nor do we know which is the suited strategy to decide for the clusters to be split until the desired number of clusters is achieved. By producing a complete hierarchy we avoid the need to decide for a K and for a strategy to choose the next cluster to split.

For the obtained hierarchies of clusters it is not possible to measure the achieved quality according to FMASO. The clusters are not partitions depicting sibling

groups in such a way that one set of sibling groups can be compared to the reference set of sibling groups. This could only be done by incorporating more "heuristics" such as where the hierarchy is to be cut so that groups of siblings can be made explicit for agglomerative hierarchical clustering and for using Bi-Secting-K-Means with a fixed number of K cluster to be produced. But in both cases the hierarchical characteristic would get lost.

Next we will have a brief look at the results obtained while performing hierarchical term clustering. Figure 5.1 and 5.2 shows the resulting cluster hierarchies for agglomerative hierarchical clustering and Bi-Secting-K-Means clustering. Figure 5.3 shows a fraction of the hierarchy produced by Bi-Secting-K-Means clustering where details can be observed. We applied hierarchical agglomerative clustering with the Unweighted Pair Group Method with Arithmetic Mean - UGPMA [Jain and Dubes, 1988] heuristic which is supposed to prevent the creation of chains of clusters for the sake of a relatively high complexity of $O(n^2 \log n)$. But for the relatively smaller number of terms to be clustered this complexity is acceptable. The hierarchical agglomerative clustering shows only a relatively small chaining effect, whereas the Bi-Secting-K-Means clustering reveals that chaining is a more serious issue.

While one manually inspects that hierarchy, the disadvantage of the two exclusive clustering approaches applied as term clustering become apparent again. In an exclusive clustering approach such as K-Means (and hierarchical agglomerative clustering and Bi-Secting-K-Means) a membership in clusters is exclusive: an instance (term) can belong to only one cluster. This disadvantage was already pointed out while K-Means was applied for term clustering. But since the quality of the clustering cannot be measured according to the FMASO, we cannot quantify to which extent the clusterings represent siblings which are in accordance with the reference ontologies regarding the sibling relations. The rough impression is that a term clustering with exclusive cluster membership is too restrictive regarding providing sufficient suggestions of plausible siblings to the ontology engineer. Concepts can have many siblings which are plausible and if one recognizes an error where the siblings depicted by the cluster hierarchy are not plausible, then one has no alterative suggestion. A term clustering which yields only a small number of information to be inspected by this cannot compensate the disadvantage of the limited number of siblings which can be observed for a concept. The idea is to perform hierarchical clustering as tagpath clustering. In a tagpath clustering, clusters need to be labelled and, therefore, the terms/features/concepts that take place constitute more than one sibling constellation. The circumstance that tagpath clustering is here more appealing is also the reason why we did not further investigate whether Bi-Secting-K-Means can be applied to produce a fixed number of clusters which could thus be compared. The disadvantage of exclusive clustering would still be prevalent.

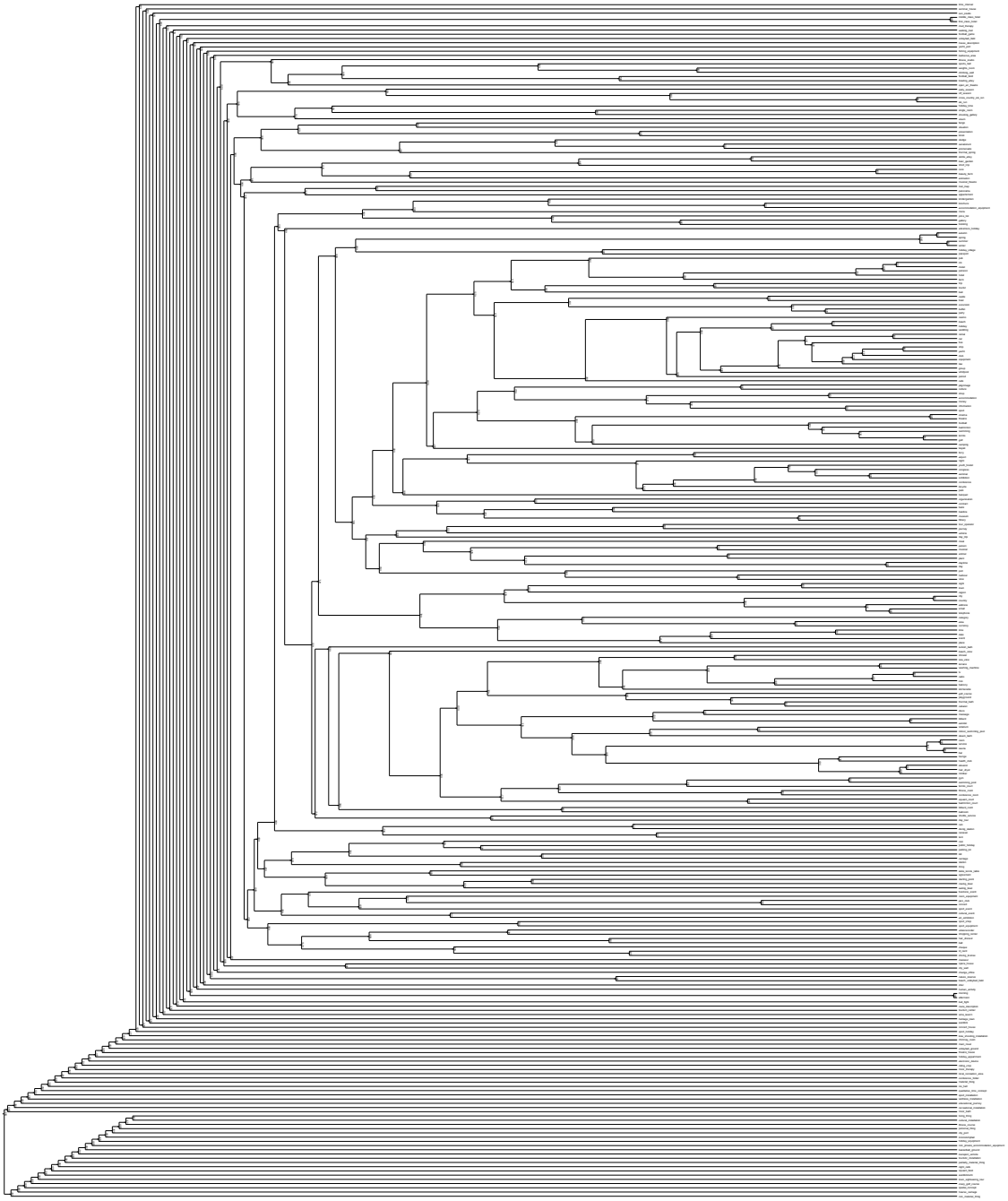


Figure 5.1: Dendrogram of a agglomerative hierarchical clustering with UGPMA metric on a GBP dataset (term clustering, GSO1)

Figure 5.2: Overall hierarchy of terms obtained with Bi-Secting-K-Means (term clustering, GSO1)

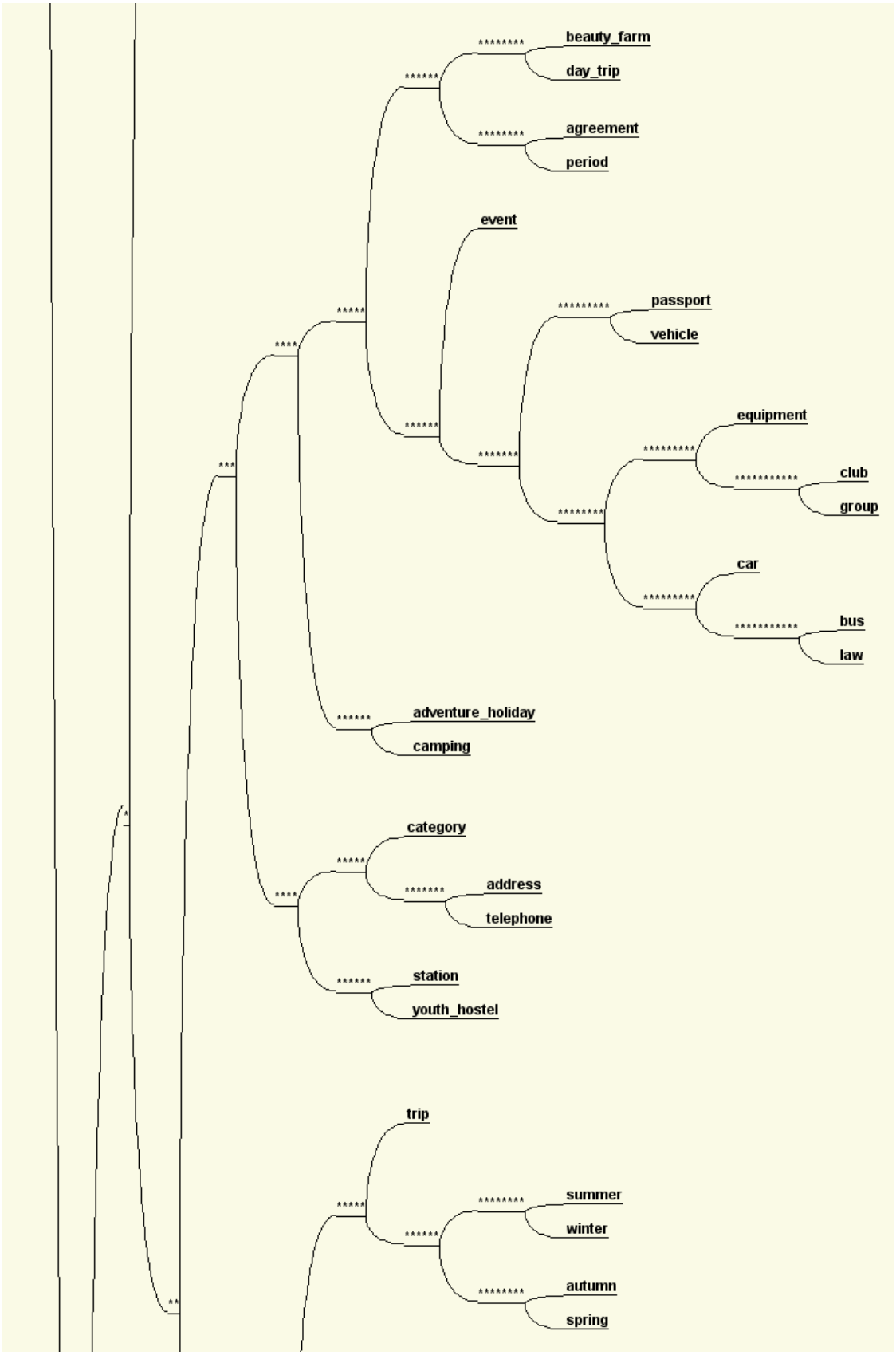


Figure 5.3: Fraction of the hierarchy of terms obtained with Bi-Secting-K-Means (term clustering, GSO1)

5.1.2 Hierarchical Tagpath Clustering

We aim at creating a tagpath clustering where terms can become a part within several sibling groups. The relatively high number of tagpaths to be clustered prevents the application of agglomerative hierarchical clustering with its high complexity on a dataset that is a big as the Group-By-Path datasets we process. We, therefore, concentrate on applying the more efficient Bi-Secting-K-Means clustering.

The first experiments with a Group-By-Path dataset have been done while performing tagpath clustering with Bi-Secting-K-Means and a K was to be specified. Figure 5.4 shows a screenshot of the Relfin UI [Schaal et al., 2005] where the table at the left side depicts the collection of K produced clusters. The left side of the screen shows the cluster label for the one selected cluster.

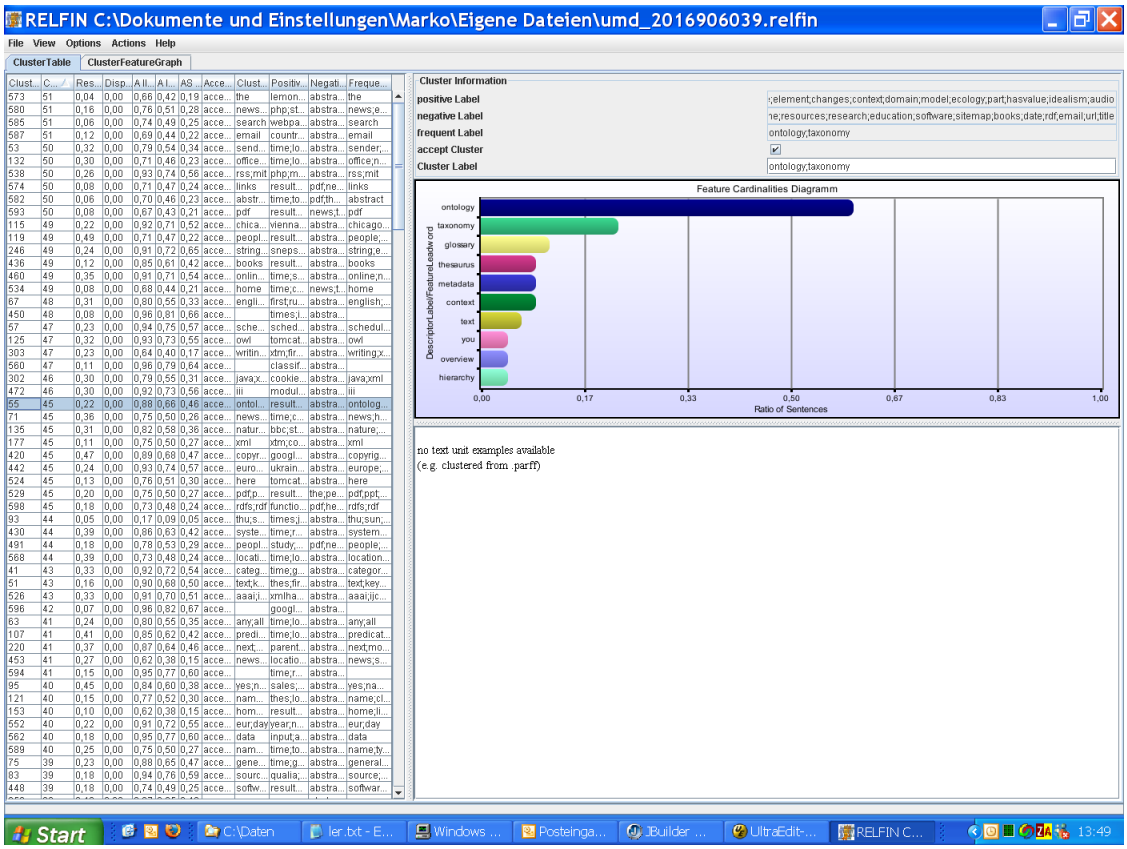


Figure 5.4: Screenshot of Relfin where a Group-By-Path dataset is clustered into a fixed number of K clusters by Bi-Secting-K-Means.

We will further apply Bi-Secting-K-Means in a way to produce a hierarchy of clusters without the need to specify a fixed number of clusters beforehand. The number of clusters is not known and, more importantly, we investigate a scenario where the human can inspect sibling groups down the hierarchy and

observe varying granularity. This variant of Bi-Secting-K-Means where a "complete" hierarchy of clusters is produced is already mentioned in [Steinbach et al., 2000], where splitting is performed until clusters contain only single instances. This is not the usually performed way of applying Bi-Secting-K-Means where a strategy [Savaresi et al., 2000] to decide for the next cluster to split is required and only a limited number of clusters is obtained. Such strategies are, for example, to select the largest cluster, the most inhomogeneous cluster given by within cluster variance or other measures such as the cluster residue [Schaal et al., 2005]. By producing a "complete" cluster hierarchy where finally the instances are leafs; a binary hierarchy is obtained, analogous to agglomerative hierarchical clustering. For practical reasons we do not produce the complete hierarchy where hundreds of thousands of clusters would be produced but we limit the depth to 15 levels. A user is not likely to be interested in even deeper hierarchies. By producing hierarchies of up to 15 levels hierarchy depth, the overall number of clusters produced exceeds the number of clusters a user is likely to inspect and thus encompasses the relevant amount of clusters. In the evaluation experiments we will vary the used hierarchy depth up to 15 levels thus investigating less depth hierarchies too.

The above described way of applying Bi-Secting-K-Means to produce a partitioning [Schaal et al., 2005] where instances are only part of one cluster dismissed the advantage of the hierarchical clustering where the granularity can be adapted while one observes the clusters. This can be prevented by showing the clusters to the user in a hierarchical way where with particular emphasis super-clusters and sub-clusters should be visible at the same time. While the user inspects such a hierarchy, a super and its sub clusters are both shown within one inspection run. This is different from the widespread application of Bi-Secting-K-Means as a partitioning clustering algorithm where only a super-cluster or its sub-clusters are used at the same time. But one should keep in mind that for our tagpath clusterings we are not interested in instances to be partitioned, but in labelled clusters. The clustering is applied as a kind of "oracle" for producing a reasonable number of patterns. The advantage of a hierarchical clustering is that patterns with varying granularity can be proposed in a run together and thus "super sibling groups" and more detailed "sub sibling groups" can be inspected.

5.1.3 XTREEM-SGH Procedure

The XTREEM-SGH process is in principle the same as the XTREEM-SG process described in section 4.1 and depicted in figure 4.1 with the exception that instead of K-Means, Bi-Secting-K-Means is used for clustering. Since Bi-Secting-K-Means is a hierarchical clustering algorithm, not merely a flat collection of clusters is produced as done by K-Means but a cluster hierarchy.

5.2 Evaluation Methodology

For the evaluation of sibling groups hierarchies (and subsets and subsets of hierarchies) we use the same inputs, references and evaluation criteria as already used for the evaluation of sibling groups.

The evaluation criteria is the FMASO described in section 4.2.1. From the cluster hierarchy we obtain "flat" set of clusters which are to be evaluated according to the FMASO criteria. From the overall cluster hierarchy we obtain several subsets of clusters and subsequently sibling groups obtained by labelling clusters according to the labelling method described in section 5.2. Only the set of unique patterns is compared as FMASO is defined to compare a set of sets to another set of sets, where doublets are excluded. This is also appropriate with the general goal of simulating a human evaluator who operates a GUI with the cluster hierarchy. If a parent and its child cluster has the same label this could be made apparent in the UI so that a human has not to inspect both.

We consider two strategies about how the sibling groups hierarchies are observed and how the collection of clusters is obtained. First we consider all sibling groups up to the hierarchy level L together. Different hierarchy levels are mixed together as a user would also have observed the parent sibling groups while he moves down the hierarchy performing a breath first traversal. We denote this as "Up-To-Hierarchy-Level-L" strategy. A special case is the "Complete-Cluster-Hierarchy", which corresponds to "Up-To-Hierarchy-Level-L" strategy where L is the deepest level which was computed and thus the entire computed cluster hierarchy is used. Moreover, we consider the strategy of a user who learned that the high quality sibling groups are not to be found close to the hierarchy root and who moves to a certain hierarchy level and will only observe those tree layer in one evaluation run. Only the clusters which are obtained at a certain hierarchy level are thus evaluated as one automatically obtained result set such as all clusters of level 2,3,4, ... up to level 15. We denote this strategy as "Separate-Hierarchy-Levels". Indeed, in this strategy the hierarchy is not used, this strategy is used as a contrastive result.

The comparison reference are the two gold standard ontologies described in section 8.3.2 depicted as GSO1 and GSO2. The dataset is the same as those denoted as $Collection1$ in chapter 4 obtained by $Query1$ – "touris*" from a 9.5 million Web document collection focused on tourism. The feature space is given by the vocabulary obtained from the labels of the concepts of GSO1 and GSO2. The parameter τ of the cluster labelling strategy described in section is set to $\tau = 0.2$.

5.3 Experiments

In the first experiments we will contrast K-Means and Bi-Secting-K-Means. Then we will compare different ways of accessing the cluster hierarchy produced by Bi-Secting-K-Means. Lastly we will look at the hierarchy levels where the best results have been obtained.

5.3.1 Experiment 1: K-Means in Comparison to Bi-Secting-K-Means

There are different strategies to obtain subsets of the cluster hierarchies which are described in the previous section. For the first experiment we apply all three hierarchy access variants and do not distinguish results regarding the "Up-To-Hierarchy-Level-L", "Complete-Cluster-Hierarchy" and "Separate-Hierarchy-Levels" strategies for obtaining hierarchy subsets since the goal is to contrast Bi-Secting-K-Means with K-Means. In experiment 2 we differentiate between the different variants.

Figure 5.5(a) shows that the quality of the results obtained by the K-Means clustering algorithm are, in general, as good as or better than those obtained via Bi-Secting-K-Means clustering. For the second reference, shown in figure 5.5(b), K-Means shows even better results, Bi-Secting-K-Means results being, with a few exceptions of some outliers, generally worse.

Conclusion: For the Group-By-Path based dataset, we cannot support the observation of Steinbach, Karypis and Kumar that "The Bi-Secting-K-Means technique is better than the standard K-Means approach" [Steinbach et al., 2000]. Our observation is that K-Means is as good as or even better than Bi-Secting-K-Means clustering – for our scenario, our goal, and our dataset. The scenario where Steinbach, Karypis and Kumar compared Bi-Secting-K-Means to K-Means was different; there K clusters produced by K-Means have been compared to K clusters produced by Bi-Secting-K-Means. This means that, in general, which clustering algorithm yields the best results might depend on the actual setting. Bi-Secting-K-Means yielded also worse results than agglomerative hierarchical clustering while creating a concept hierarchy [Cimiano et al., 2004b] and was criticized for its low traceability of the cluster creation. The traceability of results produced by Bi-Secting-K-Means in a high dimensional space is low. The bad splitting decisions of Bi-Secting-K-Means at the high level cannot be undone, whereas for K-Means, instances can change their cluster membership if appropriate. Bi-Secting-K-Means seems not able to split in such a way that on deeper hierarchy levels plausible sibling groups are created compared to K-Means.

The sparseness of vectorized sibling sets is higher than those of vectorized text documents so that there are fewer features which might enable useful splitting, but this is only a hypothesis and needs further investigations which are beyond our focus of acquiring sibling relations from Web documents.

But since the difference of Bi-Secting-K-Means to K-Means is rather small and a hierarchy has advantages of its own, Bi-Secting-K-Means clustering is nevertheless worth considering for obtaining sibling groups to be presented to a user. Next we will investigate which parameters yield the best Bi-Secting-K-Means results.

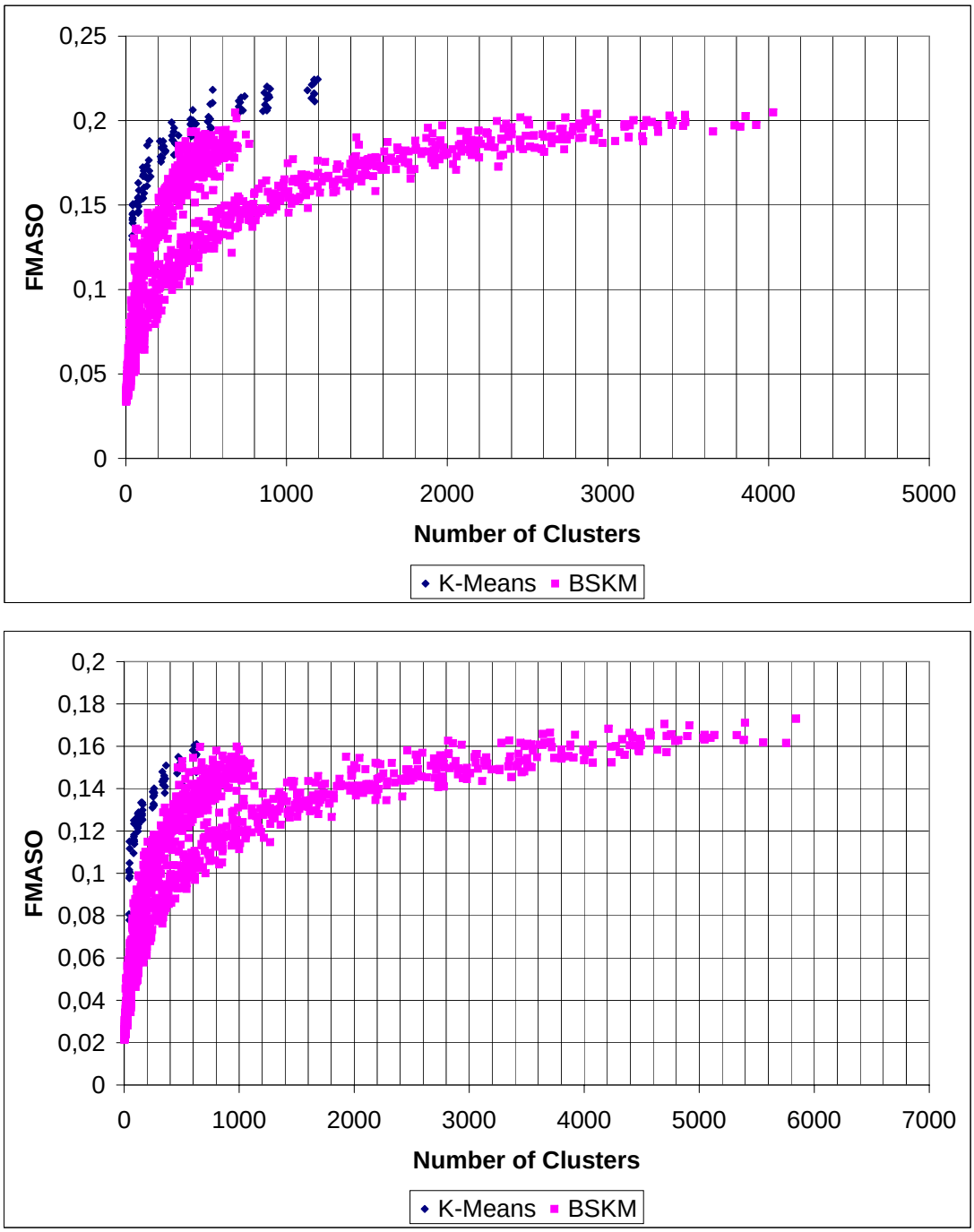


Figure 5.5: FMASO for different K and for K-Means clustering and Bi-Secting-K-Means clustering for (a) GSO1 and (b) GSO2

5.3.2 Experiment 2: Different Observation Strategies on the Cluster Hierarchy

In this experiment we will differentiate several ways of how subsets of the cluster hierarchy are obtained for evaluation runs because according to the FMASO two collections of sets are compared for an evaluation run. The hierarchical structure among the clusters has to be removed. We have performed experiments for the strategies "Up-To-Hierarchy-Level-L", "Complete-Cluster-Hierarchy" and "Separate-Hierarchy-Levels" described in section 5.2.

Figure 5.6 shows the results by differentiating the different ways of accessing the cluster hierarchies. The best results are obtained while using the clusters from a particular hierarchy level. But it has to be mentioned here that usually the hierarchy is used on a deep level and a lot of similar clusters are evaluated, but the evaluation criteria allows only for one best match. In a real world scenario, where a human would inspect the cluster hierarchy, the user could stop inspecting the cluster hierarchy if the clusters do not contribute anymore to the goal of finding meaningful sibling groups.

The finding of this experiment contradicts the idea of providing a hierarchy to the user. In the next experiment we will investigate on which hierarchy levels the best matches to the reference can be found.

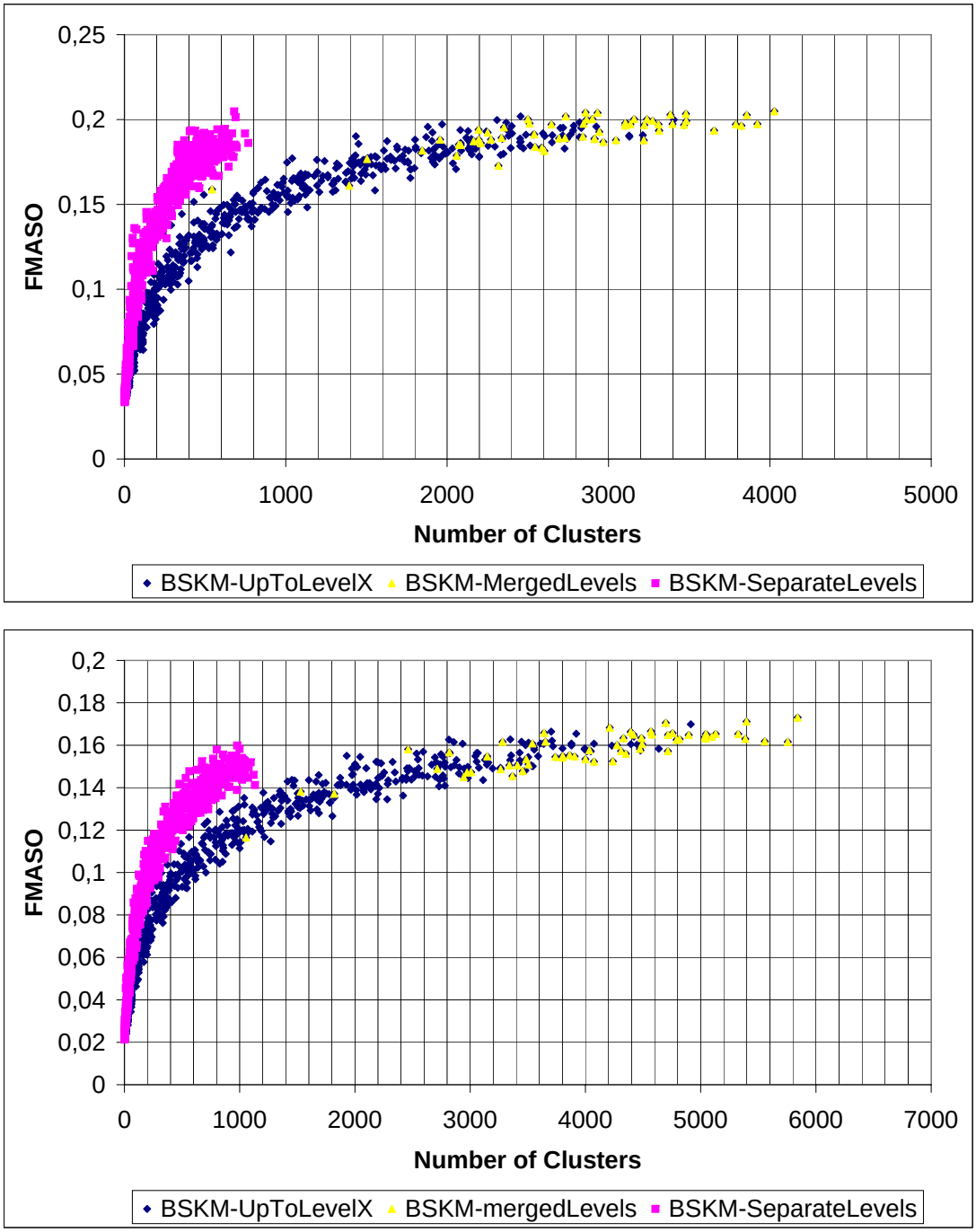
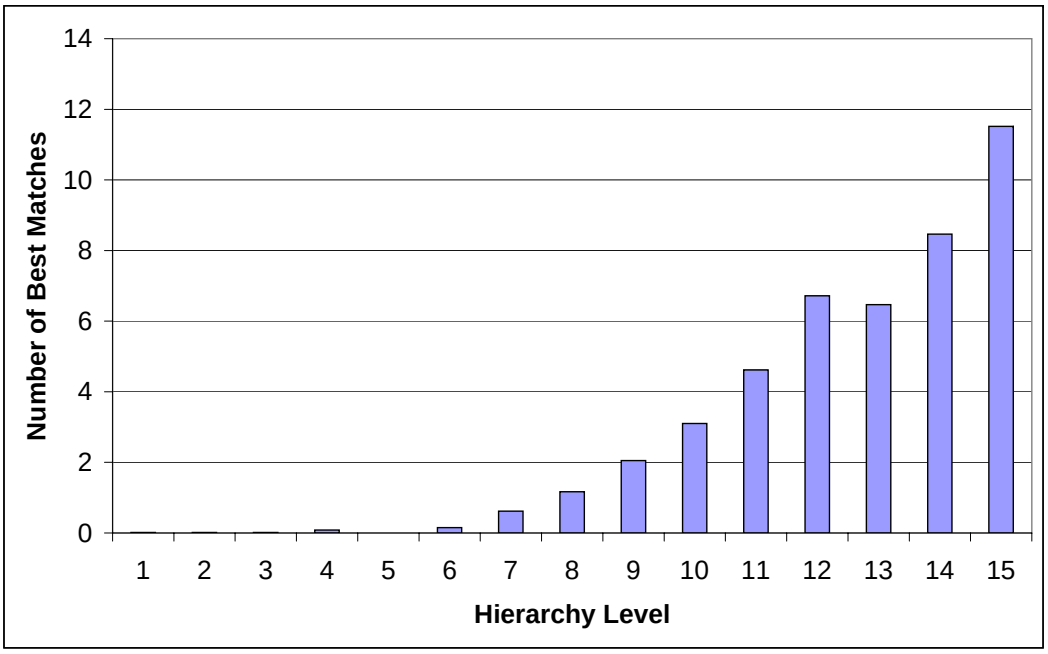


Figure 5.6: FMASO for Bi-Secting-K-Means clustering separated by different hierarchy observation strategies for (a) GSO1 and (b) GSO2

5.3.3 Experiment 3: Best Matching Hierarchy Levels

In the following experiment we will investigate on which hierarchy levels the best overlap with the reference sibling sets occurred. In figure 5.7, the distribution of the *best matching hierarchy level* is shown for GSO1 and GSO2. The results are averaged over 60 Bi-Secting-K-Means clusterings. The hierarchy is accessed in *breath first traversal* order. This corresponds to the way a human would access the hierarchy, as a human user would rather start from the root (level 0), than looking at a potentially very deep tree. As can be seen, the best matching hierarchy levels are distributed over several levels. From this we conclude that it is appropriate to present the hierarchy to the human ontology engineer. If the access to the hierarchy would be limited to only the levels near the root, many sibling groups which are evaluated as "good" would be missed. And it shows that a single level is also not appropriate if best matches are to be captured.

Conclusion: This experiment depicts the notion of recall whereas the good results of experiment 2 favouring the "Separate-Hierarchy-Levels" subsets also consider the amount of clusters to be inspected, incorporating a notion of precision.



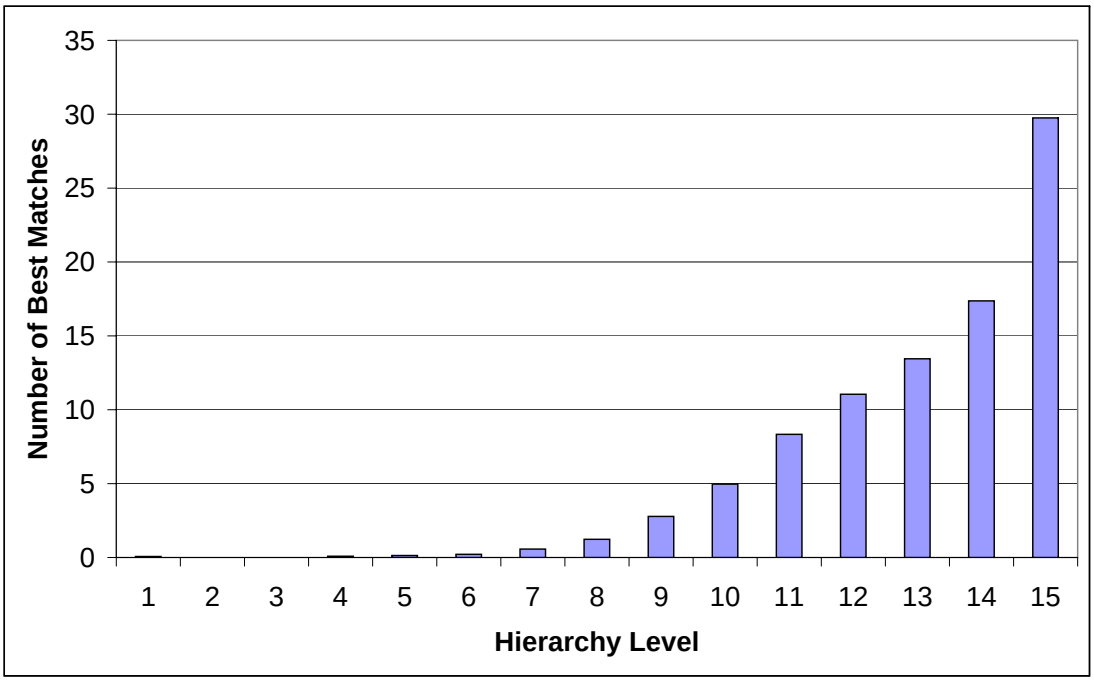


Figure 5.7: Best matching hierarchy level of Bi-Secting-K-Means for (a) GSO1 and (b) GSO2

5.4 Conclusion

Findings about Bi-Secting-K-Means from document clustering are not directly transferable to the clustering of different datasets, in this case text-span sets obtained by Group-By-Path.

On a dataset which is different from traditional text document vectorization, we investigated the quality obtained by using a flat K-Means clustering compared to using a hierarchical Bi-Secting-K-Means style clustering. We cannot state that the results obtained by Bi-Secting-K-Means clustering are in general as good as those obtained by a K-Means clustering, regarding a gold standard evaluation. But the results are not much worse. If the added value of the hierarchy can support the human ontology engineer on semi-automatic ontology learning, the slightly worse results obtained by Bi-Secting-K-Means clustering are acceptable. The best matching clusters of the generated sibling groups hierarchy can be found on various hierarchy levels, giving rise to the recommendation that the hierarchy should be presented to the user for subsequent human inspection.

6 Learning Sibling Pairs - XTREEM-SP

In the two previous chapters we have described procedures where sibling groups have been obtained by incorporating clustering algorithms. In this chapter we will perform association mining for finding sibling pairs. We refer to this approach as XTREEM-SP. It may be said that it consists of computing term associations upon a Group-By-Path dataset.

The computation of term associations is frequently performed within computational linguistics. There the notion of collocations [Smadja and McKeown, 1990] is used to extract terms/words which occur frequently together within a certain context. The "context" used for collocation computation is, for example, (1) direct neighbourhood ("Nachbarschaftskollokationen"), (2) sentence ("Satzkollokationen"), or other context windows such as a fixed range of words/terms. Those notions of context are due to the way of observing text as a "flat" sequence of words. By means of the Group-By-Path approach we are able to observe Web documents in a different way. We will compute associations on text span sets obtained by the Group-By-Path approach. By doing so we will extract term pairs where the relation is a sibling relation.

Perhaps the most essential argument for using association computation is the space and time complexity. In the last two chapters we have computed sibling groups by means of clustering. In this chapter we intend to do the computation of binary sibling relations. Binary sibling relations can be regarded as less valuable than sibling groups. But the computation is more space and time efficient. The core processing approach of the XTREEM-SP method, the computation of associations from a given co-occurrence matrix, is computationally less complex than the clustering of large datasets. The time complexity of K-Means is $O(nDK)$ where n is the number of instances, D is the number of dimensions, and K the number of clusters. The worst case time complexity of association computation from a co-occurrence representation is $O(D^2)$. The sorting (ranking of term pairs) of D^2 values ads, $D^2LN(D^2)$ yields a complexity of $O(D^2 + D^2LN(D^2)$. Actually, the sorting needs to be done only for sparse non-zero entries. For datasets with many instances and relatively low numbers of dimensions, association computation is more efficient than K-Means. This is the case for the datasets we have used in our experiments.

While performing clustering a dataset is partitioned into several groups. This is an advantage on the one side since the amount of generated groups can be controlled independent of the number of observed dimensions. On the other side, there is not

necessarily a cluster for each term where meaningful associations are observable. In the case of a feature space which is given by a manually crafted vocabulary as it is the case for our scenario where the vocabulary of an existing ontology was given as input, it can be assumed that suggestions for all terms are desired, regardless of the support since all terms are to be included finally. Recapitulating the findings of chapter 4, for tagpath clustering not all features have been observed in cluster labels and for term clustering there was a large cluster of terms which could not be clustered at all. In contrast, n-strongest related terms can be obtained for almost all terms after association scores have been computed. Here not only global patterns (tagpath clusters) but a rather local view of the strongest related terms for every term is assessed. Whether this is finally desirable depends on the objectives of the user. In general, he is likely to be interested in patterns which have a high support. But for terms where no patterns with high support can be observed, he might prefer to see patterns with low support rather than to see no patterns. The computation of associations can yield results of n-best related sibling terms even if the support would otherwise prevent the establishment of a cluster.

For binary sibling relations it is possible to compute precision and recall as we will do in our evaluation experiments in section 6.3. This raises the opportunity to obtain a well known evaluation criteria from Group-By-Path datasets.

6.1 XTREEM-SP Procedure

The XTREEM-SP procedure, depicted in figure 6.1, also comprises the first 3 processing steps of XTREEM-SG already described in chapter 4 (section 4.1.1, section 4.1.2 and section 4.1.3). Starting with a query, a Web document collection is retrieved. Upon the documents the Group-By-Path algorithm (chapter 3) is applied. This results in a collection of syntactically motivated sibling groups (text span sets). Then the filtering is performed and only text spans which are contained in the input vocabulary are kept. The two next steps, step 4 and step 5, are described in the following sections. In step 4 a co-occurrence statistic is created from the filtered text span sets which is then used to compute association strength scores in step 5. The hypothesis is that derived scores are supposed to be indicatory of semantic sibling relations. In the evaluation experiments we will investigate the extent to which the association scores are indicatory of semantic sibling relations.

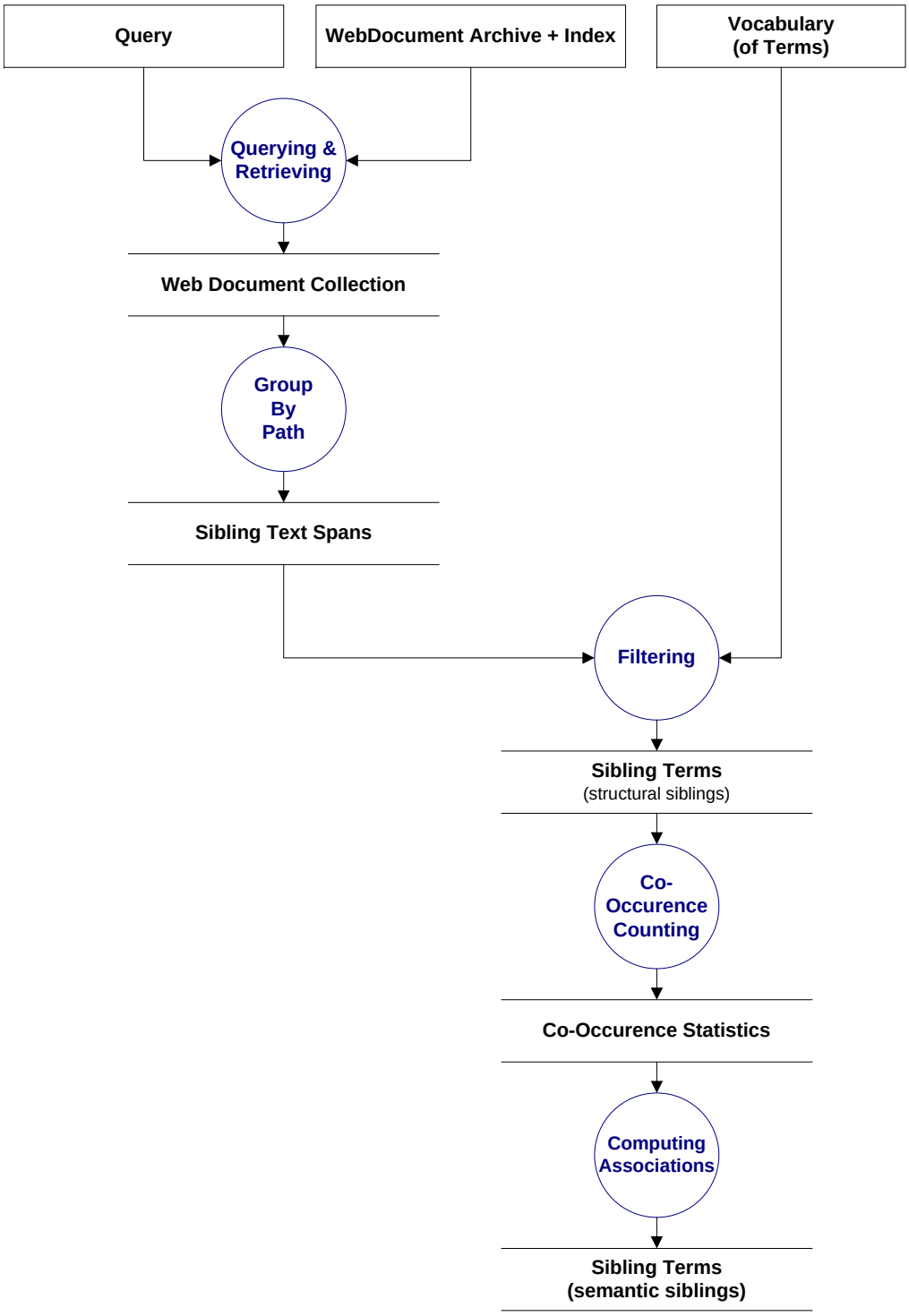


Figure 6.1: Dataflow diagram of the XTREEM-SP procedure

6.1.1 Step 4 - Co-Occurrence Counting

In this step a co-occurrence statistic is created. Recapitulating from chapter 4, after filtering a multiset ($\mathcal{A}_\mathcal{W}$) of term sets (B) was obtained. From those sets the co-occurrence counts are obtained for all pair wise occurrences of text spans $e_1 \in B_i \cap e_2 \in B_i$ for all $B_i \in F$. As a result a co-occurrence frequency for every term-term combination is obtained.

The notion of frequent "co-occurrence" is used here in a non-conventional sense: if two terms (text spans) e_1, e_2 co-occur for XTREEM-SP, then this does not imply that they are "co-located", that is in close proximity in sequential text. In fact, the identical paths that lead to them may be located in mutually remote parts of the document. However, these paths indicate that the two terms are used in similar contexts. This is a much stronger requirement than the arbitrary co-occurrence of two terms inside documents that may be large and heterogeneous.

6.1.2 Step 5 - Computing Association Scores

From the counts on term pair co-occurrence obtained in step 4, the strength of the association between the pair components can be inferred. Association measures can be used to obtain an association score for pairs of jointly occurring terms. Association measures are mathematical formulae which interpret co-occurrence frequency data. In computational linguistics, the joined occurrence of terms is referred to as collocations. Hence the association measures are also referred to as collocation measures. The automatic acquisition of collocations was first performed by Smadja and McKeown [Smadja and McKeown, 1990].

By means of association measures one can compute association scores for pairs of terms/words. The score gives an indicator about how strongly two terms/words are associated. Many association measures originate from statistics; they are based on statistical hypothesis tests (χ^2-association [Manning and Schütze, 1999]) while others are information theoretic founded (mutual information [Church and Hanks, 1989]) and yet others are heuristics such as the pure co-occurrence frequency, or the squared or cubic values of mutual information scores [Evert, 2005]. For a comprehensive overview of association measures see [Evert, 2005]. The association scores computed by different association measures cannot be compared directly. The exact association score is usually not of further interest, only the relative value of scores which results in a ranking into a list is used. The invocation of association measures originating from statistics does not imply that the association scores are compared to significance values. For example, if a χ^2-association score is derived, usually no comparison with statistical significance values is performed. The computed association score is used for comparison to other candidates and a ranking according to the association score.

There is no known best association measure; association measures compared to each other yielded no association measure which outperforms others. And

subsequently, there is no general recommendation as to which association measures should be invoked in this step. We use two association measures, a very simple one and a statistically founded one.

The first association measure which we consider is the co-occurrence frequency [Manning and Schütze, 1999, page 153]. Despite its simplicity, co-occurrence frequency is a viable choice for an association measure [Wermter and Hahn, 2006] and does not necessarily yield inferior results. We will use this as a straightforward baseline in our experiments too and denote this as "raw occurrence frequency". Furthermore, we apply the computation of association scores according to the χ^2-association measure [Manning and Schütze, 1999, page 169]. It is stated [Manning and Schütze, 1999, page 170] that the reason why χ^2 has been applied to a wider range of problems in collocation discovery is that he is also appropriate for large probabilities for which the normality assumption of the t-test fails. Its application is appropriate on sufficiently large datasets such as the ones obtained from big Web document collections used within our experiments.

The computation of binary association according to the χ^2-association measure is done as described in the following.

The χ^2-association measure is based on Pearson's χ^2-test [Plackett, 1983]. The χ^2-test can be applied to tables/populations of any size. He has a simpler form for 2-2 tables. Co-occurring terms depict such a simple case which can be represented by a 2-2 table. Table 6.1 shows a 2-2 contingency table. In this table the number of times two entities U and V occurred are represented. O_{11} depicts the number of joined occurrences of both entities, O_{12} and O_{21} the number of occurrence where only the one or the other entity occurred and O_{22} how many time neither entity occurred.

	V = v	V ≠ v
U = u	O_{11}	O_{12}
U ≠ u	O_{21}	O_{22}

Table 6.1: Observed frequencies within a 2-2 contingency table

From this co-occurrence observations the χ^2-association score is computed by the formula:

Definition 6.1 (Chi-squared Association Score- χ^2)

$$\textit{chi-squared association score} = \frac{(O_{11}+O_{22}+O_{12}+O_{21})(O_{11}O_{22}-O_{12}O_{21})^2}{(O_{11}+O_{12})(O_{11}+O_{21})(O_{12}+O_{22})(O_{21}+O_{22})} \tag{6.1}$$

As a result, an association score for every term-term combination is obtained. We denote this as co-occurrence statistic "sibling relations". It is possible to observe the terms which are (1) most related across among all observed term pairs, as well

as to focus on the (2) terms which are most related to a term. For the evaluation experiments we pursue the first variant; for other purposes, for example, where the results are displayed as lists of ranked terms, the second variant appears to be more appropriate. Both of those possible usage scenarios involved sorting values of a (sparse) matrix or vector.

6.2 Evaluation Methodology

In contrast to the evaluation of sibling groups performed for the evaluation of XTREEM-SG in chapter 4, described in section 4.2, it is possible to use precision and recall to determine the quality of binary sibling relations. As evaluation reference we use (again) the two gold standard ontologies (GSO) described in section 8.3.2. From the GSO's also sibling relations can be extracted. They also provide the closed vocabulary whereupon sibling relations are automatically derived by the XTREEM-SP procedures.

In experiment 1 we will contrast the sibling relations obtained with XTREEM-SP against the results obtained on the traditional Bag-Of-Words vector space model, and a further alternative method based on markup. In experiment 2 we examine the influence of an association measure and compare it to the use of co-occurrence frequency alone. In experiment 3 we will investigate the influence of the input query which constitutes the Web document collection to be processed. In experiment 4 we will vary the minimum support of terms within the Web document collection to be processed.

6.2.1 Evaluation Criteria: Precision and Recall

From the gold standard ontologies, we extract all concept pairs which stand in a sibling relation. Those relations are treated as "reference". The object of the evaluation is a ranked list of automatically obtained concepts pairs, whereas the ranking is given according to the association strength of the concept pair. For each automatically obtained concept pair, we determined if this relation is also supported by the reference which gives a positive count. If a concept pair is not supported by the reference a negative count is assumed. By doing so for each position in the ranked list, recall and precision can be computed.

The recall is the ratio of the number of observed true sibling pairs ($\#positive$) to the number of sibling pairs given by the reference ($\#overall$).

Definition 6.2 (Recall)

$$recall = \frac{\#positive}{\#overall} \tag{6.2}$$

The precision is the ratio of the number of true sibling pairs ($\#positive$) to the number of observed automatically generated pairs ($\#positive + \#negative$).

Definition 6.3 (Precision)

$$precision = \frac{\#positive}{\#positive + \#negative} \tag{6.3}$$

For a ranked list of associated term pairs a recall precision chart line can be obtained by a series of measurements on recall precision values.

6.2.2 Evaluation Reference

We use the same gold standard ontologies from the tourism domain we used for the evaluation of XTREEM-SG described in section 8.3.2. From the ontologies we extract pairs of terms denoting concepts standing in sibling relation. They form our reference set of sibling pairs. From GSO1 with its 293 concepts grouped into 45 sibling sets, we obtained 1176 sibling pairs. From GSO2 constituted by 693 concepts and 90 sibling sets, 4926 sibling pairs have been extracted.

6.2.3 Inputs

Archive+Index: We use the same Web crawl as already used for the evaluation of XTREEM-SG. 9.5 million Web documents in English language have been obtained by a topic focused Web crawl on the "tourism" domain. The documents are indexed, so that for a given query a Web document collection can be retrieved.

Queries: For our experiments we consider four document collections which result from querying the Archive+Index. We used the three queries already used for the evaluation of XTREEM-SG, the queries $Query1$ - "touris*", $Query2$ - "accommodation" and by the whole topic focused Web document collection reflected by $Query3$ - "*". Additionally we give the results for $Query4$ ("accomodation"). $Query4$ was foremost a misspelling on $Query2$ ("accommodation".), but since this variant is present in millions of Web documents we will present these results. Those variations are the subject of experiment 3.

Vocabulary: From GSO1 and GSO2 we used the 293 and 693 terms which label the concepts as vocabulary.

6.2.4 Variations on Procedure and Parameters

Document Representation Method: See section 4.2.4. The variation of these influences is the subject of experiment 1.

Association Measure: From the raw sibling sets obtained by accessing the document, the co-occurrence frequency of term pairs is counted. This frequency can be used as the indicator of association strength. We will refer to this method by the

term "frequency". With the χ^2-association measure [Manning and Schütze, 1999], more statistically stable values of association strength can be computed. The variation of these influences is the subject of experiment 1 and experiment 2.

Minimum Feature Support Threshold: See section 4.2.4. The variation of these influences is the subject of experiment 4.

6.3 Experiments

In the following sections we show the results obtained from the experiments. Table 6.2 shows the number of documents which adhere to a certain query. This corresponds to the size of the Web document collection which is processed by the subsequent processing steps. Table 6.2 also shows the number of candidate sibling sets obtained after performing the processing on different queries for the two vocabularies. Only terms which are present in the input vocabulary are observed. Table 6.2 also shows the number of observed pairs derived from these sets.

Table 6.2: Numbers characterising the used data sets

Document Collection	Query Phrase	Number of Documents	Number of Sibling Term Sets		Number of Sibling Term Pairs	
			GSO1	GSO1	GSO1	GSO1
1	"touris*"	1,468,279	222,037	318,009	1,600,440	3,804,214
2	"accommodation"	1,612,108	293,225	373,802	2,092,432	3,885,532
3	"*"	9,437,703	924,045	1,326,843	5,763,596	14,071,016
4	"accomodation"	471,540	78,289	98,886	686,108	1,198,224

6.3.1 Experiment 1: Sibling Relations from Group-By-Path in contrast to alternative Methods

In this experiment we contrast the quality of results on finding sibling relations obtained with the Group-By-Path based XTREEM-SP with the Bag-Of-Words vector space model and with a method based on MarkUp without path information. Web document collection1 ($Query1$: "touris*") was chosen as the Web document collection to be processed. The comparison was performed for two methods on association strength (frequency, χ^2) and for both references (GSO1,GSO2).

The diagrams which result on the usage of "frequency" as association strength indicator (figure 6.2) show that Group-By-Path performs best for both GSO's. MarkUp performs better than Bag-Of-Words. The overall measured result quality is relatively low. The top ranked association pairs Group-By-Path (and MarkUp) yield a high precision which then rapidly declines. For higher recall values the chart lines converge. Since a recall above 40 percent is only obtained on Bag-Of-Words, we can conclude that some terms never occur as siblings on tagpaths. This does not necessarily mean that Group-By-Path is weak; since the ontologies do not directly

encode sibling relations, there may exist concepts which tend not to occur together. For example, "ski school" and "surf school" may be subconcepts of "sport school", but are rather unlikely to be discovered as siblings directly from Web documents. The evaluation criteria cannot prevent from such cases.

Figure 6.3 shows the results when the association strength was computed by the χ^2-association measure. In contrast to the usage of raw co-occurrence frequency based association strength, the results of MarkUp are nearly the same as for Group-By-Path. An explanation for this is that the χ^2-association measure here performs well on diminishing sporadic occurrences which can happen on MarkUp in comparison to Group-By-Path. Bag-Of-Words performs in a worse manner here too. A possible explanation why MarkUp yields better results than Group-By-Path is that the experiments within this chapter are performed on a closed vocabulary which is rather limited in size. The choice of pairs observed in the documents is, therefore, drastically limited in comparison to using an open vocabulary (as for example done in chapter **??**). MarkUp captures all terms captured by Group-By-Path and all other terms of the vocabulary which correspond to a text span. Since this will occur relatively rarely, because of the rather small vocabulary, there is a high correlation between term sets captured by MarkUp and Group-By-Path. But the bigger the vocabularies used as feature space are, the more MarkUp mixes together terms which are not siblings while Group-By-Path is capable of separating them according to tagpaths. When using an open vocabulary the orientation of associations generated with Group-By-Path towards sibling relations, in comparison to MarkUp, becomes more visible than measured on the limited vocabulary.

Conclusion: Our experiments on automatically obtaining sibling relations showed that the XTREEM-SP procedure with its Group-By-Path approach, shows the best results. Though it was not claimed that the Bag-Of-Words model is strong on capturing sibling relations, we can confirm the hypothesis that the results obtained with XTREEM-SP are motivated by sibling relations.

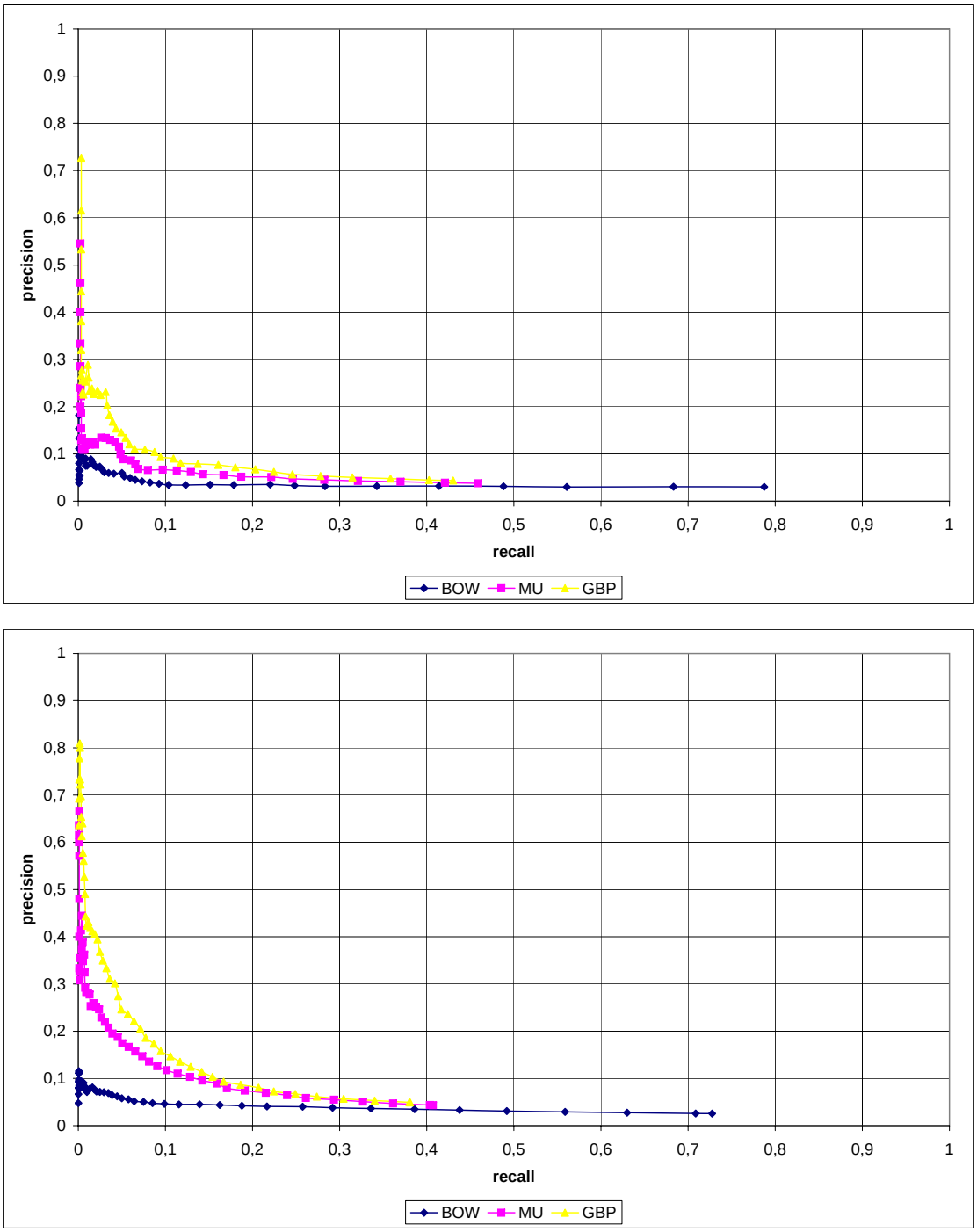


Figure 6.2: Precision and recall for different document representation methods (frequency, Web document collection 1) for (a) GSO1 and (b) GSO2

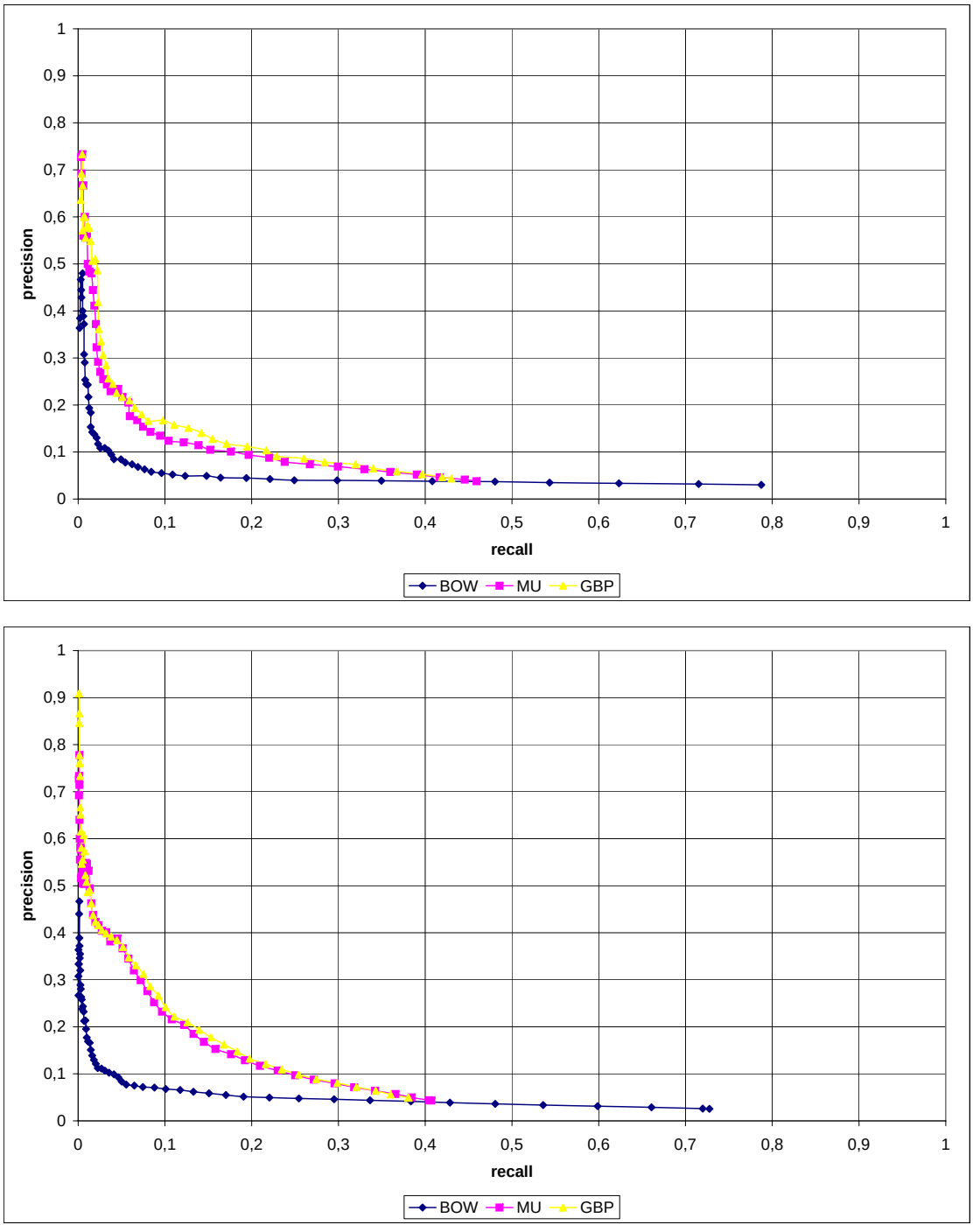


Figure 6.3: Precision and recall for different document representation methods (χ^2, Web document collection 1) for (a) GSO1 and (b) GSO2

6.3.2 Experiment 2: Association Measures in Comparison

In this experiment we will investigate the influence of the method which is used to obtain association strength. Specifically we will use the raw co-occurrence frequency and the χ^2-association measure [Manning and Schütze, 1999]. In experiment 1 for the different association strength methods this was done sequentially; in contrast, figure 6.4 shows the chart lines on Group-By-Path of figure 6.2 and figure 6.3 together in one chart.

Figure 6.4 shows that on both vocabularies/references the usage of χ^2-association strength yielded the best results. We also used mutual information [Church and Hanks, 1989] and poison association measure [Quasthoff and Wolff, 2002] as well as cosine distance. Those results are not remarkable enough to be presented separately; the results are comparable to χ^2-association or worse, but better than just frequency. The literature on the quality of these association measures mentions that different association measures perform sometimes better, sometimes worse than others, with no clear conclusions. Therefore, we did no further investigations about which alternative association measures perform better than frequency and χ^2-association since this might change on another domain.

Conclusion: In the experiments of this thesis for obtaining sibling pairs from a closed vocabulary, the χ^2-association measure gave the best results compared to the frequency based association strength.

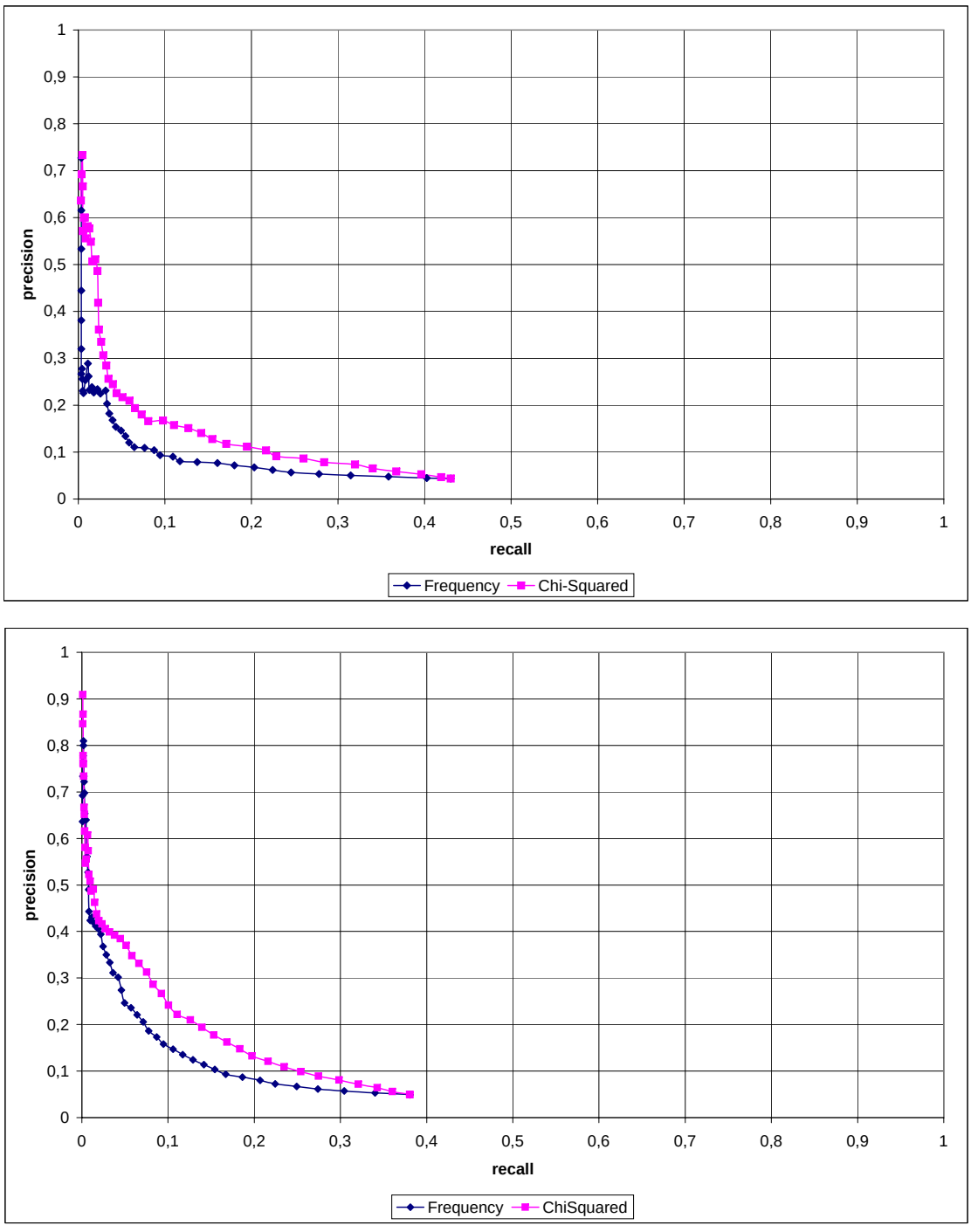


Figure 6.4: Precision and recall for frequency and χ^2 association strength (GBP, Web document collection 1) for (a) GSO1 and (b) GSO2

6.3.3 Experiment 3: Varying the Topic Bias

XTREEM-SP relies on constituting a Web document collection by a query. A query, therefore, represents the focus (topical bias) of the analyzed data. Here we will investigate how variations on the query influence the obtained results on sibling relations. The different queries are shown in table 6.2.

As the first diagram of figure 6.5 shows, the results of all 4 queries are closely together for GSO1. For GSO2 (second diagram of figure 6.5), the results vary more than for GSO1. For both GSO's, *Query*3 - "`*`", which depicts the entire focused Web document crawl, yielded the best results. An explanation for this is that with the single phrase queries (*Query*1, *Query*2 and *Query*4) a rather too narrow focused Web document collection is processed. The reference contains terms and relations which are not present on Web documents adhering to a certain "focused" query. This means that for practical settings a combined query (for example, "`tourism* OR accommodation OR holidays OR 'sport event' ...`") may be the better choice. Such a broader query prevents a too narrow focus which might yield an undesired bias. For example, while using only "`tourism*` as query, documents which are about tourism as an economic field and tourism as a political field are captured as well. On bigger or open vocabularies sibling relations from those domains are likely to be captured as well. While using queries of several terms, the domain is encircled in a more balanced way, while only documents with more than one matching term are used. But such observations have to be treated with caution since, on the other hand, an ontology engineer will likely focus on a fraction of the conceptualization to be obtained or improved at one moment and, therefore, focused queries are appropriate.

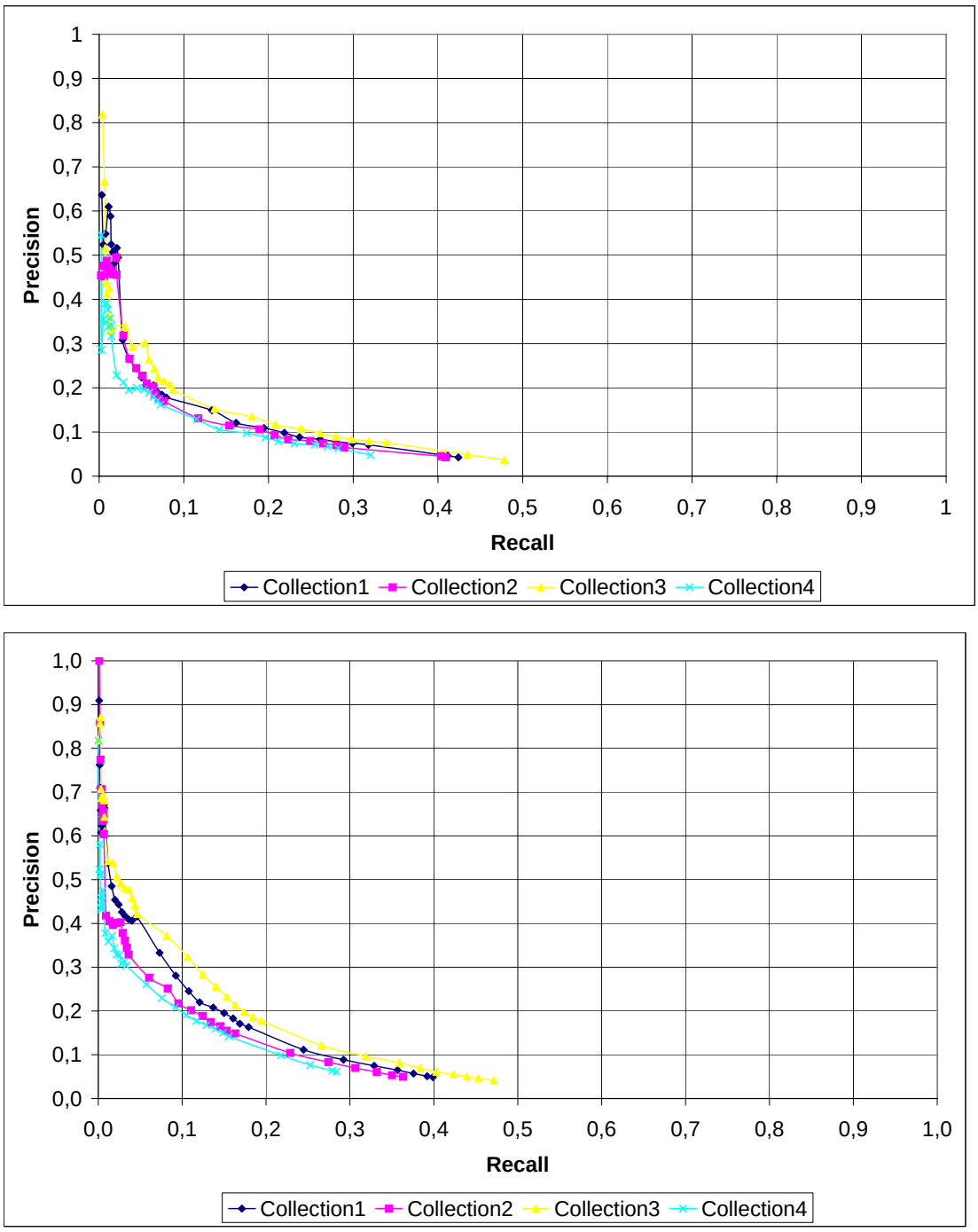


Figure 6.5: Precision and recall for different queries (GBP, χ^2) for (a) GSO1 and (b) GSO2

6.3.4 Experiment 4: Variations on the Minimum Support

In the last experiment we will investigate the influence of the term occurrence frequency in the Web document collection on the obtained results. As a side effect of an increased minimum support, errors or terms representing abstract concepts, errors and foreign language terms present in the reference ontologies, is dimmed. Furthermore, though relatively big, some terms are not present in many documents. This can be due to the fact that the topical bias of the Web crawl was not optimal and also because the terms are rare, even on the entire Web. While using only terms with higher support, terms for which not so many observations are possible are ignored. With increasing minimum support more and more sibling relations are omitted by eliminating these pairs from the reference. Table 6.3 shows the decreasing number of relations by increased minimum term occurrence support. We used the support of terms, not of the co-occurrence of term pairs which would be an alternative approach. As figure 6.6 shows, for increased minimum support, better results regarding recall and precision are obtained. This means that sibling relations of high frequent terms are found better than those in the case of less frequent ones. Terms with lower support are not found in sibling constellations regularly enough to reveal plausible sibling relations to the same extent as the terms with higher support. The occurrence used for support was not restricted to termsets found by Group-By-Path but the terms could appear anywhere in the web documents. We do so for not favouring Group-By-Path since it might be the fault of Group-By-Path not to find terms within sibling termsets.

Table 6.3: Decreasing number of reference sibling relations on increased support

Required support		0	1	10	100	1000	10000	100000
Number of reference sibling relations	GSO1	1176	1120	1033	844	637	404	161
	GSO2	4926	4553	4073	3439	2653	1006	582

Conclusion: Our experiments showed that sibling relations for terms with a high occurrence frequency are found better than sibling relations for terms with lower occurrence frequencies. By doing so we could circumvent problematic lexicalisations present in the reference ontologies which are practically not or only rarely supported by English Web documents. But this observation also leads to the recommendation that, if possible, the Web crawls should include many documents also for the not most frequent terms. It might be interesting to create procedures which obtain additional Web documents for terms with a low occurrence frequency in the already available Web document collection. By doing so the Web document collection would get biased. It bears the chance that the results of rare terms become better chances to reveal sibling relations without increasing the size of the processed data by orders of magnitude.

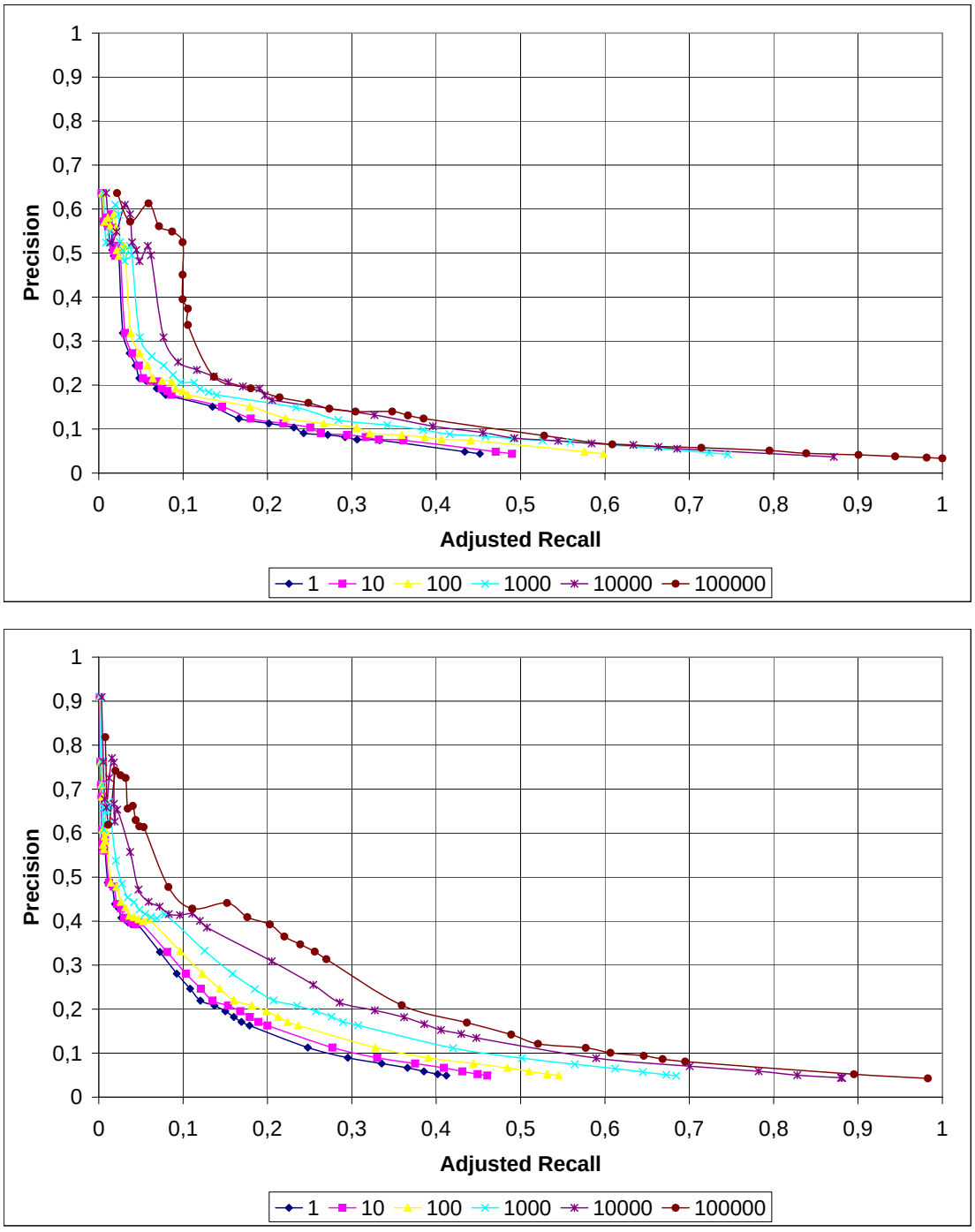


Figure 6.6: Precision and recall for different frequency support levels (Web document collection 1,GBP, χ^2) for (a) GSO1 and (b) GSO2

6.4 Conclusion

We have presented XTREEM-SP, a method for the discovery of semantic sibling term pairs. In our evaluation against gold standard ontologies, we could confirm that Group-By-Path data sets processed by means of χ^2-association measures are able to find terms standing in a semantic sibling relation.

7 Vocabulary Extraction with XTREEM-T

In the previous chapters we used Web documents to find sibling relations. In this chapter we investigate if those Web documents can also be used to find domain specific vocabularies of terms, independent of finding sibling relations. In the context of this thesis, the experiments of this chapter show to which extent frequent text spans correspond to terms of the domain vocabulary. Such a vocabulary of terms can be used for other purposes where vocabularies are beneficial. Up to date domain specific vocabularies are valuable resources [Rinaldi et al., 2005], especially for application fields ontology learning or text-mining. Usually, such domain specific resources are not available; however, manually crafting the vocabulary without support of suited tools is neither feasible nor advisable. Though the acquisition of terms is the basic layer in ontology learning, it is important since the subsequent layers rely on the vocabulary. In ontologies, terms depict the labels of ontological entities such as concepts, instances or relations. In computational linguistics, terms form the vocabulary of a domain [Mitkov, 2003]. According to [Bourigault and Jacquemin, 1999], terms correspond to sequences of words, most of the time noun phrases, which are "terminological units". But other types of terms such as verbs and adjectives also are terms which should be extracted since they can be found as well in the lexical layer of ontologies.

Since terms can contain whitespace, one can distinguish single word terms and multiword terms. Single word terms are terms such as "ocean" and "water". They do not contain whitespace. The sequence of words "Atlantic ocean" is a multiword term expression as it includes whitespace. Multiword terms, in a similar notion are also referred to as multiword expressions [Sag et al., 2002, Dowdall et al., 2004]. According to Jackendoff [Jackendoff, 1997, page 156], it is estimated that the number of multiword expressions in a speaker's lexicon is of the same order of magnitude as the number of single words. Jackendoff also notes that this might be even an underestimate, since, for example, 41 percent of the entries in WordNet 1.7 [Fellbaum, 1998] are multiword expressions and that specialized domain vocabularies overwhelmingly consist of multiword expressions. For the English language, multiword expressions depict a crucial fraction of domain vocabularies. For languages like German, where compounds are heavily used (for example "Tigerhai" for the English "tiger shark"), detecting multiword expressions is less important, but still relevant since there is still a fraction of terms which consist of several words. Consequently, we assume that multiword terms are also important for the lexical layer of ontologies. But despite the importance

of multi-terms and the circumstance that, for humans, there is usually no crisp distinction between single and multiword terms, there are, according to Zhang [Ziqi Zhang and Ciravegna, 2008], only 5 approaches are capable of acquiring single word and multiword expressions at the same time. In this thesis we will show an approach that uses Web documents to obtain terms without the necessity of incorporating training or language or domain specific software. The acquired terms include both single word terms and multiword terms.

Acquiring a vocabulary automatically from textual content is the subject of term extraction, also referred to as term acquisition. Even after decades of research, acquiring terms is not trivial because the approaches are usually domain and language dependent and require training. The adoption of existing methods to a new domain is laborious. Within Web documents, some "sequences of text" (text spans) occur frequently "marked-up" by tags. We will show that ordering the text spans according to their occurrence frequency in the Web document collection separates promising term candidates from ordinary text spans. That makes this direct approach transparent. There are no parameters and heuristics which prevent the practical applicability to other domains and languages. In contrast, the XTREEM-T approach does not rely on natural language processing resources auch as rules and other background knowledge. XTREEM-T is domain and language neutral and operates on easily obtainable Web documents. The difficulties in adopting term acquisition methods to new domains is perhaps the drawback that exacerbated the broad incorporation in application areas like ontology learning and text mining.

In the field of ontology learning there are, for example, the approaches of [Velardi et al., 2001b, Velardi et al., 2001a, Moigno et al., 2002, Gillam and Tariq, 2004, Mariam et al., 2005] which tackle the terminology acquisition step in a non trivial way. But often ontology learning is performed while only trivial term acquisition approaches are incorporated. If the task of acquiring multiword terms is omitted in ontology learning procedures and no vocabulary containing terms is given as input, the learned concepts and relations have only trivial labels of single words and it is left to the ontology engineer to correct this manually. But even worse, relations between ontology entities labelled with multiword terms are likely to be missed. Since the overall aim of performing ontology learning is to reduce the per entry cost, it is an important goal to acquire and process vocabularies which include multiword terms. All approaches presented in this thesis are capable of handling multiword terms. They can actually handle multiword terms in the same manner as single word terms; no separate processing is necessary.

7.1 Related Work

For an overview on terminology acquisition see [Jacquemin and Bourigault, 2003, Witschel, 2005, Deane, 2005, Ziqi Zhang and Ciravegna, 2008]. There

are two major types of approaches for terminology acquisition (1) approaches relying on syntactic chunks invoking linguistic parsing [Piao et al., 2003] and (2) approaches relying on statistics [Damerau, 1993, Frantzi et al., 2000, Pantel and Lin, 2001, Nakagawa, 2001, A. Ballester, 2002, Nakagawa and Mori, 2003, Wermter and Hahn, 2005b, Chen et al., 2006]. Dias et al [Dias et al., 2000] presents a combination of both types. With approaches relying on parser generated syntactic chunks, XTREEM-T shares "finding boundaries on term expressions"; with statistical approaches XTREEM-T shares the incorporation of large amounts of documents.

There are many methods such as [Bodenreider et al., 2002, Xiao and Rösner, 2004, Wermter and Hahn, 2005a, Baneyx et al., 2005] which are focused on documents from the biomedical domain, where special, high quality text corpora are used. But those approaches designed for rather pure text are not generally applicable. The approaches designed for high quality text are likely to be hampered with the noise present in Web documents, since conversion cannot be expected to be perfect. And the text obtained from Web documents can be expected to be different from traditional plain text documents in general. The navigational elements of Web documents make the task even more different from pure text methods. The conversion from semi-structured text to pure plain text also eliminates information. We regard this information as valuable not only for interpretation by the browser rendering the Web content but also for term acquisition. In [Kruschwitz, 2001b] the markup of Web documents is used to learn a domain model. There the boundaries created by Web document structure are also used but not for directly obtaining terms but as a more broad context in which terms are observed.

7.2 XTREEM-T Procedure

The dataflow diagram depicted in figure 7.1 gives an overview of the XTREEM-T approach for obtaining a domain specific vocabulary (including multiword terms) from Web document collections.

In the following the individual steps of the XTREEM-T procedure are described.

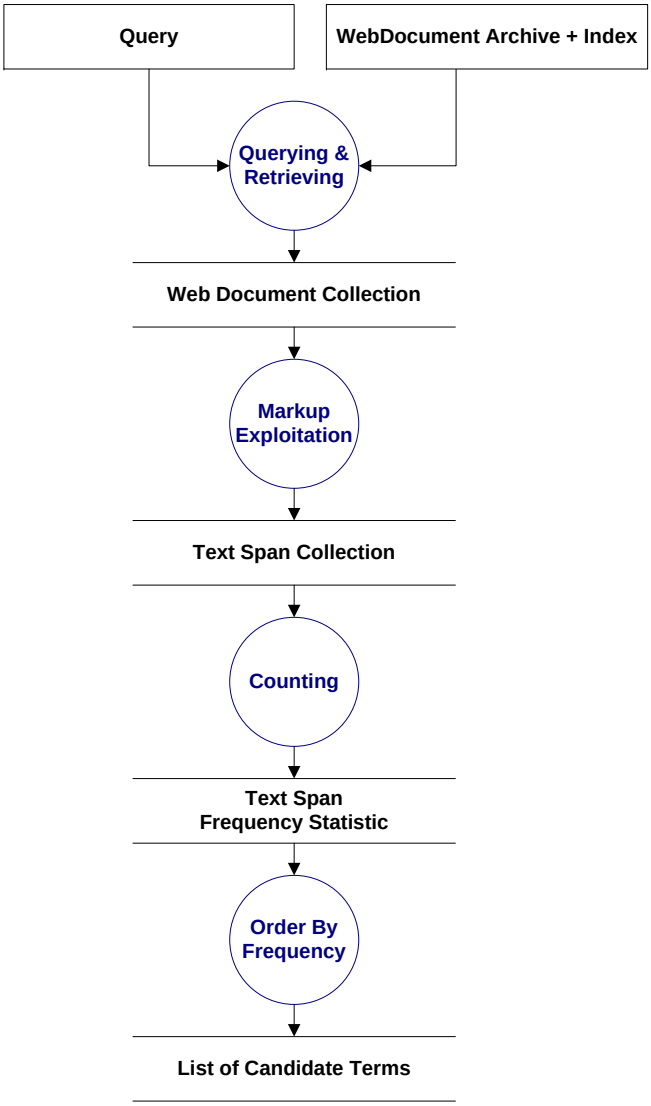


Figure 7.1: Dataflow diagram of the XTREEM-T procedure

7.2.1 Step 1 - Querying & Retrieving:

The availability of a corpus is normally a prerequisite. For some domains it is possible to get big document collections (for example, Medline); for other topics assembling a feasible document collection is a problem on its own. Since the Web is a big source of content for nearly all topics one can think of, using the Web seems to be an alternative also for a field like terminology acquisition, especially when the document collection is not itself of interest, as it is often the case for ontology learning.

The XTREEM-T procedure operates on medium size (thousands of documents) and large size (millions of documents) Web document collections. Such a Web document collection is obtained by querying an Archive+Index on a query. The Archive+Index is a large collection of Web documents, obtained by Web crawling, whereupon an index is created. The query constitutes the domain of interest whereupon semantics should be discovered. It should, therefore, encircle the documents which are supposed to entail domain relevant content, for example, "`touris*`" or "`myocardial infarction`".

The Web document collection should be big enough to contain manifold occurrences of the desired concepts. This is not supposed to be a small manually handcrafted document collection; bigger amounts of Web content which have an appropriate coverage of the domain are more desirable. To obtain such a comprehensive Web document collection, alternatively, a focused Web crawl can be performed. The Archive+Index can be easily replaced by obtaining Web document references from the public search engine API's of the major Web search engines.

7.2.2 Step 2 - Markup Exploitation:

Term extraction is based on finding boundaries which separate promising candidates from not relevant sequences of tokens. Our approach uses the boundaries available in Web documents. Those boundaries are mostly manually created by millions of Web content authors. These boundaries are explicit through the markup in semi-structured Web documents. Though the markup is usually not created to make term boundaries explicit, large amounts of such markup boundaries can be helpful for terminology acquisition.

Web content marked up with HTML tags contains textual data such has "`Here is <EM>marked up text</EM>`". The angle bracket limited tags, enclose sequences of text. The Web document can be interpreted as a collection of alternating sequences of textual data and markup sections (tags). We will refer to the textual sequences of text which are not markup as text spans. But text spans are not only created by directly enclosing a span of text but also between tags. Only text spans are of further interest for XTREEM-T, not the markup. In a tree representation of Web documents as used for describing Group-By-Path in chapter 3 those text spans are the textual content represented by text nodes. But for XTREEM-T,

the tree structure is not used at all, but only the text spans created by markup boundaries.

In figure 3.1, we have shown an exemplary fraction of (X)HTML Web content. Figure 7.2 depicts a list of text spans obtained from the content of the web document depicted in figure 3.1 by chunking on the markup tags. When considering bigger amounts of Web content treated in this manner, we postulate that promising term expressions are more frequent and can, therefore, be obtained from a frequency statistic. For example, only the text spans in line 2 to 6 are likely to be frequent among large amounts of text spans from a Web document collection. In contrast, the long text span of line 1 is unlikely to be frequent. The text spans are constituted by different numbers of tokens and can also be seen as word n-grams of variable length.

1:	`...In the following section you can find a description of oceans, the`
2:	`Atlantic Ocean`
3:	`the`
4:	`Pacific Ocean`
5:	`the`
6:	`Indian Ocean`
7:	`...`

Figure 7.2: List of text spans derived from HTML Web document

Multiple occurrences of whitespace in text spans are normalized to a single whitespace character; leading and trailing whitespace of text spans is removed. Additionally, one may perform a further cleaning of the text spans. For our experiments we used only alphabetic characters and eliminated punctuation and numbers, and all characters had been converted to lower case.

7.2.3 Step 3 - Text span Counting:

For the text spans, obtained from the Web documents in step 3, an occurrence frequency statistics is created. The occurrence frequency statistics represents the number of times a text span was observed. This frequency statistics contains text spans constituted by different numbers of whitespace separated tokens mixed together. For practical settings it is feasible to limit the length of the text spans to 5 to 10 tokens. This reduces the amount of data which has to be stored in the frequency statistic. The number of terms longer than 5 to 10 can be expected to be rather small.

7.2.4 Step 4 - Order By Frequency:

From the text span frequency statistics a list of candidate term expressions is generated by ordering text spans according to their frequency within the Web

document collection. According to our hypothesis, the top ranked text spans are term expressions. No stop word removal is required.

7.3 Evaluation Methodology

The term acquisition research field lacks an agreed evaluation vocabulary. Even if such an evaluation vocabulary will be made available, it would likely be bound to a specific document collection. For large Web document collection it is not feasible to manually assemble a gold standard which can then be used as a reference.

Because of the deficiency of gold standard vocabulary we will perform an exemplary manual evaluation on samples of term candidates. In our experiments we vary (1) the domain (topic) of the vocabulary, (2) the size of the document collection and (3) the rank-range where evaluation is performed.

7.3.1 Evaluation Criteria: Precision

The human evaluator was asked to accept or reject the presented terms. If the evaluator was in doubt about a term candidate, it was regarded as rejected, to be on the safe side.

Definition 7.1 (Precision)

$$precision = \frac{\#accepted}{\#accepted + \#rejected} \tag{7.1}$$

The *precision* is the relative number of accepted candidates to the overall number of candidates evaluated.

7.3.2 Inputs

Table 7.1 gives an overview of the experimental settings used for the evaluation. For all queries the XTREEM-T procedure was run. We retrieved 5 document collections.

The rather small Web document collections 1 and 3 are obtained by querying the Google Web search service[1]. The other document collections 2, 4 and 5 are obtained by performing large domain focused Web crawls, ranging from several hundred thousands (document collection 4) to 10 million documents (document collection 2 and 5).

The processing was limited to term expressions of up to a length of 4 tokens (unigrams, bigrams, trigrams and quadgrams) since this is usually the upper limit used for term acquisition. For inspection, subsets of the most frequent text spans have been selected as term candidates. The attempt was to determine whether

[1] http://code.google.com/apis/soapsearch/index.html

the term candidate can be regarded as a valid expression in the context of the corresponding domain.

For the 5 document collections, the top 1000 most frequent text spans have been evaluated. Additionally, for document collection 5 also the text spans with rank 10001 to 11000 and 50001 to 51000 have been inspected to investigate the decrease of quality in low ranks.

Table 7.1: Domains reflected by query phrases and the resulting number of Web documents used for the experiments

Document Collection	Domain	Query Phrase	Number of Documents
1	Ontology, Ontologies, Semantic Web	ontology OR ontologies OR "semantic web"[2]	3,974
2	Ontology, Ontologies	"ontolog*"	272,588
3	Myocardial Infarction	"acute myocardial syndrome" OR "myocardial syndrome" OR "acute myocardial" OR "myocardial infarction" OR "acute myocardial infarction"	1,037
4	Myocardial Infarction	"myocardial*"	42,768
5	Tourism	"accommodation"	1,612,108

7.4 Experiments

7.4.1 Experiment 1: Human Vocabulary Evaluation

In figure 7.3 and figure 7.4 we see a sample list of obtained term candidates. Those lists of terms have been evaluated resulting in the numbers shown Table 7.2 shows the precision obtained in the evaluation.

```
..., software, conferences, index, daml_oil, phone, site_map,
registration, tutorials, table_of_contents, figure, about_us, help,
conclusion, call_for_papers, services, artificial_intelligence,
program, at, university, main, see, project, education, java,
am, ieee_intelligent_systems, pm, topic_maps, more, price, pages,
see_also, archives, background, privacy_policy, download_now,
feedback, tools, ontoweb, iswc, applications, availability,
daml, uml, trackback, summary, technology, information_retrieval,
knowledge_representation, dublin_core, books, platforms, ...
```

Figure 7.3: Exemplary list of obtained term expressions from document collection 1 ("ontology", "ontologies", "semantic Web") ; rank 80 to rank 132

```
..., car_rentals, india, japan, hong_kong, paris, faqs, about,
information, malaysia, sweden, wales, price, denmark, fishing,
bahamas, keywords, bed_and_breakfast, czech_republic, norway,
new, directions, caribbean, croatia, weddings, website,
south_america, finland, advertise_with_us, check_in_date, hawaii,
country, indonesia, brazil, malta, resources, back_to_top, in,
amenities, self_catering, hostels, day, sydney, uk, jamaica,
other, forums, luxembourg, poland, homepage, florida, barbados,
general_information, transport, by, prices, bulgaria, currency,
travel_tools, pm, costa_rica, egypt, north_america, argentina,
meetings_events, back, russia, check_out_date, travel_guide, rome,
cars, specials, tel_fax, morocco, vacation_packages, victoria,
photos, more_info, iceland, sports, apartment, vietnam, deutsch,
directory, philippines, jobs, san_francisco, single, barcelona,
edinburgh, ...
```

Figure 7.4: Exemplary list of obtained term expressions from document collection 5 ("tourism"); rank 161 to rank 251

The first 5 rows show the obtained results in the top 1000 most frequent text spans, evaluated whether they are domain relevant term expressions or not. The best results are obtained for document collection 5 with a precision of 79%. This is also the largest document collection. The high precision on the Web document collection from the tourism domain can be explained by the fact that many of the accepted terms are valid geographic expressions such as `new_zealand`, `venice` or `sunshine_coast`. Whether to regard such candidates as good or bad is an open issue, depending on the task. The worst results originate from document collection 2 with a precision of 40%. These worse results can be explained by the fact that the keyword, which constituted the document collection, is polysemous: there are a couple of terms belonging to "ontology in philosophy" such as `martin_heidegger` and `philosophy_of_mind` and not to "ontology in computer science". This shows the influence of assembling a document collection. Focusing search results by

eliminating unwanted senses can be relatively easily done by adopting the query which constitutes the domain Web document collection.

Then we also evaluated lower rank regions of frequent text spans for document collection 5. There the precision values are lower than for the top 1000 most frequent text spans, but still reasonably good. The still high number of term expressions regarded as relevant is indicative of the following finding. Without further domain restrictions the vocabulary of the tourism domain (given by the query phrase "accommodation") is rather large. A vocabulary for the tourism domain, where also many proper names can be found, is likely to consist of many thousands or even hundreds of thousands of terms. This is also the reason why an evaluation against the vocabulary of known tourism gold standard ontologies is not feasible since most of the acquired term candidates are not within the gold standard though they are valid domain relevant term expressions.

When looking at the results for multiword term expressions separately (numbers in parenthesis of table 7.2), it can be seen that multiword terms are captured with reasonable quality. The quality ranges from 21% 79%. For the lower rank regions, the results for multiword terms are even above those for unigrams.

Table 7.2: Evaluation results for term candidates, the results for multiword terms are shown in parenthesis

Document Collection	Order Criterion	Evaluated Rank Region	Accepted	Rejected	Precision
1	frequency	1-1000	512 (148)	488 (165)	51% (47%)
2	frequency	1-1000	396 (60)	604 (223)	40% (21%)
3	frequency	1-1000	522 (214)	478 (277)	52% (44%)
4	frequency	1-1000	530 (197)	470 (256)	53% (43%)
5	frequency	1-1000	793 (240)	207 (93)	79% (72%)
5	frequency	10001-11000	619 (485)	381 (224)	62% (68%)
5	frequency	50001-51000	522 (497)	478 (300)	52% (62%)

Conclusion: In all performed experiments on term acquisition with XTREEM-T, approximately half of the candidates may be regarded as relevant term expressions.

7.4.2 Experiment 2: N-Gram Level Distribution

In the following we will show the distribution of the length of text spans (the number of tokens) found in the candidate terms list generated with XTREEM-T. The x-axis represents the number of tokens (the n of n-grams) while the y-axis shows the relative share of n-grams with this length in percentage. There are 5 chart lines obtained by the 1000 to 10,000,000 top ranked text spans. The diagrams of figure 7.5 show that the fraction of n-grams with higher n is steadily decreasing. For

10,000 and more topmost frequently considered n-grams, the fraction of unigrams is even lower than that of bigrams.

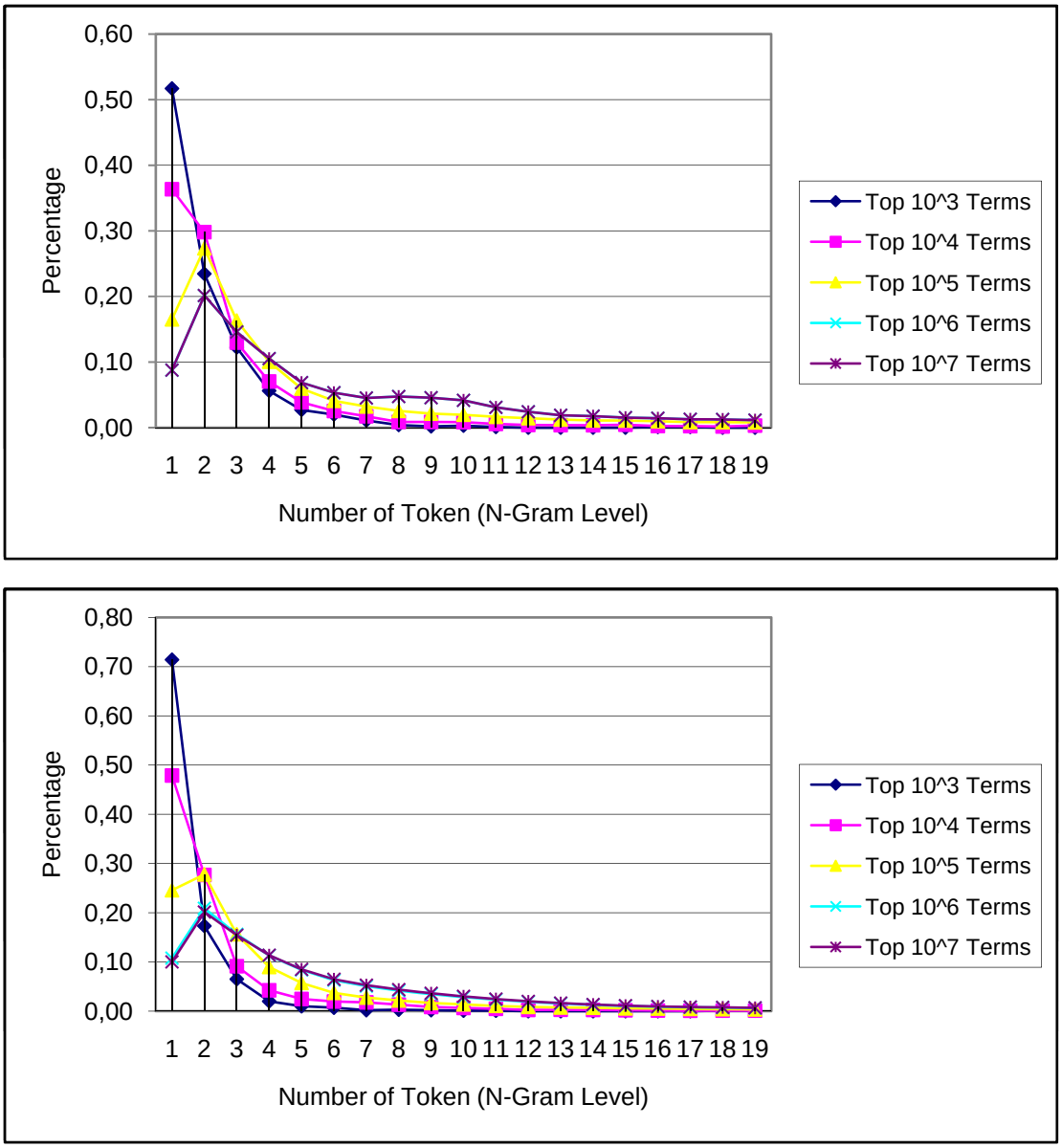


Figure 7.5: N-Gram level distribution among the top 1000 to 10,000,000 most frequent text spans for (a) document collection 2 and (b) document collection 4

7.4.3 Experiment 3: POS Patterns

In this experiment we want to investigate the constitution of frequent text spans regarding their Part-of-Speech (POS). The analysis of the POS is somewhat complicated, because POS taggers often rely on the context window of regular text to determine the proper POS. Since such a sequential textual context is not given to the terms of marked-up navigation elements, the outcome of a POS statistics might be insufficient. Furthermore, the POS tagger was not trained for the domains analysed. It is hard to judge how strongly the results are influenced. We found that the POS-Tagger (QTag[3]) didn't perform well on many processed words. We will, nevertheless, present the results and take conclusions under the assumption that the statistics is rather correct.

The POS pattern statistics reveals that most of the found candidate terms are noun constructs. This means that in some positions a noun word is participating in the term. This observation is not surprising since when one imagines what is to be marked-up, one does not usually markup "filling phrases". But text spans which are filling phrases are also created if a sequence of text is surrounded by marked-up text spans. The knowledge about what is a filling phrase and what it is not is not known, but frequency eliminates the rather strong varying filling phrases well.

Our method does no rely on the a priori application of a POS pattern filter to generate candidate terms. We investigated a posteriori the constitution of the POS patterns for the found candidate terms. This issue gives XTREEM-T further merits, since the availability of even a POS-Tagger incorporated in many natural language processing based term acquisition methods, which performs well in arbitrary domains, is unlikely; let alone parsers.

The finding is that a low fraction of text spans do not contain a noun component. For instance, for the top 100,000 most frequent text spans of document collection 4, only 2.3 percent of the POS patterns include no noun. This means that when short sequences of text are marked-up in Web documents, there is likely to be a noun involved. The lower share of non-noun term expressions, which is not categorically rejected by our approach, reflects a part of the vocabulary which is, therefore, not a priori excluded.

7.5 Conclusion

In this chapter we presented an approach for obtaining terms from Web document collections. The obtained terms comprise single word terms and multiword terms and thus this belongs to the rare number of methods which are capable of obtaining both at the same time. XTREEM-T can be seen as a special case of Group-By-Path, where not only tagpaths sets with at least two text spans are considered but single text spans also and where the text span sets are multi sets, multiple occurrences of the same text span at the same tagpath are considered as well. The

[3]http://www.english.bham.ac.uk/staff/omason/software/qtag.html

finding while inspecting the results obtained with XTREEM-T is that around half of the terms are terms which belong to the domain of interest. This observation is relevant for the approaches aiming at finding sibling pairs, since if a feature space of a Group-By-Path dataset is automatically obtained that can be expected to be of similar quality as the term lists obtained with XTREEM-T.

8 Finding Synonyms with XTREEM-S

In chapter 6 we showed that terms pairs where a sibling relation exists can be found based on the Group-By-Path operation with a satisfying quality. This was done by computing associations, more exactly first order associations. For the detection of synonyms, first order associations and second order associations can be applied. We will now also investigate whether second order associations based on Group-By-Path are beneficial for finding synonyms.

In ontology engineering, synonyms are terms which denote the same concept. The knowledge about synonymous terms can be reflected in the lexical layer of ontology entities, for instance, by having the synonyms as alternative labels or by more complex modelling [Buitelaar et al., 2009].

Synonyms are words with the same meaning or very similar meaning like *car* and *automobile*. A requirement for making words synonyms is that they are interchangeable. This means that they can be substituted against each other while the meaning of the surrounding text (or speech) remains constant. It is questionable if absolute synonymy, where a term can be exchanged against its synonym in all contexts, exists at all. In linguistics, synonymy [Cruse, 2004, page 154-156] of terms is discussed and several grades are distinguished. Cruse [Cruse, 2004, page 154] distinguishes absolute synonymy, propositional synonymy and near synonymy. The last one, near synonymy, is approached by several methods which try to obtain synonym relations from texts. Terms are regarded as near synonyms when they are exchangeable in some contexts. In this thesis we rely on the definition of synonyms as those of Wordnet [Fellbaum, 1998] synsets, where words are regarded as synonyms if they share a common meaning which can be used as a basis to form a concept relevant for the domain in question.

The basic hypothesis is that good synonym candidates are words which nearly never occur together but which have a very similar context. For example, `car` occurs often together with `bike` and `bus`. `automobile` occurs often together with `bike` and `bus`. But `car` and `automobile` only very seldom occur together. In [Dorow, 2006], it was shown that this hypothesis does not always hold true. This is valid for certain circumstances. One reason is that the change between synonyms is a narrow distance of textual context window. It is also not uncommon that two synonymous words are used in close coordination, for example, "Consequence in the form of `penalty` and `punishment` is the subject of the next chapter." (Example taken from [Dorow, 2006]). Dorow revised the thesis that synonymous words do not occur together (*first order association*), and stated that synonymous words

only have a similar context (*second order association*). This was done on regular consecutive plain text. Since the Group-By-Path approach enables us to obtain sets of terms which have a different "constitution bias" in comparison to traditional text access (for example by using Bag-Of-Words), we want to examine whether Group-By-Path data yields suitable results by means of the updated hypothesis that synonymous words have similar contexts (second order association).

We will introduce XTREEM-S (XTREEM for Synonyms), an approach for obtaining information about synonymy between terms of a given vocabulary. XTREEM-S uses a standard procedure for computation of second order association on a dataset based on the Group-By-Path operation described in chapter 3. In an experiment we will investigate if this approach is able to perform well in finding terms which are good candidates as synonyms compared to the Bag-Of-Words vector space model.

8.1 Related Work

The detection of information about near synonyms [Hirst, 1995, Inkpen and Hirst, 2003, Inkpen and Hirst, 2006, Inkpen, 2007b, Inkpen, 2007a] deals with the task of finding words which are interchangeable in some contexts. For this purpose the distributional hypothesis of Harris [Harris, 1968] is used, for example, in [Lin et al., 2003, van der Plas and Tiedemann, 2006, Freitag et al., 2005, Maria Ruiz-Casado and Castells, 2005, Inkpen and Hirst, 2002, Wu and Zhou, 2003]. Harris distributional hypothesis states that words that occur in the same contexts tend to have similar meanings. Recently, the Web was also used for synonym detection [Baroni and Bisi, 2004, Ruenes, 2007]. Several approaches focuses on the biomedical domain [Bogdan and Sacaleanu, 2002, Xiao and Rösner, 2004, Yu et al., 2002]. Recently also tags from Web 2.0 Web sites have been investigated [Cattuto et al., 2008]. Often terms have been required to occur together only rarely but to co-occur with a common set of terms. From Dorow [Dorow, 2006] we have obtained the revised hypothesis that for terms to become synonyms, they are only required to have similar contexts. LSA [Landauer and Dumais, 1997] was successfully applied for finding synonyms according to the TOEFL test [Turney, 2001].

8.2 XTREEM-S Procedure

For finding synonyms by means of statistical processing, the hypothesis of Dorow [Dorow, 2006] is that candidates for synonyms are terms which occur together with a similar context (of terms). In chapter 6 we have shown the computation of first order associations depicting sibling relations. To compute similar contexts, second order associations are

incorporated [Biemann et al., 2004a]. The overall procedure for XTREEM-S is shown in the dataflow diagram of figure 8.1.

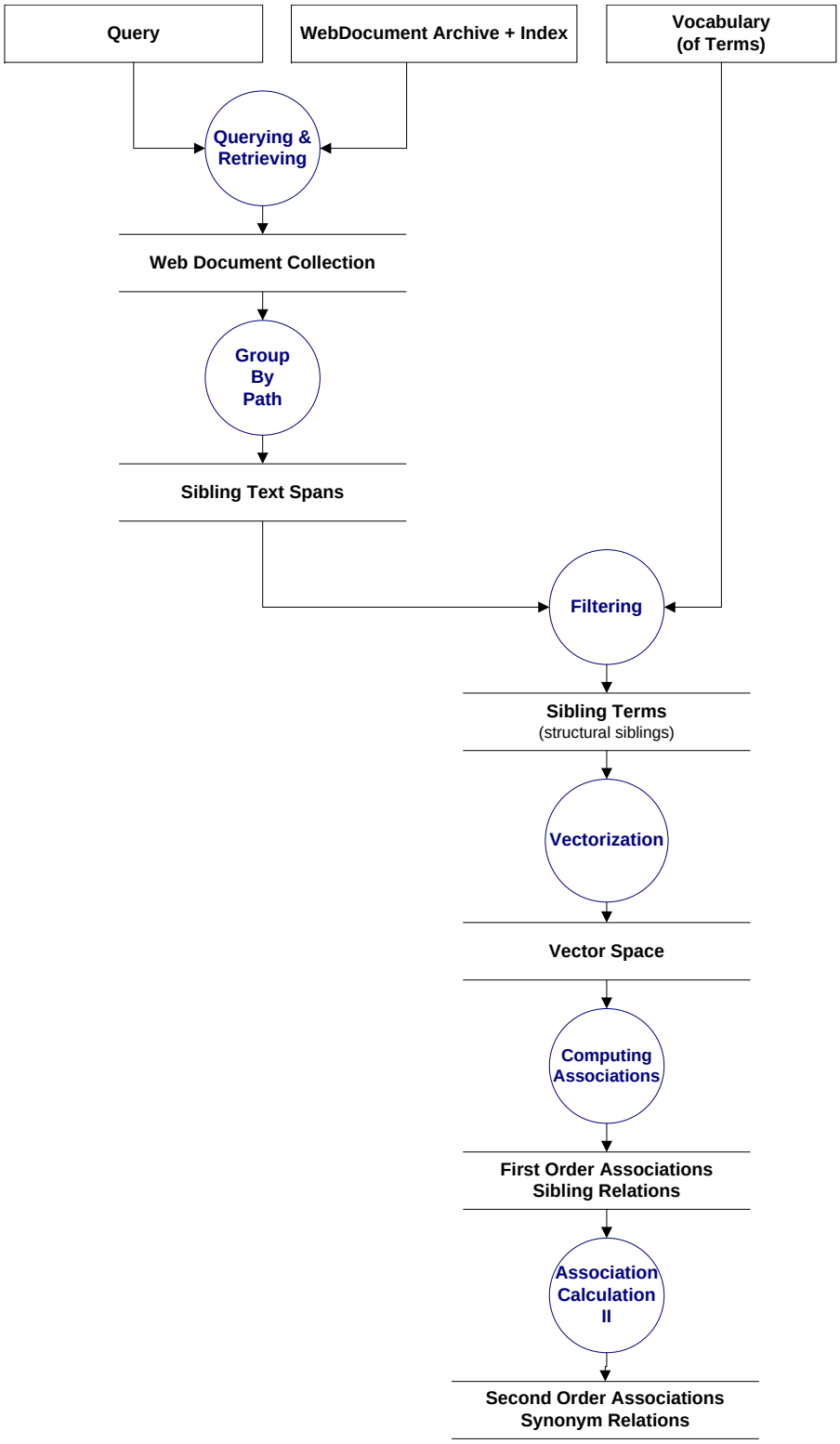


Figure 8.1: Dataflow diagram of the XTREEM-S procedure

8.2.1 Step 1 - Querying & Retrieving:

The first step is the same as those of the already described approaches, for example XTREEM-SG in chapter 4 or XTREEM-T in chapter 7. This step is described in detail in section 4.1.1. For a given query, a Web document collection is obtained. For our evaluation, the query "`tourism OR tourist`" was used to obtain a Web document collection of the tourism domain. This query resulted in a document collection of 1,468,324 documents.

8.2.2 Step 2 - Group-By-Path:

For each document the Group-By-Path algorithm, described in chapter 3 is applied. As a result we obtained a collection of 13,177,526 text span sets.

8.2.3 Step 3 - Filtering:

For the following steps we consider only the text spans which are contained in a given vocabulary. Further, we are only processing text span sets with a cardinality of at least two, otherwise no co-occurrence can be observed at all.

As the vocabulary to be processed we took the terms of two tourism gold standard ontologies described in section 8.3.2. Both ontologies did not contain synonyms; therefore, we additionally added the synonyms which occurred together with the terms of the initial vocabulary from the ontologies in Wordnet [Fellbaum, 1998] synsets. This was done to ensure that for the terms of the vocabulary a synonym relation exists for most of the terms. The enhanced vocabulary consists of 1786 terms. The synonyms which have been obtained from Wordnet are also used for evaluation as *gold standard synonym reference*.

From the 13,177,526 text span sets found in the Group-By-Path step, 864,431 sibling term sets, which are constituted by at least two terms of the vocabulary, have been obtained.

8.2.4 Step 4 - Vectorization:

The text span sets obtained are then represented as vectors. The feature space is given by the input vocabulary. For each term of the input vocabulary there is a corresponding vector spanned over all contexts (tag paths).

8.2.5 Step 5 - First Order Association Computation:

For each pair-wise combination of terms, the corresponding similarity is computed by similarity function S_1. With S_1 the similarity of the corresponding context vectors is computed. As a result, one obtains a symmetric matrix E_1 where the pair-wise term similarities are stored. This matrix E_1 represents the *first order association* among terms. Similarity is given by their joint occurrence within the same contexts.

As the similarity function of our evaluation we used the cosine similarity among tagpath vectors. Choosing cosine similarity results in association strength values ranging from 0 to 1. This makes the association computation of the next iteration easier. An alternative would be to do the computation as described in chapter 6, applying association measures on co-occurrence statistics.

8.2.6 Step 6 - Second Order Association Computation:

The matrix obtained in step 5 can also be treated as a list of vectors. For every term, there is a vector with the first order association score to all other terms, the so called context vector. Now, for each combination of first order association vectors, the similarity is computed again by similarity function S_2. The result is then stored in a matrix E_2. This matrix E_2 represents the *second order association* among terms. Similarity is given by similar context profiles (first order association vectors).

The choice of the association measure for the first iteration should be carefully examined; its choice has an impact on which association measure should be applied to get meaningful results in the second iteration association computation. As the similarity function of our evaluation for second order association computation, we again used cosine similarity. Applying cosine similarity upon cosine derived values yielded useable results. When for the first iteration a different association measure has to be used, the association measure of the second iteration has to be carefully chosen so that in the second iteration also useable results can be computed. For example, χ^2-association scores are not bound to the range [0,1] as cosine values. For computing the similarity between vectors representing χ^2 scores more techniques such as normalization have to be considered to make the vectors comparable in a manner yielding useful outcomes.

8.3 Evaluation Methodology

For the evaluation of synonymy detection in computational linguistics, the recent trend is to use the TOEFL task. In this task, one has to select a synonymous word out of a given set of 5 terms. For the evaluation of synonymy detection in the context of ontology learning, this is an unrealistic scenario, since for ontology learning usually a mid size vocabulary of several hundreds or thousands of terms is to be processed. Choosing synonym candidates out of a thousand candidate terms is much harder than to choose a synonym candidate out of 5. Instead we will perform an evaluation where no restriction is imposed. This is a much harder task, but this fits better to the context of ontology learning where no prior restriction can be expected in real world scenarios.

We will perform a gold standard evaluation. As reference, the synonym relations from Wordnet [Fellbaum, 1998] have been used. The synonym groups (synsets) obtained from Wordnet have been transformed into a collection of term pairs which

stand in a synonym relation. The evaluation has a bias against the automatically generated results since Wordnet (the reference) also contains synonyms for other domains than the used tourism domain. This is especially the case for terms with more than one sense. We perform no sense disambiguation. In more detailed experiments this circumstance should be accounted for.

The object of the evaluation is a ranked list of automatically obtained concepts pairs, whereas the ranking is given according to the second order association strength of the term pairs. For each automatically obtained term pair, it can be determined if this relation is also supported by the reference which gives a positive count. If a term pair is not supported by the reference, a negative count is assumed. With this, for each position in the ranked list, recall and precision can be computed.

8.3.1 Evaluation Criteria: Precision and Recall

The recall is the number of true synonym pairs already seen ($\#positive$) as against the number of synonym pairs given by the reference ($\#overall$). The precision is the number of true synonym pairs ($\#positive$) as against the number of automatically generated pairs ($\#positive + \#negative$). This is analogous to the evaluation of term pairs found with XTREEM-SP in chapter 6, but here we focus on synonym relations instead of sibling relations.

Definition 8.1 (Recall)

$$recall = \frac{\#positive}{\#overall} \tag{8.1}$$

Definition 8.2 (Precision)

$$precision = \frac{\#positive}{\#positive + \#negative} \tag{8.2}$$

For a ranked list of associated term pairs a chart line can be obtained for a series of measurements on precision values for several numbers of seen candidate term pairs. We evaluated the top-N candidate term pairs for several N. N increases from the left to the right (N=10, 20, 30, ..., 100, 200, 300, ..., 1000, 2000, 3000,..., 10000) .

8.3.2 Evaluation Reference

We use the synonym relations which can be assessed among the synsets member terms of Wordnet. Among the terms of the input vocabulary, 13854 synonym pairs can be assessed.

8.4 Experiment

As a contrast dataset, we applied the computation of first and second order associations on a dataset which was obtained from the same Web document collection but by means of the Bag-Of-Words approach. This is not the optimal way to obtain synonyms but used as a rough baseline. In figure 8.2, the results of our experiment are shown. The Group-By-Path approach is contrasted to the traditional Bag-Of-Words approach. Group-By-Path performs better than Bag-Of-Words. For the alternative BOW method, there are no synonyms at all within the top 100 evaluated candidate term pairs. For higher numbers of evaluated term pairs the lines approach each other on low level. For Group-By-Path there is a region ranging up to the first top 400 synonym candidates, where a precision of over 10 percent can be achieved. This means that there are a rather small number of synonyms which can be found with an acceptable precision. Compared to the overall number of synonyms which are supposed to be present according to the reference, this is only a small fraction. The corresponding recall values are low, for example, while observing 400 term pairs, with a precision of 10.97 percent, the recall is only 3.09 percent. To obtain a recall of 9.43 percent, 10000 candidate pairs have to be inspected; the precision is then only 1.33 percent, which is practically unacceptable. For the traditional Bag-Of-Words approach the results are even worse.

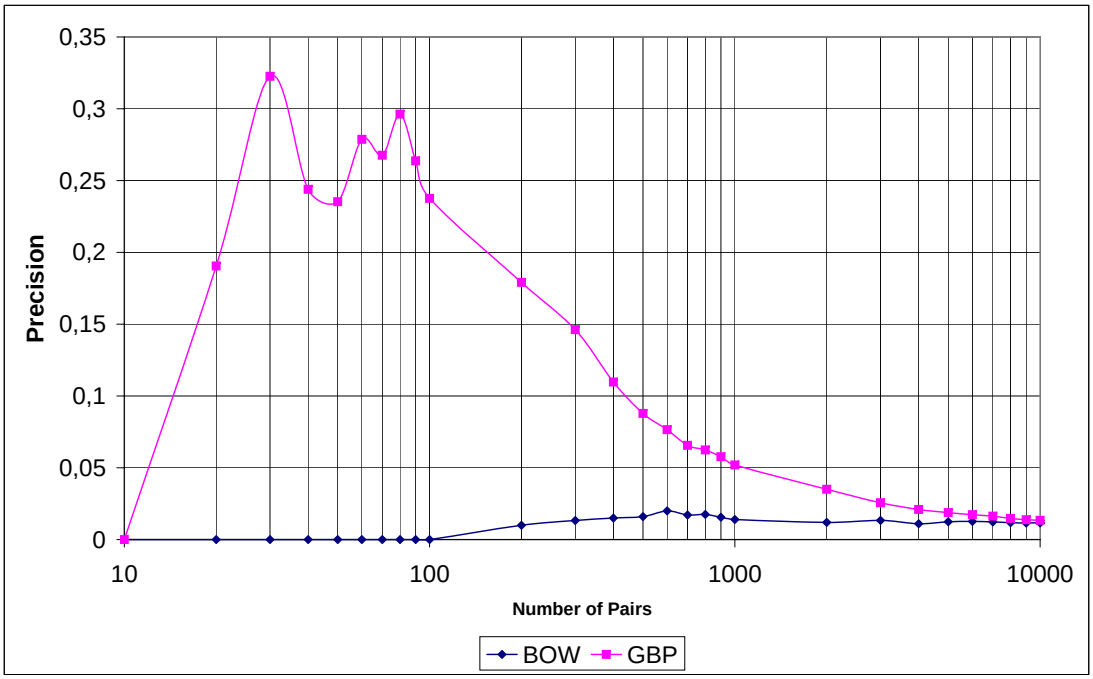


Figure 8.2: Precision and recall of Bag-Of-Words and Group-By-Path on finding synonyms

The recall which was achieved with an acceptable precision is too low to present it in a diagram. This is a circumstance which is common to automatic approaches for finding synonyms; they can only find a small fraction of synonym relations.

8.5 Conclusion

Although with Group-By-Path a better results on finding synonymous terms were achieved than with a traditional Bag-Of-Words approach even those improved results are inadequate for finding all synonyms which are expected to exist in a vocabulary. One can only state that it is possible to find a rather small number of synonyms with acceptable precision.

9 Domain Relevance enhanced Term Weighting for Learning Sibling Groups - XTREEM-SG$_{T,DR}$

In this chapter we present the TF×IDF×DR term weighting scheme. It is derived from TF×IDF term weighting and additionally incorporates a further domain relevance (DR) factor reflecting the degree to which a term is considered characteristic within the dataset in comparison to an external comparison ground. The newly proposed DR enhanced term weighting scheme is applied on a variant of an XTREEM-SG procedure where in contrast to chapter 4, the vocabulary depicting the feature space is automatically obtained by the XTREEM-T approach described in chapter 7. In such an automatically obtained feature space terms which are of less importance to the domain of interest can occur but they should get only little influence on the results.

Term weighting is often performed for processing textual data represented in the vector space model. The most prominent weighting scheme is TF×IDF [Salton and Buckley, 1987]. In the next section we describe the motivation for creating a new term weighting schema. The DR enhanced term weighting is supposed to bring up sibling groups given by cluster labels which are more domain relevant than without DR term weighting as we will investigate in the evaluation experiments.

9.1 Motivation

The motivation to extend the existing TF×IDF term weighting is twofold. The first is that the frequency distribution of terms in datasets obtained by Group-By-Path which is used to derive IDF scores is "different" or "distorted" compared to Bag-Of-Words text document datasets. This aspect is further described in section 9.1.1. The other reason described in section 9.1.2 is that in the context of clustering based ontology learning, the obtained results, sibling group clusters in particular and otherwise motivated clusters in general, are to be consumed by ontology engineers. The ontology engineer who intends to conceptualize a domain can be expected to be interested in "domain specific" concepts more than in general world patterns.

9.1.1 Distorted Occurrence Distributions

TF×IDF is intended to be useful on regular vector space models obtained by vectorising textual documents. The Group-By-Path approach presented in chapter 3 allows us to "access" and represent semi-structured Web documents in a different way compared to traditional text document vectorizations. A vectorization of a Web document collection with the aim of finding semantic sibling relations is described in chapter 4. The vectorization performed in the processing procedure of XTREEM-SG in chapter 4, a vocabulary of terms, is required input to the procedure. The vocabulary is manually crafted and, therefore, does rarely contain terms which are not of user interest. In practice, one cannot always expect the input vocabulary to be of high quality. The feature space might be automatically obtained without support of any terminology acquisition method at all. Even when there was an automatic acquisition of terms by means of a terminology acquisition method, as, for example, those described in chapter 7, the obtained vocabularies can be expected to be erroneous.

Text document vectorizations with TF×IDF term weighting can cope with noisy vocabularies. Terms which are referred to as stop words are handled well by TF×IDF on traditional Bag-Of-Words vectorizations; but this can be different for Group-By-Path vectorizations. If a feature space contains the term "the", for traditional Bag-Of-Words vectorizations there is likely to be a non-zero term score in each vector. The term "the" has nearly no "separation strength" and is outweighed by TF×IDF term weighting. In contrast, by accessing Web documents according to the Group-By-Path approach described in chapter 3, the term "the" might be captured as a candidate sibling term. And since this happens rather seldom, "the" occurs together with other sibling terms and can be scored by TF×IDF as if it is a reasonable good candidate term, TF×IDF does not punish this term as hard as for Bag-Of-Words vectorizations. We refer to this circumstance as "distorted frequency distributions". The "uninformative" terms which have a high frequency according to Zipf's law [Zipf, 1949] are not necessarily the terms with high frequency obtained in Group-By-Path vectorizations. The proposed TF×IDF×DR is supposed to be able to better cope with distorted occurrence distributions by incorporating a measure of term relevance influenced by external evidence. Terms which are captured in a certain fraction of paths, such as "home", "top", "feedback" and so on might be those which lead to the establishment of clusters. By potentially punishing such terms which are not characteristics for a domain Web document collection, those terms get less influence.

9.1.2 Interest towards Domain Relevant Terms

From our experiments on mining semantic sibling relations from Web documents on an open vocabulary by means of the XTREEM Group-By-Path approach [Brunzel and Spiliopoulou, 2006a], it became desirable to reduce the influence of non domain relevant terms on the results. Though correct

with respect to being siblings, clusters such as $July, August, September$ and $Thursday, Wednesday, Saturday$ have not been in the focus of the domain Web document collection which was about technologies ("semantic Web", "ontology" and "ontologies"), therefore, not informative for the human domain ontology engineer. One could manually create a Web document and/or domain specific stop word list. This is not desirable for several reasons. (a) it is laborious and more important, (b) it is not easy to decide if something is an irrelevant stop word or not.

If the feature space is automatically derived by domain relevance comparison, for example [Schaal et al., 2005], a Boolean decision is taken. If a term is included in the feature space because it has passed a certain threshold or it is within the top-n most domain relevant terms, the gradually notion of domain relevance is lost. We argue to keep (or push) this information on domain relevance into the subsequent processing. The processing, therefore, stays unsupervised to a bigger extent, though it can benefit from domain relevance information computed before. As a result, the clusters should be labelled by "domain relevant" terms to a bigger extent than without domain relevance enhanced term weighting which we will investigate in our experiments described in section 9.5.

9.2 Related Work

Related work on the combination of two different weightings methods is the work of Krkoska and Pekar [Pekar and Krkoska, 2003]. *Discriminative Feature Weighting* [Davidsson, 1997] and *Characteristic Feature Weighting* [Pekar and Krkoska, 2003] are combined for solving classification tasks. They obtain weights which are not only discriminating against other classes but which are characteristic of a certain class. Pekar and Krkoska [Pekar and Krkoska, 2003] and Pekar et al [Pekar et al., 2004] discuss the application of such weighting methods for the classification of words into predefined classes. In contrast we investigate the combination of term weighting approaches in a clustering task, where no classes which can be used for discrimination are available. The characterisation which is there done for classes is done for the entire data set within our approach and the characteristics are obtained by comparing to an external comparison base.

Next we describe the traditional TFxIDF [Salton and Buckley, 1987] term weighting which relies on the distribution of term occurrences within the dataset itself. Afterwards we describe the computation of DR scores which are obtained by comparing the occurrence within a dataset to an external contrastive dataset.

9.2.1 Term Weighting

The Term Frequency - Inverse Document Frequency [Salton and Buckley, 1987] or just TF×IDF is an often used term weighting approach applied for weighting term document vectors in clustering classification and information retrieval. TF×IDF favours terms which have a high discriminating power. One component is

the *Inverse Document Frequency* IDF [Jones, 1973]. The other is the occurrence frequency of terms in documents $TF_{t,d}$. Detailed discussions on IDF can be found in [Robertson, 2004, Papineni, 2001, Lee, 2007, Metzler, 2008]. Inverse Document Frequency is defined as:

Definition 9.1 (IDF)

$$IDF_t = \log \frac{N}{df_t} \tag{9.1}$$

N hereby refers to the overall number of documents in the document collection, df_t is the number of documents containing the term. As a result, the IDF of a rare term is high, whereas the IDF of a term occurring in many documents is low.

Combined together, TF and IDF depict the Document Frequency – Inverse Document Frequency (TF×IDF) term weighting scheme [Salton and Buckley, 1987].

Definition 9.2 ((TF×IDF [Salton and Buckley, 1987])

$$TF{\times}IDF_{t,d} = TF_{t,d} \cdot IDF_t \tag{9.2}$$

The main essence of TF×IDF term weighting is that terms which are present in many documents, and which can be regarded as not very distinguishing between document vectors, get a lower weight than terms which are characteristic of only some documents. TF×IDF term weighting relies on the occurrence frequency distribution within the dataset itself.

TF×IDF is traditionally applied in information retrieval and has been proved to be beneficial [Papineni, 2001]. It is also applied for text classification [Joachims, 1997, Debole and Sebastiani, 2003, Lan et al., 2005]. This is less relevant since our approach aims at improving unsupervised processing. There are plenty of variants on term weighting schemes. For example, in [Reed et al., 2006] an approach is proposed which is regarded as beneficial for streaming documents because a weight can be computed for new instances without considering the full corpus.

9.2.2 Domain Relevance

For the purpose of performing domain relevance enhanced term weighting, we will compute a domain relevance score. In this section we describe the computation of domain relevance scores. Within computational linguistics, there is the notion of the so called "domain relevance". Here a domain relevance score is computed which should express to which extent a term is characteristic within a certain "domain" compared to a contrastive basis. The "domain" is hereby given by a collection of text documents. A collection of documents is sometimes referred to as "corpus", but since it is often an object of discussion of what a corpus is and what it is not, we will usually use the term "document collection". For "relevance" also the term "specificity" is used in the literature. Comparing corpora has a

long tradition in corpus linguistics [Pierre, 1980, Damerau, 1993, Kilgarriff, 2001]. It is applied in several approaches such as in [Velardi et al., 2001b, Chung, 2003, Drouin, 2004]. The fundamental principle is to compare the occurrence frequency in characteristics in an analysis corpus (domain corpus) with the occurrence characteristics in a reference corpus (general language corpus). The reference corpus used for comparison is usually a large document collection, such as, for example, the British National Corpus (BNC) [Aston and Burnard, 1998]. Recently also the Web is used as contrastive reference [Kilgarriff and Grefenstette, 2003] and [Lüdeling et al., 2007]. In our experiments we also incorporate frequency counts obtained from the BNC and from the Web.

The domain relevance score should reflect the extent to which a term t is characteristic of a domain corpus compared to other terms of this corpus. The occurrence frequency is the primary object of comparison. There are different measures which can be applied to obtain a domain relevance score. One can use the χ^2-measure or simply the frequency share. The χ^2-measure is statistically motivated; bigger numbers are treated differently from the way lower numbers are treated. The computation of domain relevance scores can be done directly on the basis of frequency counts but also by comparing ranks after the terms have been ordered by their frequency. Both methods are described next. In the following RC refers to the reference corpus, AC to the analysis corpus.

Next we describe two variants for computing domain relevance scores. One method is to use relative frequency ratios, the other is to compute rank ratio scores.

Te computation of domain relevance scores according to a relative frequency ration is described in [Damerau, 1993] and [Manning and Schütze, 1999, page 175].

Definition 9.3 (DR by Relative Frequency Share Value – DR_{freq})

$$DR_{freq}(t) = \frac{\frac{f_{AC}(t)}{F_{AC}}}{\frac{f_{RC}(t)}{F_{RC}}} \tag{9.3}$$

$f(t)$ depicts the number of occurrences of a term within the analysis document collection and F is the sum of all frequency counts– the size of the corpus in number of terms. The domain relevance score is computed as the coefficient of the relative frequency coefficients from both document collections.

Example 3 *If a term occurs 100 times among 10,000 terms in the AC and 1000 times among 1,000,000 terms of the RC, the score is computed as:* $\frac{\frac{100}{10000}}{\frac{1000}{1000000}} = \frac{0.01}{0.001} = 10$. *This term occurs proportionally more often in the domain document collection to be analyzed Such a DR score would boost the term weight of this term. In contrast, if another term occurs 100 times among 10,000 terms in the AC and 10000 times among 100,000 terms of the RC, the score is computed as:* $\frac{\frac{100}{10000}}{\frac{10000}{100000}} = \frac{0.01}{0.1} = 0,1$. *This term is not very indicative for the analyzed document collection. Its DR score will reduce its term weight.*

An alternative method for computing a DR score is to compare the rank of the term in both corpora. To do so, the terms are ordered according to their frequency in the corresponding corpora. Now the resulting ranks of the terms in both ordered term lists can be compared.

Definition 9.4 (DR by Rank Ratio – DR_{rank})

$$DR_{rank}(t) = \frac{rank_{RC}(t)}{rank_{AC}(t)} \tag{9.4}$$

By computing the ratio of the ranks, two ordinal numbers are used, the resulting values are, therefore, not directly derived. The ranks extracted from particular occurrence frequencies are acceptable for a heuristic. DR scores are incorporated as heuristics of bonus and malus in the TF×IDF×DR term weighting schema.

Example 4 *If a term has rank 1000 in the RC and rank 100 in AC the resulting score is computed as* $\frac{1000}{100} = 10$. *But a term that has rank 1000 in the RC and only rank 10000 in AC can be expected to be less relevant if it has a DR score of 0.1* ($\frac{1000}{10000} = 0.1$)

Common to both DR$_{freq}$ and DR$_{rank}$ is to give a score which can be applied to boost domain relevant terms and punish less domain relevant terms. Subsequently, the DR component decreases or increases the term weight. The different approaches of domain relevance computation result in different value distributions. In our experiments in section chapter 9.5 we will use both approaches.

9.3 XTREEM-SG$_{T,DR}$ Procedure

The process wherein we apply the DR enhanced term weighting is a variant of the XTREEM-SG process described in section 4.1. We denote this procedure as XTREEM-SG$_{T,DR}$ – XTREEM-SG incorporating XTREEM-T and DR enhanced term weighting. The feature space is not given as input but obtained with XTREEM-T (described in chapter 7). The XTREEM-SG$_T$ part of the process (without the DR enhanced term weighting) was initially just called XTREEM [Brunzel and Spiliopoulou, 2006a]. We enhance this process by term weighting according to the TF×IDF×DR instead of just using TF×IDF. The overall process is shown in the data flow diagram of figure 9.1.

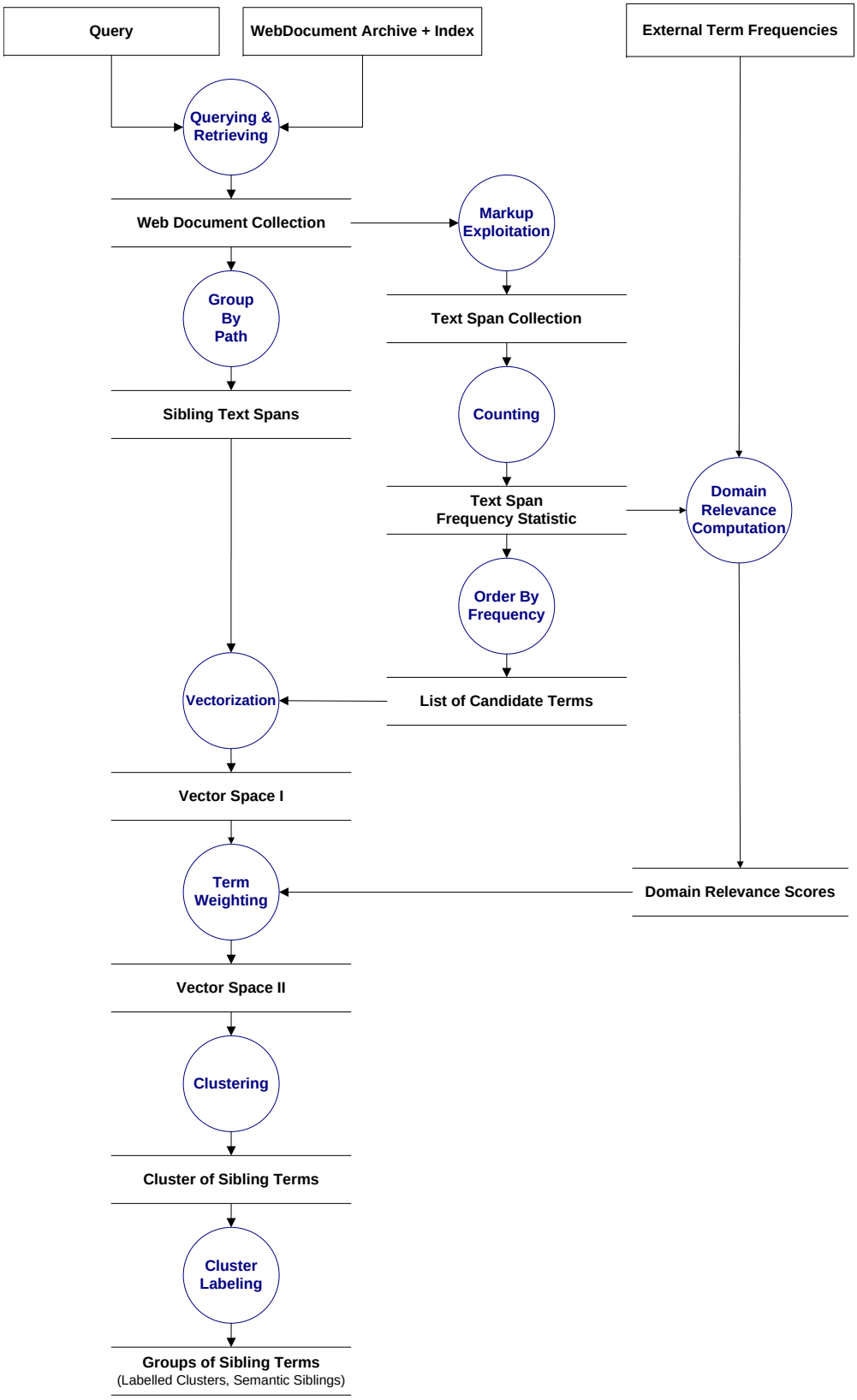


Figure 9.1: Dataflow diagram of the XTREEM-SG$_{T,DR}$ procedure

In the following we describe the computation of TF×IDF×DR term weights. In section 9.2, we described the computation of TF×IDF term weighting scores and DR domain relevance scores; those two approaches are combined into the TF×IDF×DR term weighting schema.

Definition 9.5 (TF×IDF×DR)

$$TF{\times}IDF{\times}DR = TF \times IDF \cdot DR \tag{9.5}$$

This results in a term weight which uses the "inner" occurrence distribution given by TF×IDF and additionally the outer occurrence distribution given by a domain relevance score DR.

For computing DR scores $f_{RC}(t)$ and F_{RC} have to be obtained. While using the Web as external reference document collection, it is not possible to obtain the overall number of terms constituting the Web documents index by the Web search provider. We, therefore, sum the frequencies for all terms we consider. Subsequently, the score of DR_{freq} is altered to a certain extent, the DR_{rank} scores remain unchanged. But using the Web as a source for term frequencies has other advantages which make that fact acceptable.

9.4 Evaluation Methodology

A gold standard evaluation is not suitable for the evaluation of the TF×IDF×DR term weighting, because for the gold standard evaluation, the vocabulary of the gold standard is usually given as input in the form of the feature space as shown in chapter 4, chapter 5 and also other approaches [Cimiano and Staab, 2005]. In such a scenario, only approved terms are present, and the TF×IDF×DR approach would be of limited advantage. The aim of incorporating domain relevance in term weighting is to obtain clusters which are characterised by domain relevant terms to a bigger extent than without a DR incorporating term weighting. For proving this objective we will perform several measurements where the overall domain relevance of a clustering is determined. Those measures are described next. Then we describe the inputs and the parameters we vary on the procedure.

9.4.1 Evaluation Criteria: DR$_{Sum}$

The goal of the evaluation is to determine the amount/extent of domain relevant terms characterising clusterings. For this purpose we define 3 criteria.

$DR_{Sum}I$: We multiply the fraction a certain term occurs within a cluster (relative inner cluster frequency) with a computed value of domain relevance. We do so for all terms of a cluster, and for all clusters. The resulting sum value reflects how much the cluster characteristics of a clustering are influenced on domain relevant/specific terms.

$DR_{Sum}II$: In contrast to $DR_{Sum}I$, also the cluster labelling is applied. With this measurement we exclude terms that only have a minor share in a cluster, regardless of their domain relevance. Only one occurrence of a term in a label is considered. Multiple occurrences are treated as one occurrence, in other words, the set of unique labelling terms is multiplied by the corresponding domain relevance value.

$DR_{Sum}III$: In contrast to $DR_{Sum}II$, only terms with a positive domain relevance/specificity are counted. Here the domain relevance value is set to 1 for all domain relevant terms and 0 for all terms with a "negative" (DR < 1) domain relevance.

9.4.2 Evaluation Reference

Evaluation references are the domain relevance scores obtained from the reference corpora described in the next section.

9.4.3 Inputs

We applied the XTREEM-SG$_{T,DR}$ procedure described previously in the section 9.3. The establishment of the document collection is the first task of the XTREEM-SG$_{T,DR}$ procedure. The seed query terms are the keywords "Semantic Web", "Ontology" and "Ontologies". We issued these queries towards the Google Web Search service[1] in October 2004. The result was a set of 4209 distinct URLs, from which we retrieved 4015 Web documents from 2112 domains. From these, we have removed approximately 10 percent documents that were recognized as non-English language documents. According to the XTREEM-T procedure, the Web documents have been converted to XHTML and the frequencies of text elements over the whole document collections have been counted. We have chosen the 1000 most frequent text spans as features. The Group-By-Path algorithm has processed 22462 tagpaths. For the purpose of finding siblings, only sets with at least two text spans are required. As a result 7713 sibling groups have been vectorized.

Reference Corpora: For the computation of domain relevance we will use three different reference corpora. First we use the frequencies form the British National Corpus (BNC) [Aston and Burnard, 1998]. For obtaining frequency counts from the Web we consider the frequency counts which could be obtained by the publicly available Web search services of Google[2] and Yahoo[3]. For the Web search services, the option for only "English language" documents was activated. If a term was not available in a reference, the corresponding term weight is not altered, a neutral score of 1 is assumed.

[1] http://code.google.com/apis/soapsearch/index.html, accessed on October 2004
[2] http://code.google.com/apis/soapsearch/index.html, accessed on July 2006
[3] http://developer.yahoo.com/search/web/V1/webSearch.html, accessed on October 2006

9.4.4 Variations on Procedure and Parameters

Domain Relevance Computation: We will compute domain relevance according to the Relative Frequency Share (DR_{freq}, Definition 9.3)) and according to the Rank Ratio (DR_{rank}, Definition 9.4)). Both approaches are described in section 9.2.2.

Variants of Term Weighting Schemes: We will apply term weighting according to the TF×IDF×DR scheme and according to the TF×IDF scheme. Furthermore, we apply pure TFxDR and just TF. For finding sibling terms only Boolean occurrence of terms is relevant. Consequently, the term frequency TF is zero or one. This results in a Boolean variant of the vector space model while only TF is used. The other three first mentioned schemes (TF×IDF×DR, TF×IDF, TFxDR) are then simplified to IDFxDR, IDF and DR.

Unit Length Normalization: For the vector space model it is possible to normalize or not normalize vectors to unit length. We use both variants, unit vectors and vectors with the Boolean values.

Clustering: For the number of clusters to be generated by the K-Means clustering algorithm, we set K=100, a heuristically chosen value which is suitable for 1000 features. Since the result of a K-Means clustering is dependent on the seed centroids, we will perform each clustering 10 times with different randomly chosen seed centroids. The DR_{Sum} values will be averaged over these 10 runs.

Cluster Labelling: For each feature, the relative within cluster support can be computed. All terms which have a within cluster support above a certain threshold τ are used to label a cluster. The within cluster support is the relative fraction of instances of a cluster in which a certain feature occurs. Those terms form a sibling group. For τ we use the heuristically sound values of $\tau = 0.2$ and $\tau = 0.5$. See section 4.1.5 for a description of cluster labelling.

9.5 Experiments

Computing the DR_{Sum} of a clustering relies on DR as evaluation input itself. The highlighted diagonal of the matrix for the result (table 9.1, 9.2, 9.3 depicts the combination where the evaluation is performed with the same DR computation setting as used for term weighting. The highest values of each column (DR computation setting) are highlighted by bold style numbers.

9.5.1 Experiment 1: $DR_{Sum}I$

In the first experiment we conducted the processing with and without unit length normalization. Table 9.1 shows the results. For the results obtained while

performing unit length normalization there is a clear trend: term weighting incorporating DR is always better than term weighting not incorporating DR. For the results not using unit length normalization the results relying on computing DR_{rank} do not confirm the former observation. For this constellation traditional IDF gives the highest measured $DR_{Sum}I$ results.

Table 9.1: $DR_{Sum}I$ with (a) and without (b) unit length normalization

			with unit length					
			bnc		google		yahoo	
			freq	rank	freq	rank	freq	rank
-	-	-	88041	3302	1444267	4253	1234187	3327
-	-	IDF	91198	4129	1313978	5985	1113277	4242
bnc	freq	DR	101802	3878	1686530	6566	1474482	3980
		DR_IDF	109286	**4260**	1797919	**7069**	1518519	**4446**
	rank	DR	35844	2841	6655	3736	12612	2601
		DR_IDF	35588	2794	6710	3937	12256	2599
google	freq	DR	121552	3782	2195621	4870	1695343	4047
		DR_IDF	**124752**	3973	**2244363**	5244	1713570	4287
	rank	DR	86140	2953	1392441	3556	1166881	2936
		DR_IDF	86015	2965	1444027	3565	1163825	2918
yahoo	freq	DR	115062	3740	2163835	5106	1668258	3901
		DR_IDF	120489	4004	2218186	5644	**1736469**	4197
	rank	DR	84947	2937	1458456	3537	1146079	2813
		DR_IDF	85232	2948	1455027	3504	1134075	2796

			without unit length					
			bnc		google		yahoo	
			freq	rank	freq	rank	freq	rank
-	-	-	94783	3508	1553142	4664	1295142	3586
-	-	IDF	113899	**3697**	1997611	**4971**	1591797	**3849**
bnc	freq	DR	155902	2127	2895411	2507	2430355	2520
		DR_IDF	**163433**	2056	3193150	2408	2682369	2519
	rank	DR	29659	2991	4880	3450	14138	2581
		DR_IDF	31849	2986	5381	3333	14582	2569
google	freq	DR	91752	1445	**5594077**	2035	3704046	1685
		DR_IDF	94684	1395	5519429	1968	3639378	1642
	rank	DR	52187	2969	628851	4486	496797	3158
		DR_IDF	56090	2927	690737	4338	573977	3102
yahoo	freq	DR	94333	1432	5352077	2047	**4043380**	1758
		DR_IDF	95726	1301	5394219	1856	4032325	1638
	rank	DR	55067	2855	649005	3939	550395	3091
		DR_IDF	59001	2850	752128	3775	606587	3067

9.5.2 Experiment 2: $DR_{Sum}II$

$DR_{Sum}II$ is dependent on a cluster labelling threshold and it is clearly distinguished whether a feature is in the cluster label or not. In this experiment we apply two different cluster labelling thresholds. Table 9.2 shows the results obtained by computing $DR_{Sum}II$ for a relative within clustering support threshold of $\tau = 0.2$ and $\tau = 0.5$. For both labelling thresholds there is again a clear trend; for $\tau = 0.2$:

term weighting incorporating DR is better regarding $DR_{Sum}II$ than term weighting not incorporating DR. For $\tau = 0.5$ there is only one outlier not conforming to our hypothesis.

Table 9.2: $DR_{Sum}II$ for labelling threshold $\tau = 0.2$ (a) and $\tau = 0.5$ (b)

			labelling threshold = 0.2					
			bnc		google		yahoo	
			freq	rank	freq	rank	freq	rank
-	-	-	32853	1370	1121928	1591	778968	1348
-	-	IDF	30851	1359	873437	1577	495569	1336
bnc	freq	DR	47873	1360	1197305	1571	892668	1334
		DR_IDF	50516	1368	1287164	1580	870184	1345
	rank	DR	16585	1378	2517	1444	5570	1171
		DR_IDF	16562	**1379**	2441	1446	5525	1172
google	freq	DR	50681	1336	2169107	1617	1414617	1370
		DR_IDF	**52084**	1332	**2202251**	1626	1508750	1378
	rank	DR	30968	1344	839903	1581	618024	1333
		DR_IDF	30594	1344	749615	1579	548569	1329
yahoo	freq	DR	49623	1349	2087297	1628	1513275	1389
		DR_IDF	50694	1355	2038614	**1632**	**1529218**	**1398**
	rank	DR	30498	1358	841627	1576	633386	1338
		DR_IDF	31286	1357	1017864	1586	694542	1348

			labelling threshold = 0.5					
			bnc		google		yahoo	
			freq	rank	freq	rank	freq	rank
-	-	-	11782	1031	74209	**1235**	54057	**1001**
-	-	IDF	11619	996	12580	1198	33804	968
bnc	freq	DR	19154	1003	289845	1179	189093	970
		DR_IDF	**19708**	997	288942	1169	186982	962
	rank	DR	8498	1045	1146	1128	3923	869
		DR_IDF	8717	**1047**	1206	1129	3984	872
google	freq	DR	16963	951	578671	1202	335976	966
		DR_IDF	16912	947	**636917**	1196	335721	947
	rank	DR	10163	987	10857	1209	33429	970
		DR_IDF	10533	982	12116	1212	36153	974
yahoo	freq	DR	16461	957	529854	1212	321161	994
		DR_IDF	16690	946	534099	1196	**341655**	985
	rank	DR	11128	992	16868	1206	53400	990
		DR_IDF	11729	992	17618	1209	54404	996

9.5.3 Experiment 3: $DR_{Sum}III$

The results measured by $DR_{Sum}III$ are shown in Table 9.3. This number depicts how many of the rather domain relevant terms are found in the cluster labels. For this version of DR_{Sum} our hypothesis can be confirmed again: DR incorporating term weighting yields better results. The most numbers of domain relevant terms are created by clusterings relying on term weighting incorporating domain relevance, in concrete by using IDFxDR.

Table 9.3: $DR_{Sum}III$

			bnc		google		yahoo	
			freq	rank	freq	rank	freq	rank
-	-	-	193	148	154	182	161	177
-	-	IDF	193	148	150	178	158	174
bnc	freq	DR	195	144	163	184	169	180
		DR_IDF	200	146	167	187	173	184
	rank	DR	166	149	133	153	139	151
		DR_IDF	168	150	135	154	140	153
google	freq	DR	204	146	184	199	186	195
		DR_IDF	206	147	**187**	202	190	199
	rank	DR	178	141	153	175	157	168
		DR_IDF	178	140	153	175	156	167
yahoo	freq	DR	205	149	184	200	190	198
		DR_IDF	**210**	**153**	**187**	**204**	**194**	**203**
	rank	DR	181	145	151	176	159	173
		DR_IDF	183	145	153	177	162	175

9.6 Conclusion

In this chapter we presented the TF×IDF×DR method for weighting terms on a vector space model. We have presented our term weighting approach which combines two widespread approaches TF×IDF and DR into one. Traditional term weighting only relies on the inner occurrence distribution of terms, the occurrence distribution within a dataset to be processed. For supervised learning like classification, where the separability of terms/features is of major interest, this can be regarded as sufficient. This can be different for unsupervised learning approaches like clustering. There the results are often presented to a human user with the aim of revealing patterns which are highly relevant within the domain of interest. General world knowledge given by patterns of not so domain relevant terms is likely to be of minor interest and should be automatically discarded if possible.

With $DR_{Sum}I$ to $DR_{Sum}III$ we intended to measure the contribution of domain relevant terms on the results of a clustering. Our experiments support our hypothesis that IDFxDR term weighting can be regarded as leading to better results regarding bringing domain relevant terms on top of labelled clusters than traditional term weighting without domain relevance. The computation of domain relevance by rank comparison also supported our hypothesis that DR enhanced term weighting produced domain characteristics clusters, when unit length normalization was incorporated. Not using unit length normalization, domain relevance by rank comparison did not support the hypothesis. The evaluation scores $DR_{Sum}I$ to $DR_{Sum}III$ verified our hypothesis over different processing stages. DR enhanced term weighting brings indeed domain relevant terms to the top labelling features of a cluster. The differences are not that large, but for human consumption even small improvements are desirable.

We observed that using the frequency from Web search engines is appropriate for computing domain relevance scores. Frequency counts of Web search engines are suited for computing domain relevance for weighting feature spaces (limited in size) since one can expect that there are general language frequency counts for all features (compared to the BNC, outdated, limited in coverage).

10 Indexing and Retrieving of Sibling Terms with – XTREEM-SL

The Web represents a huge collection of documents on arbitrary content. Indexes for searching Web documents are an important service. Usually, documents are retrieved to be read by a human consumer who has an information need which is satisfied by documents or by document fragments. Nowadays there is a shift from working on a document level towards working on the concept level. The need for working on concept level for example arises on engineering ontologies. By means of the approach described in this part, the user can search for sibling terms, terms which are supposed to represent ontological entities standing in sibling relation.

In this chapter we present an approach to establish an index over large amounts of Web documents regarding sibling terms. We call this approach Xhtml TREE Mining for Sibling Lists (XTREEM-SL). While using XTREEM-SL, we expect to find lists of sibling terms for one or more given seed siblings. Figure 10.1 shows the concepts "`river`" and "`mountain`", already mentioned in figure 1.3, of the introduction chapter. While using "`river`" and "`mountain`" as input, we expect to obtain a list of plausible siblings such as "`valley`", "`desert`" and "`glacier`".

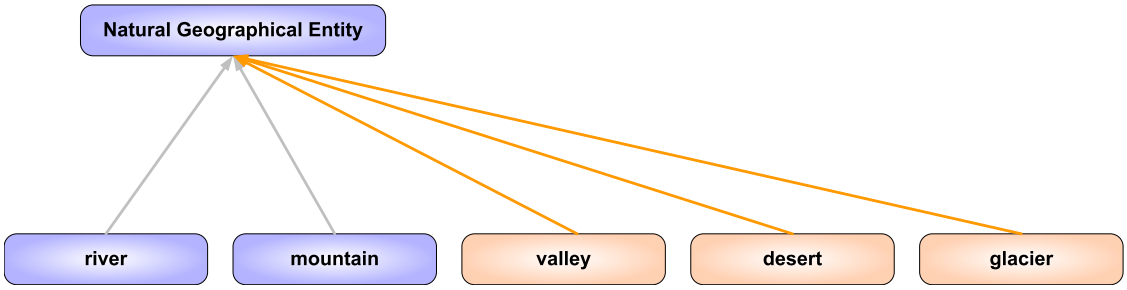


Figure 10.1: Example hierarchy of geographic entities where the sibling concepts depicted by orange boxes have been added

The XTREEM-SL is a system which allows us to retrieve sibling terms of an open vocabulary. The XTREEM-SG and XTREEM-SP procedures were conducted while processing a closed vocabulary, the terms to be processed have been restricted by the terms given as input in advance. There are good reasons for working with a closed vocabulary. On the one hand, there are indeed real world scenarios where

a vocabulary is given which should be structured. The other reason is the ability to evaluate the obtained results. For the purpose of evaluating a procedure on how good it is on finding sibling relations, processing a closed vocabulary is more appropriate since otherwise the ability to find a vocabulary would also be measured at the same time. But for real world scenarios being capable of processing an open vocabulary is more desirable. The level of detail which should be represented in an ontology can be expected to vary largely since an ontology as an abstraction of reality cannot model every aspect with the same maximum level of detail. By processing an open vocabulary the early restriction on a limited set of terms is avoided.

This focused retrieval of sibling terms does not require the application of time consuming processing techniques such as clustering as it is required by XTREEM-SG. This enables us to provide results in a rather short time frame; since the process can be easily performed in parallel, ad hoc answers can be achieved more easily than, for example, for XTREEM-SG.

The contribution of this chapter and the presented XTREEM-SL is not limited to ontology learning in general and the acquisition of sibling terms in particular. The acquisition of semantic relations is an important task but equally important is the assessment of existing ontologies. Often ontologies do not find widespread adoption; this can also be due to design errors which could be easily eliminated if the authors would have been alerted on potential suspicious constellations. An application of XTREEM-SL is to use it for the evaluation of existing ontologies. During the experiments we will see that insights into different "ontology realization problems" are given rather as a by-product. The application of the presented approach is not exclusively for the acquisition of sibling relations but also for the evaluation of existing ontologies regarding the plausibility of their labelled sibling entities.

10.1 Related Work

Related to our work are two methods for finding similar terms, given a set of input terms. (1) The proprietary, unpublished algorithm behind Google Sets[1] and (2) an approach on Bayesian sets [Ghahramani and Heller, 2005] which tries to give a published alternative to (1). Google Set delivers for a given number of up to 5 terms a result with varying number of related terms. In section 10.4.6 we will compare the results obtained by our XTREEM-SL approach with those delivered by Google Sets. The approach based on Bayesian Sets does not use the structure of Web documents as we do; their approach can operate on large but still closed vocabularies, a threshold has to be set and, therefore, not all terms are processed.

[1] http://labs.google.com/sets

10.2 XTREEM-SL Procedure

The overall XTREEM-SL approach presented in this chapter is realized by two sub-processes. First, a time consuming offline process for the index creation, described in section 10.2.1 and secondly, a process for retrieving sibling terms described in section 10.2.2 where results are delivered in near real time.

10.2.1 Creating the XTREEM-SL Index

In this section we describe the creation of an index structure over large amounts of sibling text spans. Raw groups of sibling text spans are obtained by applying the Group-By-Path operation. Those raw groups of sibling text spans are then indexed. The challenge herein lies in creating the index efficiently so that a relatively compact index is created preserving as much as possible of the information which can bring up desirable results. The overall procedure for the index creation is shown in the dataflow diagram of figure 10.2. Next, the single steps are described.

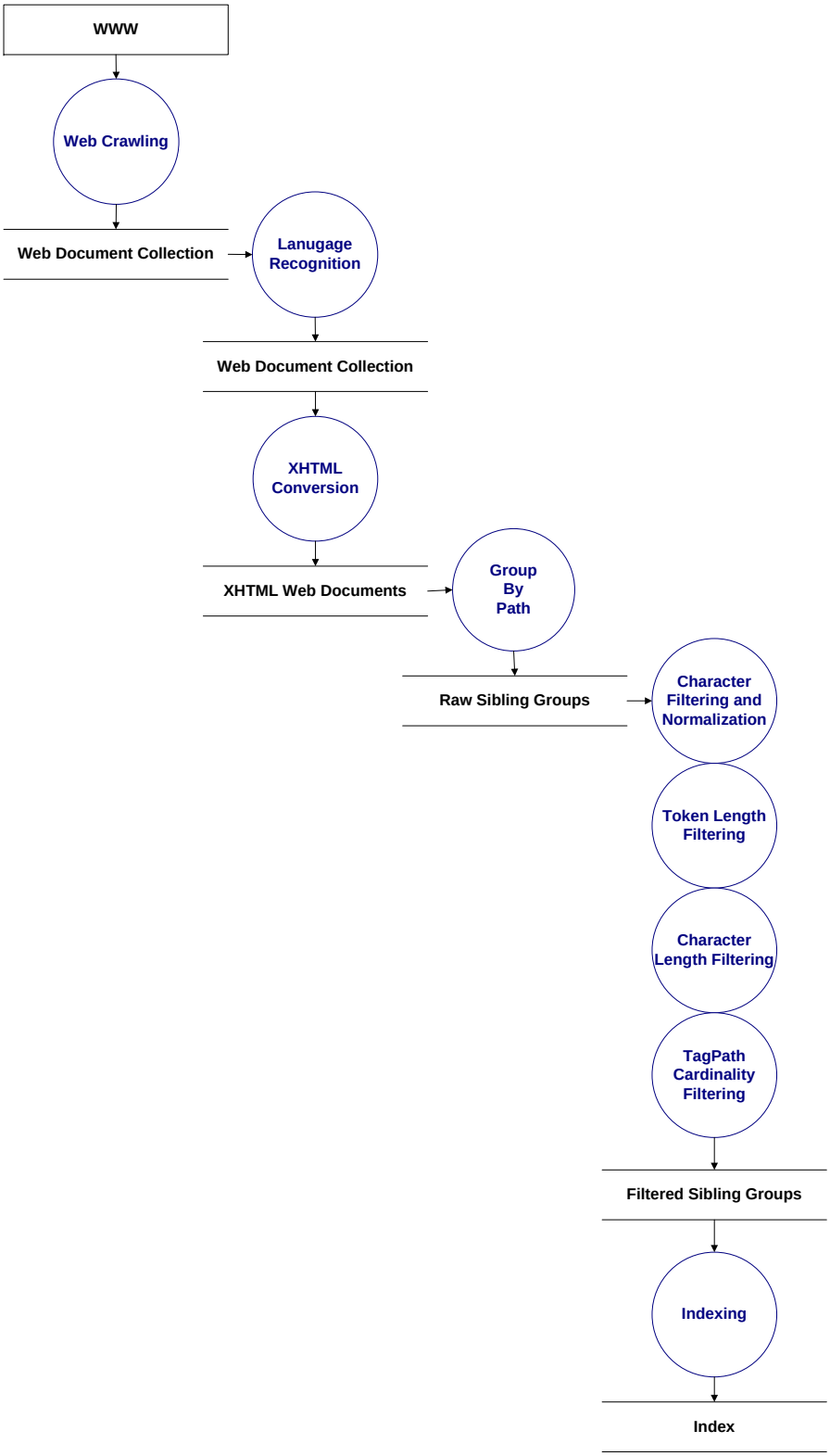


Figure 10.2: Dataflow diagram for creating a XTREEM-SL

Step 1 - Web Crawling: A necessary precursor is to have large amounts of Web documents locally available. Performing Web crawls is not only a conceptual problem but also something which has a significant engineering aspect. Those circumstances of Web crawls have already scratched in section 4.1.1. Here we rely only on the circumstance that a Web crawl was conducted and that the Web documents have been fetched locally. This Web crawl has to cover sufficient amounts of relevant pages. The size of such a Web document collection will typically range from thousands of pages up to billions of pages. For our experiments we crawled about 20 million Web documents, approximately 1/1000 of the amount of Web documents indexed by major Web search engines at the time when the crawl was performed[2].

Step 2 - Language Recognition: In principle, the presented procedure is language independent and, therefore, documents from arbitrary languages can be indexed together. In our earlier experiments we did not incorporate language identification which was often not problematic regarding the obtained results. For the sake of building a compact index, which reflects the target domain, it can also be important not to index documents from languages different from the target language. If flexibility regarding the indexed languages is desired, it is also possible to put information about the recognized language within the index, so that a restriction on languages can be done on retrieval time. For language recognition the language recognizer[3] provided by the Nutch Web crawler was used.

Step 3 - XHTML Conversion: A necessary requirement for performing the Group-By-Path operation is that the potentially non XHTML conformant Web documents are converted to well-formed XHTML documents.

Step 4 - Group-By-Path: On each XHTML document the Group-By-Path operation described in chapter 3 is applied. For each Web document several sets of text spans are obtained.

Step 5 - Character Filtering and Normalization: The aim of this step is to normalize the input text sequences. The input text spans from the Web documents were processed as follows:

- Characters have been converted to lower case.
- Non alphabetic characters have been replaced by whitespace.
- Multiple occurrences of whitespace are replaced by one whitespace.
- Leading and trailing whitespace is eliminated.

[2]September / October 2005

[3]http://wiki.apache.org/nutch/LanguageIdentifier,
http://lucene.apache.org/nutch/apidocs/org/apache/nutch/analysis/lang/LanguageIdentifier.html

This filtering and normalization of the raw character sequences cleans the input text content from the manifold presence of whitespace in marked-up Web documents. The character filtering can also be done on other criteria, by allowing only alphabetic characters of the target language. By doing so, numerical characters, as they appear on numbered list items or headings, are removed. Punctuation is filtered out as well. For an index which should not be targeted to any particular language or language family (such as Latin languages), this step can be relaxed towards only eliminating numbers and punctuation and normalizing the whitespace occurrences.

The textual content of semi-structured Web documents can be of arbitrary character length. Subsequently, a text span can be constituted by an arbitrary number of tokens, whereas a token is a character sequence without whitespace characters. For many text spans, which do not contain whitespace, the situation is straightforward; the text span is likely a valid term. Other text spans composed of several tokens often correspond to multiword terms of several words. In principle, Group-By-Path groups text spans of arbitrary length. But practically, only small text spans are likely to be a term expression, which is worthy to be indexed and to be presented to a human user. For example, when looking at figure 3.4, there are text spans such as "`Dangerous Sharks`", "`There are some shark species` ..." "`Great White Shark`" and so on. Some of them are valid term expressions, for example, "`Dangerous Sharks`", whereas other text spans such as "`There are some shark species` ..." are more complex linguistic constructs. The terms of our natural language tend to be constituted by a rather small number of words (tokens). For an increasing number of words, there are much fewer terms in the vocabularies of natural languages. For example there is only an extraordinarily small number of terms which are constituted by more than 5,6,7,... words. This observation can be used to create a more efficient index structure. If all text spans would be indexed, the index would grow (at least) to the memory space of the textual content of the entire Web document collection in the whole. By limiting the maximum number of tokens a text span to be indexed is allowed to consist of, unlike term expressions can be eliminated to create an index which is more memory compact without losing much of the potentially valuable information.

Step 6 - Text span Token Length Filtering: As already mentioned (for example, in section 7.4.2 of the XTREEM-T chapter), text spans consisting of large numbers of tokens are unlikely to be valid term expressions. By means of the *maximum text span token length* parameter, longer text spans can be rejected regardless of their occurrence frequency. This practically eliminates long passages of textual content, the unstructured parts of Web documents. A long paragraph of text is neither a desirable sibling term nor a term at all. In computational linguistics a length of up to 4 tokens (quadgrams) is often set for the maximum length of term expressions to be found or processed. Since the number of terms with more than 5 words length is exceptionally low, we used a maximum text span token length of 5 tokens. Allowing for longer token lengths is relatively computationally cheap while using XTREEM-

SL, whereas for standard n-gram based systems higher numbers of n-grams are more computationally expensive.

Step 7 - Text span Character Length Filtering: Only text spans which have a character length between a minimum and a maximum character length are preserved. For the *minimum text span character length* it makes sense for many languages to require at least a length of two characters to focus on terms. This can be different for other languages where even a single character can be a valid and useful term. For those languages, a minimum text span character length of 1 may be chosen.

Only terms with a length up to *maximum text span character length* are indexed. This parameter is to a certain degree correlated with the filtering performed in step 6 since long passages of text may also contain some whitespaces. This parameter will additionally eliminate long text spans where no whitespace is included, which have passed step 6 erroneously while not being promising term candidates. Practically, a maximum length of up to 50 characters should be sufficient.

Step 8 - Tagpath Cardinality Filtering: Whereas the former filtering steps were applied on single text spans, there is also a filtering according to the number of text spans occurring on a tagpath.

By requiring a *minimum tagpath cardinality* of at least two, text spans which have no sibling text spans are discarded. By means of a *maximum tagpath cardinality*, the size of text span groups to be indexed can be limited. Only groups of a reasonable size can be regarded as useful sibling groups. If there are more text spans with the same tagpath, the tagpath is unlikely to be a good separator among a semantically coherent sibling group. The group is likely to be created by a not well-structured Web document and, therefore, terms will be mixed in an undesired way. Such large groups of sibling text spans can be excluded from the index since they are likely to introduce noise.

10.2.2 Term Retrieval on the XTREEM-SL Index

In section 10.2.1 we described the creation of an index upon the raw siblings found with the Group-By-Path operation. In this section we describe the process to query the index to obtain a list of candidate sibling terms for a given set of input terms. This procedure is depicted in the data flow diagram of figure 10.3.

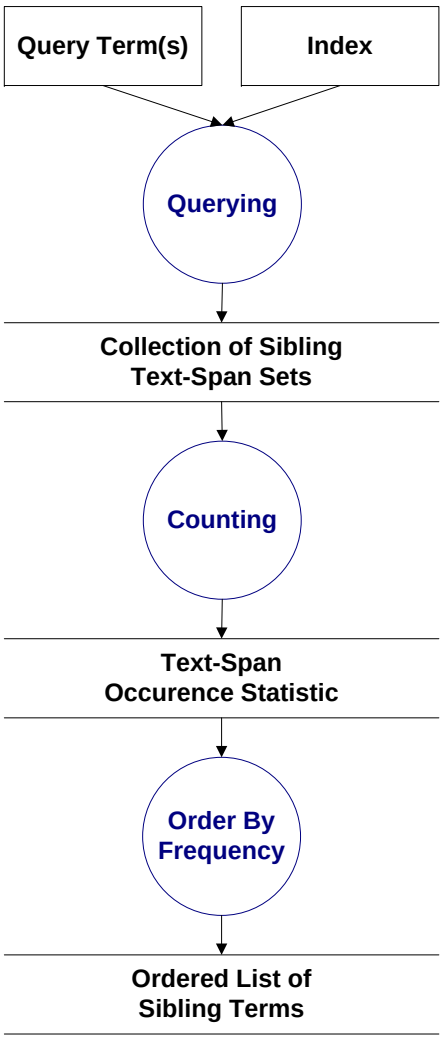


Figure 10.3: Dataflow diagram for retrieving sibling terms from XTREEM-SL

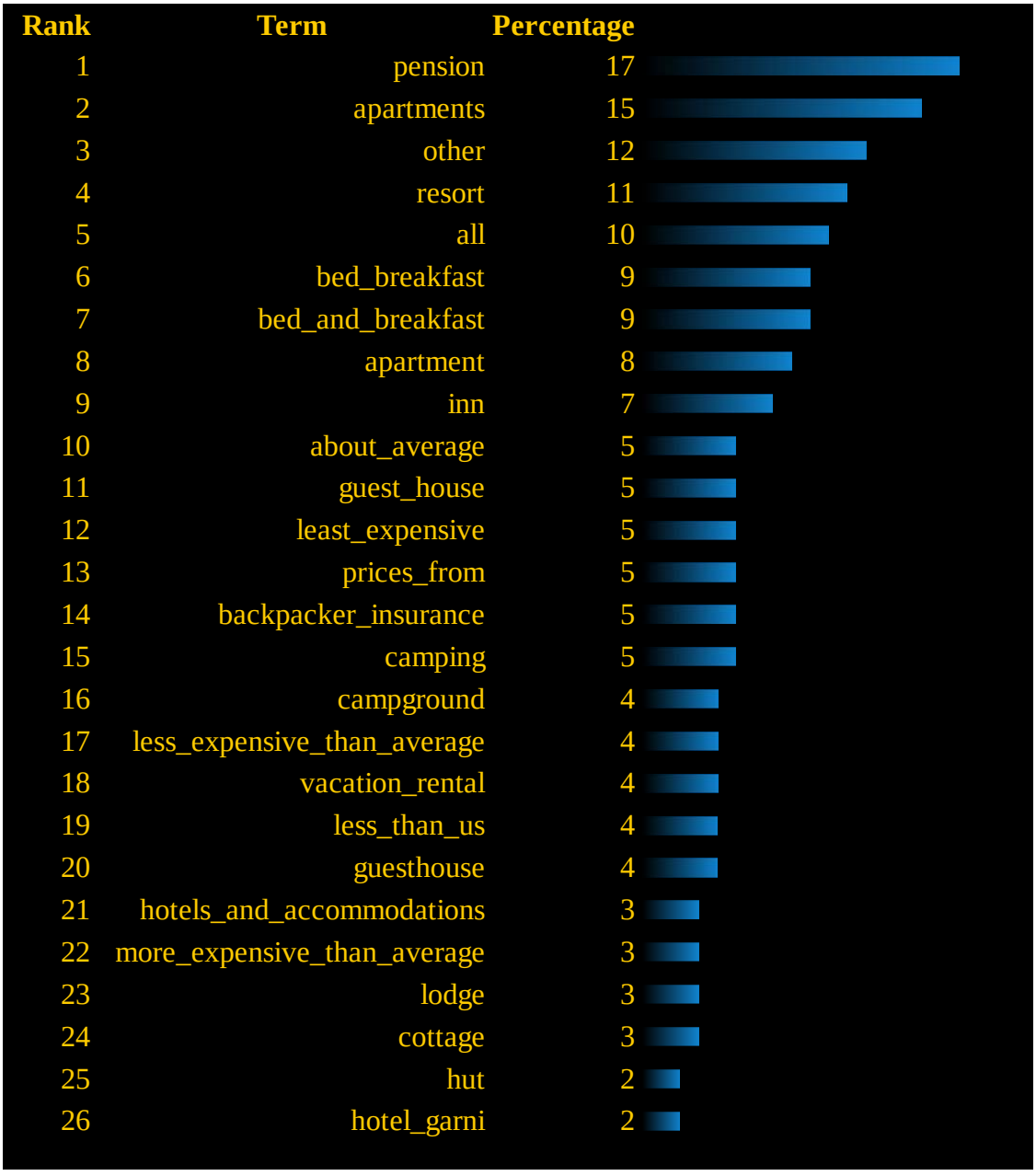
Step 1 - Querying: Input is a set of terms, for which related sibling terms should be found. The terms are combined by AND or OR conjunction to form a query. For these input terms the index returns raw sibling groups, which satisfy the given query. The groups are compared to the query by cosine similarity upon TF×IDF weighted term vectors as provided by default by the used indexing implementation[4]. Vector length normalization is also part of the similarity computation strategy of the incorporated indexing facility.

Step 2 - Text Span Counting: In this step, an occurrence frequency statistic for text spans of the retrieved text span groups is created. In the most straightforward manner, every occurrence of a term in a retrieved sibling group is counted with

[4]http://lucene.apache.org

the same weight. A more sophisticated variant would give a higher weight to an occurrence within a smaller sibling group. By doing so, large, and probably more inhomogeneous groups get a lower weight. It is also possible to discard large sibling groups from the processing if they have not already been filtered out on index creation.

Step 3 - Ordering: According to the occurrence statistic created in step 2, the terms can be ordered. The top-n most frequent text spans can now be presented to the user as a list of candidate siblings. Such an exemplary list of candidate sibling terms is shown in a screenshot of the XTREEM-SL Web user interface depicted in figure 10.4. It shows the presentation of found sibling terms within the Web interface of XTREEM-SL.



Rank	Term	Percentage
1	pension	17
2	apartments	15
3	other	12
4	resort	11
5	all	10
6	bed_breakfast	9
7	bed_and_breakfast	9
8	apartment	8
9	inn	7
10	about_average	5
11	guest_house	5
12	least_expensive	5
13	prices_from	5
14	backpacker_insurance	5
15	camping	5
16	campground	4
17	less_expensive_than_average	4
18	vacation_rental	4
19	less_than_us	4
20	guesthouse	4
21	hotels_and_accommodations	3
22	more_expensive_than_average	3
23	lodge	3
24	cottage	3
25	hut	2
26	hotel_garni	2

Figure 10.4: Retrieval of sibling terms through Web interface of XTREEM-SL. The shown list of terms has been retrieved for the terms “hotel”, “hostel” and “motel”.

10.3 Evaluation Methodology

For the evaluation of discovered sibling relations from a closed vocabulary it was possible to measure group overlap (chapter 4) and recall and precision (chapter 6) by performing a gold standard evaluation. For the evaluation of XTREEM-SL which operates on an open vocabulary, a given gold standard of limited size which is expected not to cover the domains of interest exhaustively is even more problematic. We will also include an evaluation against two existing references, namely the two ontologies from the tourism domain already used for the evaluation in chapter 4 to 6 and described in sections 8.3.2 but also perform experiments where the open vocabulary is not dismissed. For the evaluation against the reference we use the evaluation measure described in the next section. Our first two experiments represent general statistics about text spans occurring on tag-paths.

10.3.1 Evaluation Criteria: Rediscovering Rank

We will evaluate the sibling terms retrieved from the XTREEM-SL by measuring the rank in the ordered results list where a term from the original reference sibling groups can be found again after this term was removed from the group and the group remainder was used for the retrieval of sibling terms.

An ontology provides a set $G = \{H_1, \ldots, H_r\}$ of sibling groups $H = \{T_1, \ldots, T_s\}$ consisting of terms. For a sibling group, one term T_{target} is removed from the group. The remaining group we refer to as Q. All terms from Q are combined as a "`OR`" conjunction into a query expression which is used to query the XTREEM-SL index. Q hereby reflects the notion of contrast sets. The results are obtained and processed as described in section 10.2.2. The result is an ordered list of terms $S = \langle T_1, \ldots, T_u \rangle$. Next we can obtain the rank R on which the term T_{target} occurs within the result. R is the number of terms which a user has to inspect until he has found the term which belongs to the sibling group of the reference. This is done for all $T \in H$ and for all $H \in G$.

It has to be stressed here again that the candidates inspected might contain terms which are indeed plausible siblings, but since they are not supported by the reference, they are counted as if they are wrong.

There are two ways to determine the rank. For the first type, all occurrences of terms within the result are considered. This corresponds to the evaluation according to an open vocabulary. Another way to determine the rediscovering rank is to consider only the terms of the known vocabulary. In experiment 3, both rediscovering ranks will be determined.

In experiment 3 we will investigate the previously described rediscovering rank for all terms from all reference sibling groups. As a result we get an overview of how many result terms have to be inspected to find the reference terms according to sibling relations again.

10.3.2 Evaluation Reference

We use the two ontologies from the tourism domain described in section 8.3.2 as reference.

10.3.3 Inputs

The basis for creating the XTREEM-SL index is the availability of large amounts of Web documents. Normally, the Web documents can be obtained by Web crawling as it is done by internet search engines and briefly described in section 4.1.1. For performing the Web crawl we used the Nutch[5][Cafarella and Cutting, 2004, Rohit Khare, 2004, Cutting, 2005] Web crawler software. Nutch is a freely available Web crawling software which can handle crawls of millions of Web documents. We performed Web crawls of approximately 22 million (21,984,342) Web documents. This accounts for approximately 1/1000 of the total number of documents indexed by the large internet search engines at the time the Web crawl was performed (September 2005 to October 2005). In the following section we describe general measurements on this crawled Web documents whereupon the Group-By-Path operation was performed.

10.3.4 Variations on Procedure and Parameters

Filtering Parameters: For efficiency reasons, a compact index can be created by incorporating thresholds (see section 10.2.1 step 5 to step 8) which eliminate many text spans. These are text spans which would consume much space but which are unlikely to contribute to the results as described in section 10.2.1. Table 10.1 shows the parameters which have been used for the creation of the index used within the experiments. The application of the index creation procedure with those parameters upon the 22 million Web document collection resulted in an index of 109 (108,504,520) million sibling groups. The XTREEM-SL index has a size of 12.6 Gigabytes. The index creation took about 43 hours on a 3.2 GHz single core computer.

Table 10.1: Filtering parameters applied while creating a XTREEM-SL index

	Minimum	Maximum
Text span Token Length	1	5
Text span Character Length	2	50
Tagpath Cardinality	2	50

[5]Nutch: http://lucene.apache.org/nutch/

10.4 Experiments

The first two experiments are general measurements on the Web document collection. The evaluation criteria described in the last section will be applied in experiment 3, described in section 10.4.3. In this experiment we evaluate according to a gold standard reference. In experiment 4 we will determine the occurrence frequencies in combination with the achieved rediscovering ranks. In experiment 5 we will perform an exemplary manual evaluation of the obtained sibling terms. In experiment 6 we will contrast the results obtained by using XTREEM-SL against the results obtained from Google Sets.

10.4.1 Experiment 1: Text span Length

Next we will present general statistics on text spans and tagpaths from a large Web document collection. The numbers are generated for a 10 million (9,673,739) Web document collection.

We measured the number of tokens which constitute the text spans of Web documents. By text span length we refer to the number of whitespace separated tokens contained in text spans. For text spans which are terms, this corresponds to the number of words constituting the (multiword) term. Figure 10.5 shows the occurrence frequencies for varying token numbers (term lengths). The result shows that the data follows a power law distribution.

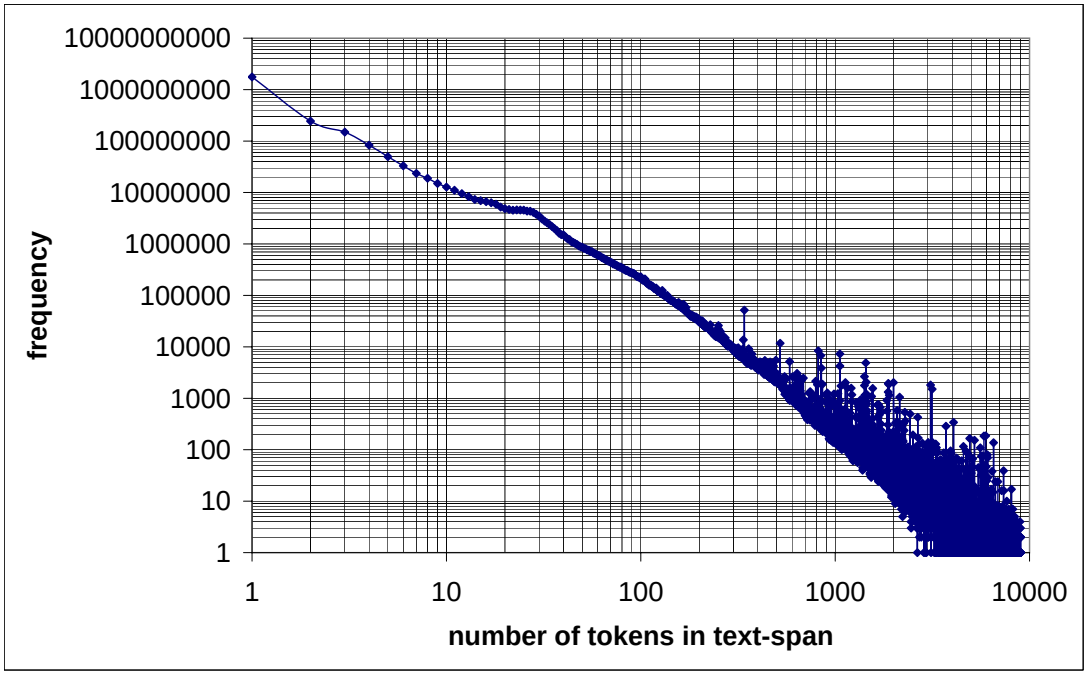


Figure 10.5: Frequency of text spans constituted by varying numbers of tokens (log-log)

10.4.2 Experiment 2: Tagpath Cardinality

In the second experiment we investigated the number of text spans which have the same tagpath in common; or, in other words, how many text spans can be found for specific tagpaths. The results, illustrated in figure 10.6, show that there is a power law distribution. For many tagpaths only a small number of text spans occur. In other words, there are only a small number of text spans which have a tagpath in common. There are only very few tagpaths with a high number of text spans.

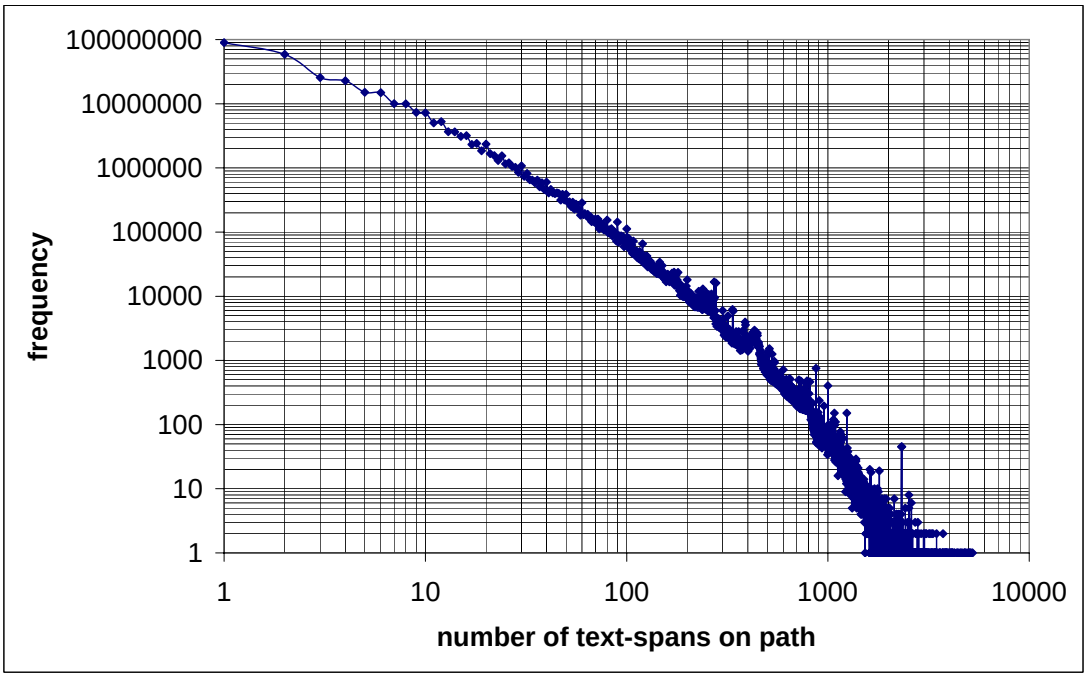


Figure 10.6: Frequency of tagpaths with varying numbers of text spans (log-log)

10.4.3 Experiment 3: A Priory Evaluation

In this experiment we apply the evaluation criteria rediscovering rank described in section 10.3.1. In figure 10.7 (a) and (b) we show the distribution of rediscovering ranks according to the method of determining the rank where only the terms from the closed vocabulary are considered. 18 out of 293 terms for GSO1 and 63 out of 693 terms for GSO2 can be found within the top-10 candidates.

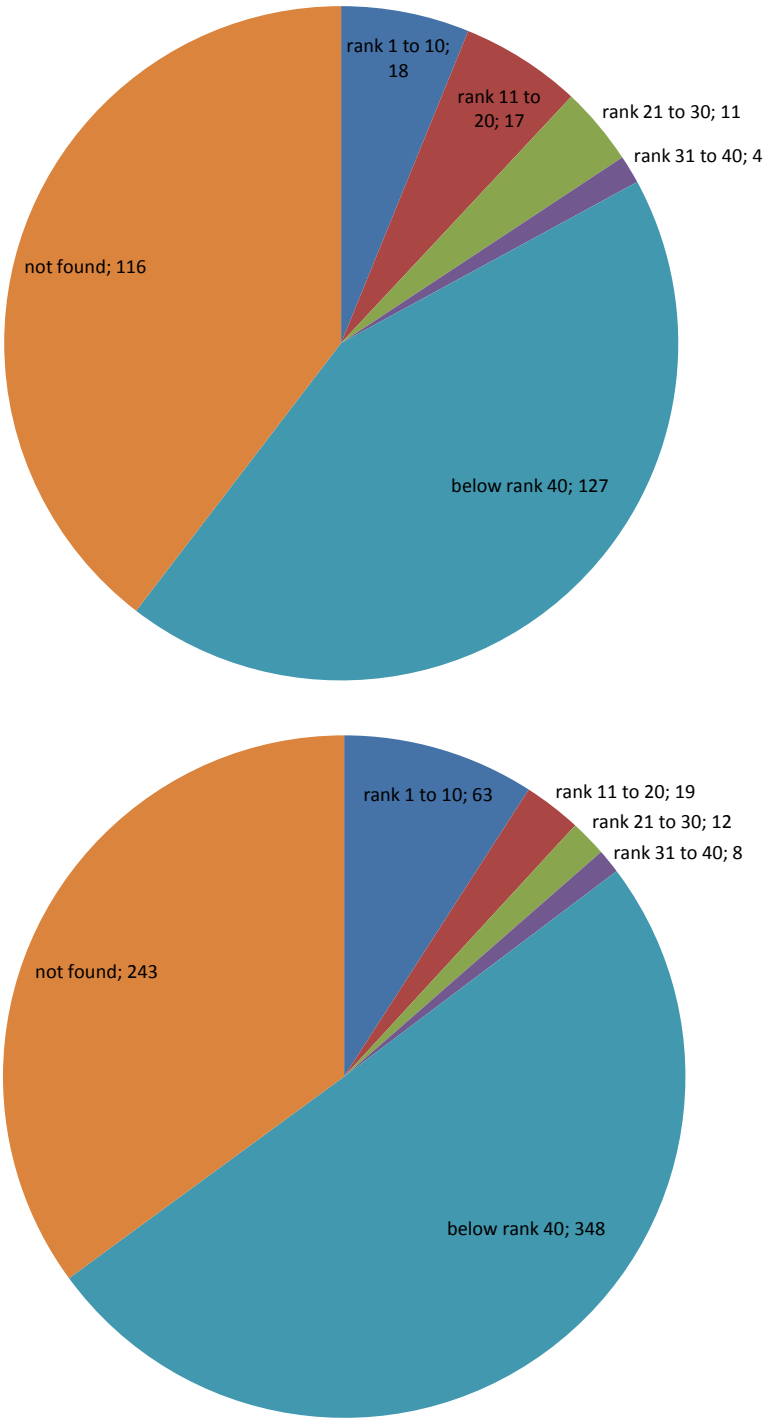


Figure 10.7: Distribution of rediscovering ranks of XTREEM-SL for GSO1 (a) and GSO2 (b)

Conclusion: A fraction of about 10 percent of the terms can be found within the top 10 results of the obtained sibling term list. This observation does not account for the potential fact that while retrieving a particular result term, other terms not counted as appropriate plausible results have also been retrieved. This will be investigated in later experiments.

10.4.4 Experiment 4: Occurrence Frequency

When the results obtained in experiment 3 are studied in isolation, they give the impression that the achieved quality is rather poor. One possible reason that could prevent good results is a low coverage of suitable occurrences within the used Web document collection. For example, the gold standard contains the term `chimney_room` (dt: Kaminzimmer). The number of occurrence within the Web is about 12400[6]. This is a rather small number of occurrences and it is difficult to imagine that such a rare term can be found good, especially within a large, but still limited Web crawl. It is rather exotic for a general tourism ontology. If such concepts are included, the ontology would be rather several thousands concepts large while covering also other concepts of this importance.

In this experiment we will determine if terms which are frequently used can be found better than terms which are rare. For this purpose we determine the occurrence frequency of terms; both within the XTREEM-SL and according to the Yahoo Web search service[7]. The occurrence obtained from the Yahoo Web Search service considers all occurrences, also in regular text, while the number of occurrences obtained from XTREEM-SL only considered occurrences within sibling groups. The frequency scores from both data sources are not directly compared against each other, but both should rather be used as hints for how strongly a term is represented.

In figure 10.8 and 10.9 we show the frequency from Yahoo Web Search service (Web Frequency) and within the XTREEM-SL in combination with the rediscovering rank. Figure 10.8 shows the rediscovering rank according to an open vocabulary, whereas 10.9 shows the ranks in the reference vocabulary.

These figures reveal that for both rediscovering rank types and for GSOs there is the common trend that terms with a high occurrence frequency are found with better rediscovering ranks which means that they can be found more easily. In figure 10.8 for GSO2 in contrast to GSO1 there are more terms that have a rank that is in the top-10 but which are not so frequent.

In figure 10.9, where only the closed vocabulary is considered, many terms are rediscovered within the top-20 to top-50 ranks. This strong improvement of ranks on a closed vocabulary raises the question whether the terms that caused the lower ranks on the open vocabulary are indeed errors or if they are reasonable candidates which are missing in the gold standard ontologies.

[6] measured on 24.05.2008 from google.de

[7] http://developer.yahoo.com/search/web/V1/webSearch.html

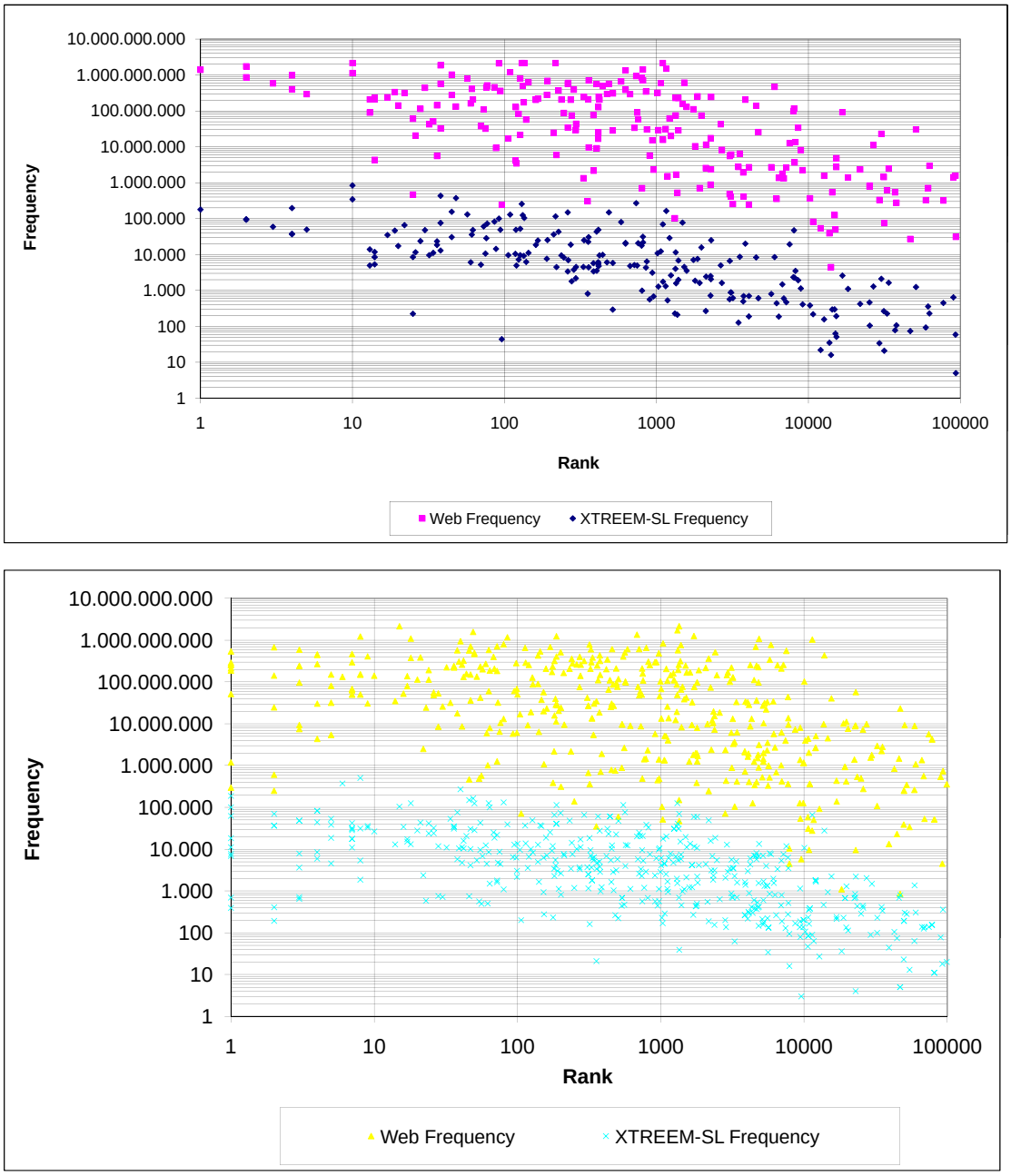


Figure 10.8: Rediscovering rank and occurrence frequency (log) for GSO1 (a) and GSO2 (b), ranks are shown while considering an open vocabulary (also terms NOT present in the GSO's)

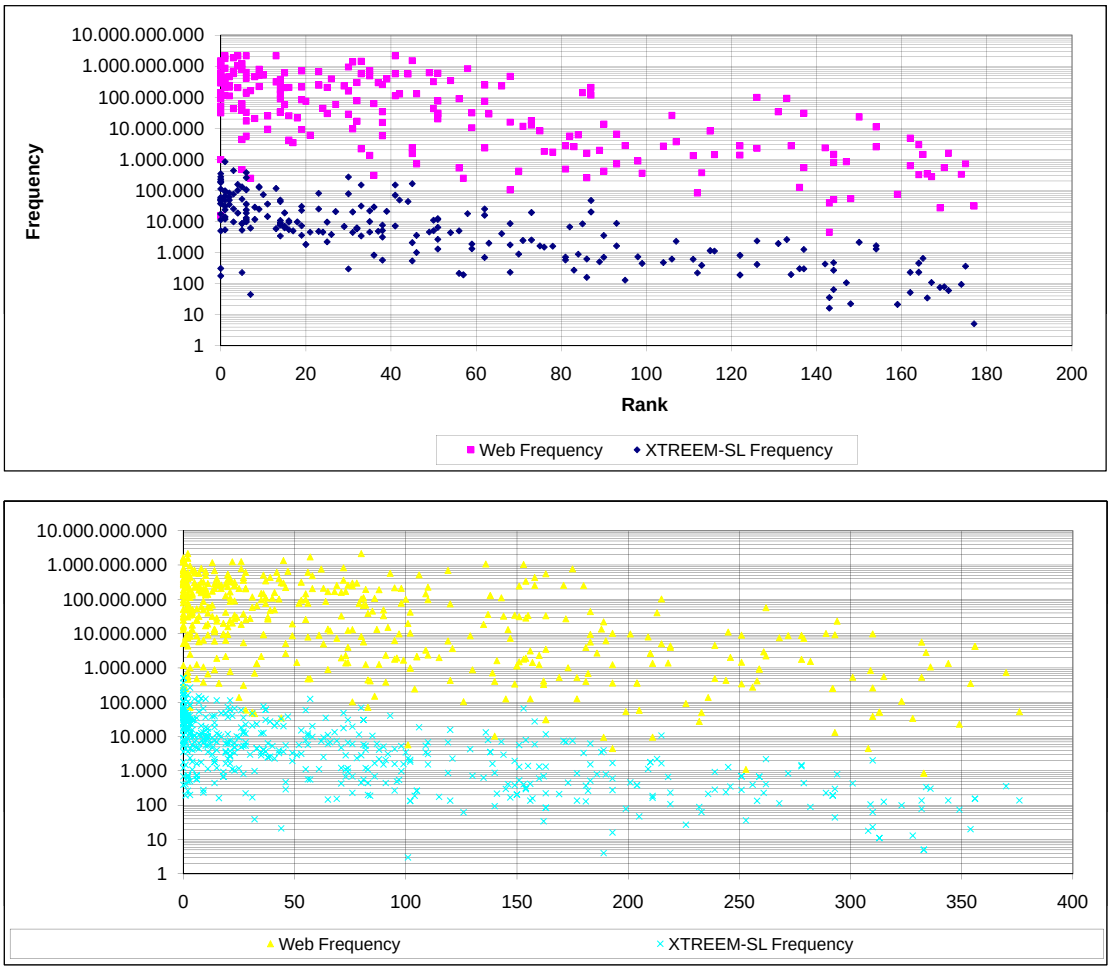


Figure 10.9: Rediscovering rank and occurrence frequency (log) for GSO1 (a) and GSO2 (b), ranks are shown while considering only terms present in the GSO's

The low occurrence frequency of a term which is the label of a concept can be due to several reasons. First, the concept may really be rare. This can occur because the concept is a new topic and, therefore, the number of available Web documents containing that concept in the form of a term is still small. But, on the other hand, there is also the circumstance that the concept label is rather badly chosen. For example, GSO 1 contains the term `bow_shooting_installation`. This term yielded no single hit (on www.google.de, 24.05.2008)) from the Web search engine. The term part `bow_shooting` which is used in the compound term `bow_shooting_installation` yielded 96,300 hits (www.google.de, 24.05.2008). In contrast, `archery`, which is probably what the ontology engineer wanted to express, yielded 16,100,000 hits (www.google.de, 24.05.2008). The occurrence frequencies with several orders of magnitude differences are a possible explanation why some terms are rarely found whereas other terms are found frequently. This is possibly a kind of *translation error*; the ontology engineer has surely the correct concept in mind but expressed this by the wrong label. Practical translation error can be a serious problem since even if ontologies have an English lexicalization; they are likely to have been created by ontology engineers and/or domain experts who are not native English speakers. This can be shortly summarized: strange concept labels cannot be easily (re)learned. For labels of abstract concepts this situation has to be accepted since there are perhaps no or only few occurrences on the Web. But on the other hand, the labels of rather abstract concepts are not what can be learned well by ontology learning. The ontology engineer has to pay special attention to the abstract super-concepts to a higher degree than to the leaf concepts.

But not only translation errors are problematic; also naming inconsistence in written Web texts are a potential source of problems. For the term `base ball` there are 2,670,000 hits (www.google.de, 24.05.2008) whereas for `baseball` there are 217,000,000 hits (www.google.de, 24.05.2008)). Even the less broadly used version has a relatively high occurrence frequency. The most important requirement to cope with this situation is that the labels in the ontology to be used for enhancement (and/or for evaluation) should follow a unique naming style, perhaps always with whitespace separator or even better by incorporating a representation that can copy with lexical diversity [Buitelaar et al., 2009]. Having both types intermixed is problematic since it is likely that the authors of Web documents pursue a consistent naming within their Web documents. Therefore, finding sibling groups where those writing variants are mixed is more difficult than finding sibling groups with a consistent naming.

Conclusion: The general trend in our observations is that frequently occurring terms are found better as sibling terms than infrequently occurring terms. A high frequency does not necessarily lead to good results but a low frequency barely yields good results.

10.4.5 Experiment 5: A Posteriori Evaluation

As we have seen in the previous experiment, the result numbers are not all that good as one would expect. In this experiment we will, therefore, present a sample evaluation where we will also consider the quality of discovered terms by manual inspection. Due to the high manual effort required, this evaluation will not give a quantified number for many examples but we will concentrate on an example.

The reference ontology GSO1 contains the sibling group "`{bicycle, car, bus, ferry, carriage, ship, yacht, boat}`". We will now focus on the term "`bicycle`". While removing "`bicycle`" from the original sibling group we obtain the sibling group "`{car, bus, ferry, carriage, ship, yacht, boat}`". This results in a query: "`car OR bus OR ferry OR carriage OR ship OR yacht OR boat`". This query is issued towards the XTREEM-SL. As a result we obtained the ordered list of terms:

Table 10.2: List of siblings for "`{car, bus, ferry, carriage, ship, yacht, boat}`". ("`bicycle`", the sibling to be re-discovered is found at rank 52.

1. train	27. resources_newsletter
2. air	28. hotels
3. airplane	29. shopping
4. taxi	30. jump
5. home	31. tour
6. hire	32. contact_us
7. rail	33. date
8. helicopter	34. vacation
9. plane	35. auto
10. airport	36. flight
11. welcome_to_nutrition_dome	37. itinerary
12. subway	38. type
13. railway	39. n_a
14. fax	40. road
15. beach	41. base
16. rental	42. site_map
17. boat_directory	43. insurance
18. faq	44. mrs
19. boat_articles	45. contact
20. tours	46. mr
21. cruise	47. airports
22. add_your_site	48. email
23. submit_articles	49. i_do_not_know_yet
24. related_products	50. miss
25. price	51. crew
26. sign_up_for_the	52. bicycle

The above example can be assumed to be a rather optimistic example. Not for all cases can the difference between a priori and a posteriori evaluation be expected to be so significant.

The list depicts the terms from rank 1 to rank 52. Our target term "`bicycle`" was rediscovered at rank 52. This might indeed be an example which provides a positive view upon the results, but this drastic example should give the reader an impression about how the evaluation result numbers might vary from case to case when a human inspection/evaluation is performed. Among the first 13 candidate terms, the 8 terms "`train, airplane, taxi, rail, helicopter, plane, subway, railway`" might be useful enhancements of the given reference sibling group about "`transport_vehicle`". In practice, not all results can be expected to lift so many useful enhancement candidates, but this example shows the weakness of an automatic evaluation according to an incomplete reference. There are likely to be numerous candidates for enhancement of the gold standard ontologies.

10.4.6 Experiment 6: XTREEM-SL in Comparison to Google Sets

In this experiment we contrast the results obtained by Google Sets[8] with the results obtained by term retrieval upon XTREEM-SL. The exemplary results from Google Sets have been obtained at two different points of time, first in October 2006 and later in May 2008. As of October 2009, there is no difference to the May 2008 results. The tables 10.3, 10.4 and 10.5 show the results for both facilities. The query terms of XTREMM-SL have been combined by "`AND`" conjunction. As table 10.3 shows, for the results for `{hotel}` and `{hotel, hostel}` it is rather hard to judge who performs better. In case of `{hotel, hostel, motel}`, Google Sets provides only a few result terms, whereas XTREEM-SL returns a comprehensive list, where even on lower ranks one can find some relevant sibling terms. The first case `{ontologies, taxonomies}` depicted in table 10.4 reveals better results for XTREEM-SL. For example, the terms `thesauri` and `controlled_vocabularies` are not retrieved by Google Sets though they are plausible siblings. For the next two cases `helgoland` and `sylt`, two islands in the North Sea, the results obtained by Google Sets are even worse. In contrast, XTREEM-SL returns many good sibling candidates. Here we have to mention again that the Web crawl was restricted to English documents; the good results of XTREEM-SL come from English Web documents. From this observation we conclude that for non frequent terms (technical terms, proper names, ...), Google Sets performs worse.

[8] http://labs.google.com/sets

Table 10.3: Exemplary results from Google Sets and XTREEM-SL (AND conjunction)

	{hotel}			{hotel, hostel}			{hotel, hostel, motel}		
Google Sets 10/2006	Google Sets 5/2008	XTREEM-SL	Google Sets 10/2006	Google Sets 5/2008	XTREEM-SL	Google Sets 10/2006	Google Sets 5/2008	XTREEM-SL
Hotel	hotel	hostel	Hostel	hotel	hostel	Hostel	hotel	hostel
Retail	hotel accommodation	bed_breakfast	Hotel	hostel	bed_breakfast	Motel	motel	motel
Office	travel hotel	hotels	Bed Breakfast	pension	hotels	Hotel	hostel	bed_breakfast
Residential	inn/lodge 1	prices_from	Motel	camping	prices_from	Apartment	pension	apartment
Restaurant	resort 1	guesthouse	Apartment	motel	guesthouse	Homestay	camping	apartments
Travel	raddison hotel	apartment	b b	jugendherberge	apartment	Backpacker	hostelry	guest_house
Industrial	conference... 2	home_page	Camping	hostelry	home_page	resort and	jugendherberge	hotels
Tour	travelocity hotels	apartment_rental	condominium	youth hostel	apartment_rental	Bed Breakfast	youth hostel	other
AIR	hotels.com official site	country_accommodation	Resorts	kurhotel	country_accommodation	serviced apartment	kurhotel	prices_from
CAR	conference... 3	guest_house	Camp Grounds	bauernhof	guest_house	travel	bauernhof	guesthouse
Transportation	priceline hotel discounts	backpacker_insurance	Trailer parks	appartement ferienwohnung	backpacker_insurance		appartement ferienwohnung	bed_and_breakfast
Research	opryland hotel	camping	Cottage rentals	almhütte berghütte	camping		almhütte berghütte	us
Car Rental	hoteles	us	Homestay	privatvermieter	us		privatvermieter	apartment_rental
Manufacturing	hotels	about_average	Inns	ferienhaus bungalow	about_average		ferienhaus bungalow	home_page
Other	radison hotel	less_expensive_than_average	Backpacker	reisebüro tourismusverband	less_expensive_than_average		reisebüro tourismusverband	resort
Medical Laboratories	choice hotels	least_expensive	Restaurant	sportanbieter	least_expensive		sportanbieter	camping
Investment	luxury hotel	apartments	inn	apartment house	apartments		private accomodation	country_accommodation
Commercial	motel 1	less_than_us	guesthouses	private accomodation	less_than_us		apartment house	about_average
Vacation	all suite 1	rooms	serviced apartment	holiday house	rooms		holiday house	backpacker_insurance
Parking	cheap hotel	other	uk holidays	last minute	other		village tourism	less_expensive_than_average
Multi family	motel 2	home	VILLA	village tourism	home		youth camp	inn
Showers	hotel 4	prague	FARMHOUSE	gasthof	prague		gasthof	least_expensive
Pro Shop	restaurant	more_expensive_than_average	hotels	youth camp	more_expensive_than_average		forest school	prague
TOURS	overnachten	hotel_prague	caravan sites	forest school	hotel_prague		restaurant	all
Accessories	hotel chains	hotels_and_accommodations	Holiday Home	restaurant	hotels_and_accommodations		ferienwohnung	pension
Rail	extended stay 1	double	resort and	ferienwohnung	double		camping, campingplatz	restaurant
Multifamily	hotel 5	inn	campsites	appartement	inn		appartement	more_expensive_than_average
Luggage	hotel 8	farmhouse_b_b	travel	camping, campingplatz	farmhouse_b_b		last minute	less_than_us
Community Development	hotel 2	family	Selfcatering Accommodation	holidays	family		zimmer frei	category
CRUISES	hotel 7	suites	CASOLARE	zimmer frei	suites		apartmenthaus	home
Apartments	hotel search	pub_inn	Campground	apartmenthaus	pub_inn		spa hotel	hotels_and_accommodations
Educational	all suite 2	bed_and_breakfast	dwelling	spa hotel	bed_and_breakfast		farm	location
Warehouse	hotel 6	twin	student lodging	farm	twin		bungalowpark	suites
Service	hotel 10	hostels_in_prague	quartering	bungalowpark	hostels_in_prague		travel agency	hotel_prague
Institutional	apt/condo 1	hotels_in_prague	block	travel agency	hotels_in_prague		hunting lodge	super_motel
Cruise	hotel 3	eurotel	quarters	dorftourismus	eurotel		dorftourismus	double
Retail Fixtures	travelzoo® hotel deals	pension	billet	hunting lodge	pension		sports supplier	twin
Industrial properties	london hotel	cheap_prague_hotels_pensions	home	holiday apartment	cheap_prague_hotels_pensions		private rental	house
hospital	hotel 1	hotels_pensions	camp	mountainhut shelter	hotels_pensions		mountainhut shelter	farmhouse_b_b
Laboratory	cabin 1	individual_city_tours	domicile	private rental	individual_city_tours		holiday apartment	family
Automotive	hollandhotels	prague_apartment_rentals	rattrap	sports supplier	prague_apartment_rentals		schutzhaus, jugendherberge	pub_inn
Placement Firms	other 1	oskar_vodafone	abode	schutzhaus, jugendherberge	oskar_vodafone		bungalow	stars
Lockers	fietsen	reservations	youth hostel	bungalow	reservations		jugendlager	hostels_in_prague
Publishing		self_catering	hospice	jugendlager	self_catering		resort	swimming_pool
Self storage		town	flat	resort	town		waldschule	b_b
Financial Services		to_entries_listed_below	habitation	waldschule	to_entries_listed_below			land
		location			location			industrial_buildings
		you_can			you_can			contact_us
		please_see			please_see			hotels_in_prague

Table 10.4: Exemplary results from Google Sets and XTREEM-SL (AND conjunction)

{ontologies, taxonomies}			{helgoland}				{sylt}		
Google Sets 10/2006	Google Sets 5/2008	XTREEM-SL	Google Sets 10/2006	Google Sets 9/2008	XTREEM-SL	Google Sets 10/2006	Google Sets 5/2008	XTREEM-SL
Ontologies	taxonomies	taxonomies	Helgoland	helgoland	horumersiel	Sylt	sylt	amrum
Taxonomies	ontologies	thesauri	Oman	malta	wangerland_minsen	Buhne	berlin	f_hr
Systems	html	metadata	Tanganyika	gibraltar	tossens		hamburg	north_frisian_islands
Research Groups	xml	content_development	Tanzania	cyprus	st_peter_ording		bayern	east_frisian_islands
People	pdf	metamodels	Historical Flags	deutschland	aurich		hessen	kachelotplate
Search Engines	rtf	terminology_extraction	Nordfriesland	portugal	sylt		brandenburg	borkum
Class interfaces	text	concept_systems	Holsteinische Schweiz	halstenbek	norden		bremen	pellworm
Problem solving methods	xhtml	methodology_standardization	Probstei	rellingen	wilhelmshaven		niedersachsen	terschelling
Data Model Mapping	excel	controlled_vocabularies	Fehmarn	greece	bad_zwischenahn		baden württemberg	nordstrand_germany
Ontological Modelling	sgml	other_subject_based_techniques	Tafellied	egypt	wittmund		sachsen	l_beck
agents	csv	faceted_classification	Aquarius	lebanon	westerstede		schleswig holstein	neuwerk
KIF	postscript	what_is_metadata	Westfalen	liechtenstein	leer		rheinland pfalz	schiermonnikoog
Domain knowledge static	word	metadata_as_a_finding_aid		spain	juist		saarland	texel
Reasoning knowledge dynamic	css	subjects_and_precision		andorra	nieuweschans		thüringen	vlieland
Question answer corpora	svg	the_names_of_subjects		luxembourg	stadskanaal		sachsen anhalt	wieringen
Electronic dictionaries	ascii	occurrences		italy	info_motylek_com		nordrhein westfalen	west_frisian_islands
Knowledge Management	txt	types		switzerland	spiekeroog		mecklenburg vorpommern	ameland
RDF	xslt	associations		guernsey	norderham			kiel
Semiotic Modelling	latex	benefits_and_costs		pinneberg	cuxhaven			rottumeroog
Introduction	flash	identity_and_merging		jersey	wangerooge			rottumerplaat
FIPA agent standards	folksonomy	searching		ireland	husum			hiddensee
Semantic Web	javascript	schemas		germany	bremerhaven			fehmarn
Knowledge Representation	doc	owl		turkey	sylt_keitum			helgoland
Argumentation	wml	formal_is_a_relationships		iceland	bus_charter			flensburg
Hypotheses	web	value_restrictions		tornesch	vila_dos_remedios			heligoland
Variables	dom	pdf		bahamas	h_li_ni			bad_schwartau
The Research Question	xls	disjointness_inverse_part_of		denmark	schleswig			lubeck
Validity and Reliability	sylk	xml_schemas		france	kashi			lindau
Levels of measurement	dyomedea	xml		belize	stanley			ruden
	rdf	informal_is_a_relationships		israel	jalalabad			hamburg
	web2.0	formal_instances		monaco	heligoland			germany
	ps	microsoft_excel		croatia	roter_sand			oland
	sql	frames_properties		appen	port_blair			vilm
	http	word_processing_documents		romania	friedrichstadt			r_gen
	dtd	relational_databases		panama	braunschweig			nordstrand
	tagging	general_logical_constraints		oldenburg	brunsbuettel_st_michaelisdonn			poel
	software	html		sweden	linie			home
	pcl	classification_schemes		australia	ushuaia			greifswalder_oie
	tiff	faceted_metadata			san_andres			usedom
		schema			robinson_crusoe			reeperbahn
		neuralnetworks			bremen_farge			schleswig
		informationretrieval			damp			mallorca
		som			hamburg			eckernf_rde
		ma			langeoog			neumuenster
		topic_maps			germany			rimini
		en			settlement			nebel_amrum
		learning			hahnenklee_bockswiese			uetersen
		hierarchies			lun			toscana_punta_ala
		nlp			airport_express			saxony

Table 10.5: Exemplary results from Google Sets and XTREEM-SL (AND conjunction)

{syft}					{shark diving}				{whale watching}		
Google Sets 10/2006	Google Sets 5/2008	XTREEM-SL		Google Sets 10/2006	Google Sets 5/2008	XTREEM-SL	Google Sets 10/2006	Google Sets 5/2008	XTREEM-SL
Sylt	syft	amrum		shark diving	shark diving	whale_watching	Whale Watching	whale watching	fraser_island
Buhne	berlin	f_hr		Rafting	scuba diving	cape_town	Navatek Cruises	sight seeing	hervey_bay
	hamburg	north_frisian_islands		Rock Climbing	skydiving	scuba_diving	USS Arizona Memorial	hiking	great_barrier_reef
	bayern	east_frisian_islands		Sea Kayaking	paragliding	quad_biking	Sea Life Park	bird watching	monterey_movie_tours
	hessen	kachelotplate		Bungee Jumping	bungee jumping	klooting	Hawaii Vacation Rentals	beachcombing	worlds_best_k
	brandenburg	borkum		Mountaineering	surfing	surfing	Waimea Falls	jet skiing	light_tackle_fishing
	bremen	pellworm		Abseil Africa	hang gliding	hot_air_ballooning		deepsea fishing	fantasy_trail_rides
	niedersachsen	terschelling		wreck diving	ballooning	bungee_jumping		snorkeling	accommodation
	baden württemberg	nordstrand_germany		Sailing	mountaineering	water_skiing		outlet shopping	zodiac_charters
	sachsen	l_beck		Caving	canyoning	township_tours		pier fishing	horseback_riding
	schleswig holstein	neuwerk		Horse Riding	parachuting	sandboarding		live theater	skiing
	rheinland pfalz	schiermonnikoog		Cycling	scuba	skydiving		surf fishing	sailing_cruises
	saarland	texel			diving	microlighting		freshwater fishing	zoos
	thüringen	vlieland			gliding	helicopter_tours		sound/bay fishing	victoria_falls
	sachsen anhalt	wieringen			zorbing	table_mountain		fly fishing	bird_watching
	nordrhein westfalen	west_frisian_islands			aerobatics	kayaking_canoeing		paddle boating	cruise_guide
	mecklenburg vorpommern	ameland			flying disc	abseiling_rock_climbing		water tubing	legoland_california
		kiel			aeromodelling	robben_island		mountain climbing	universal_studios
		rottumeroog			casting	horse_riding		gambling casinos	fishing
		rottumerplaat			orienteering	exclusive_shopping		rafting	cape_big_six_experience
		hiddensee			sport fishing	wine_tasting		ice skating	package_tours
		fehmarn			sailing	cape_peninsula		sledding	great_white_shark
		helgoland			caving	the_franschhoek		whitewater rafting	more
		flensburg			rock climbing	the_long_beach		luaus	travel_tips
		heiligoland			diving vacation	the_bishops_court		spelunking	ensenada_fishing
		bad_schwartau			whale watching	the_constantia			sea_kayaking
		lubeck			windsurfing	cape_winelands			wd_hire
		lindau			mountain biking	cape_town_beaches			home
		ruden			kayaking	golf_courses			host_u
		hamburg			padi	helicopter_trips			fishing_charters
		germany			surf	home			harbor_dinner
		oland			water skiing	water_activities			windjammers
		vlm			felt running	topless_bus_tour			puffin_tours
		r_gen			fishing	sunset_cruise			map_of_south_africa
		nordstrand			snowboarding	mt_kilimanjaro_tour			golf_courses
		poel			water sport	home_page			extend_this_holiday
		home			sailing vacation	fishing			russian_tours
		greifswalder_oie			sailing school	experience_sa_network			sea_kayaking_mountain_biking
		usedom			scuba lesson	the_vintage_hotel			diving
		reeperbahn			surf shop	contact_us			lunenburg_fisheries_museum
		schleswig			scuba gear	why_we_are_different			bluenose_golf_course
		mallorca			sailing lesson	the_vintage			whitsundays
		eckernf_rde			ocean sailing	diving			asa_sailing_instruction
		neumuenster			climbing	photo_gallery			beaches
		rimini			horse riding	snorkeling			whitewater_rafting
		nebel_amrum			hiking	free_brochure			kayaking
		uetersen				rates_packages			romantic_packages
		toscana_punta_ala				private_pilots			dolphins
		saxony				fishing_map			lunenburg_academy

Conclusion: The term sets retrieved by our approach can be regarded as having a stronger semantic coherence, with regard to being semantic siblings, than those obtained by Google Sets. Our approach works also well for rather infrequent domain specific terms where Google Sets performs weaker. This is an important observation, since engineering domain ontologies operate on rather infrequent terms. Therefore, XTREEM-SL can be regarded as being better suited for this purpose; using semantically founded term retrieval for ontology learning is enabled by XTREEM-SL, doing so with Google Sets seems not feasible.

10.5 Conclusion

In this chapter we described an approach for obtaining sibling terms within an open vocabulary. We showed an evaluation according to reference ontologies, and exemplary evaluations. While the measured quality according to the rediscovering ranks measures yielded not good results in general, the manual inspection revealed that the result contains a considerable number of plausible sibling terms which are not present in the gold standard ontologies.

We have performed experiments on a data set of millions of documents. For indexes covering bigger parts of the Web our method can be expected to scale well. The process can be made parallel easily.

11 Conclusions and Outlook

In this thesis we contribute to the state-of-the art in ontology learning by presenting approaches for acquiring terms, synonyms and with emphasis sibling relations from large collections of Web documents. For this purpose we rely on the added value provided by Web documents in contrast to plain text, the mark-up. While extracting knowledge the emphasis is on extracting sibling relations. Sibling relations are orthogonal to the direct hierarchical relations of sub-ordination. The knowledge about concepts (terms depicting concepts) standing in sibling relation can be used while grouping the concepts beyond appropriate super-concepts. Furthermore, for an existing concept, sibling concept candidates can be assigned as sub-concepts to the parent/super concept thus enhancing the given conceptualization in sibling direction.

11.1 Main Contributions

The main principle which is used for the extraction of knowledge from Web documents in this thesis is the Group-By-Path appproach. According to the Group-By-Path-approach text spans created by the semi-structure of Web documents are grouped according to structural regularities, the so called tagpaths. Tagpaths are the sequence of tags which lead to the text-spans within the Web document tree structure. By applying the Group-By-Path-approach, text spans which are sibling due to the structural regularity can be extracted.

Those extracted sets of sibling text spans are then further processed to obtain different types of results. While doing those processing, sibling relations which are not frequently found are dimmed. With XTREEM-SG (chapter 4) we apply flat clustering to obtain a collection of groups of terms which are supposed to stand in sibling relation. In our evaluation experiments we investigated to which extent the obtained sibling groups overlap with sibling groups from gold standard ontologies. The results yielded that XTREEM-SG is capable of obtaining sibling groups with better quality than in the previously published methods.

With XTREEM-SGH (chapter 5) we applied hierarchical clustering to obtain hierarchies of sibling groups. By doing so not only a flat collection of sibling groups has to be inspected which is generated once, but a sibling groups hierarchy where the desired granularity among the results can be varied while inspecting the results. The findings were that this can be done with an equal or slightly worse quality than XTREEM-SG. Nevertheless, hierarchically arranged sibling groups are a considerable option in combination with an appropriate user interface.

Furthermore, we performed the computation of binary sibling relations by computing associations with XTREMM-SP (chapter 6). This has the advantage that the computation of sibling associations in the same dimensionality for a large number of instances is less complex and, therefore, such results can be yielded more quickly, which can be an important issue since then the results can be presented more quickly to the ontology engineer. The time advantage is accompanied by the drawback that binary sibling pairs are less valuable for modelling ontologies than larger groups of terms standing in sibling relation.

We extended the computation sibling relations to the computation of synonymous terms by applying XTREEM-S (chapter 8). XTREEM-S relies on associations computed from sibling associations. The finding was that only a small fraction of to be found synonym relations could be extracted. In isolation, computing synonyms is not promising, but since in combination with acquiring sibling relations it is a viable add on, since term pairs obtained with XTREEM-S which are not synonyms are likely to be plausible sibling terms.

But not only sibling or synonym relations can be acquired from Web documents, the Web document markup is also beneficial for acquiring vocabularies of terms. Vocabularies of terms are by itself a valuable resource for many purposes besides performing ontology learning. The finding for XTREEM-T (chapter 7) was that approximately half of the ranked result terms are terms which are of probable interest for the domain in question. This observation has also to be seen in conjunction with applying approaches such as XTREEM-SG or XTREEM-SP without manually crafted vocabularies. While performing automatic feature space building a similar quality regarding the extracted text spans corresponding to the terms of the domain can be expected.

While we extract terms, we are able to extract single word and multiword terms at the same time; only few approaches are known to be capable to do so. The advantage of the methods for acquiring vocabularies is that it is language and domain independent. No linguistic resources are required, nor is training required. This is a fact that holds true for all approaches presented in this thesis, they are language and domain independent and are capable of acquiring and processing multiword terms.

In chapter 9 we presented the domain relevance enhanced term weighting. The domain relevance enhanced term weighting combines two widespread methods for weighting terms into one. It is supposed to be beneficial for clustering Group-By-Path data obtained while using a feature space which was not manually cleaned. Domain relevance enhanced term weighting favours domain relevant terms for creating clusters.

Last but not least, we presented the XTREEM-SL approach (chapter 10) for creating an index over large amounts of sibling groups extracted with Group-By-Path from large numbers of Web documents. Obtaining siblings with the XTREEM-SL approach enables the acquisition of terms from an open vocabulary without the need to restrict the number of considered terms. Querying this index allows the retrieval of lists of sibling terms. While the creation of the index is a

time consuming process, attaining the result siblings can be done quickly, enabling the presentation of results in an ad-hoc time frame. As such XTREEM-SL is an approach which is truly web-scale. But the usage of XTREEM-SL is not restricted to acquiring new sibling relations, XTREEM-SL can be used to assess the plausibility of existing conceptualization regarding sibling relations.

In summary, this thesis contributes to the ontology learning field by providing several approaches which use Web documents for learning. Herby the added value of Web documents, the markup, is compared to plain text, not removed but used as the actual source of knowledge. Since the presented techniques rely on publicly available Web documents which can be obtained automatically, the ontology engineer can be freed from manually assembling a document collection. Actually, only a small fraction of the information provided by Web documents is used but this is not so much a waste since the Web provides huge amounts of documents available for processing. The relations we acquire are only acquired if they are prevalent among large amounts of Web document markup. The results can subsequently be regarded as shared more likely than for patterns obtained from a manually crafted text document collection of much smaller size. This can be regarded as highly desirable for ontology learning since ontologies are supposed to be shared conceptualizations. Shared mark-up is used to facilitate shared conceptualizations. The Word Wide Web is used for ontology learning to facilitate a potential future Giant Global Graph [Berners-Lee, 2007]. The Web proofed beneficial for performing task otherwise hard to accomplish with plain text and natural language processing [Ravichandran et al., 2004, Lüdeling et al., 2007, Halevy et al., 2009].

11.2 Future Work

Tagpath Constitution: The foundational principle which enabled the grouping of text spans is that the paths can distinguish the sibling textspans. More and more of the structuring within Web documents is done with style sheets [Bos et al., 2007] and RDFa [Group, 2008]. To capture such information also while keeping the approach straightforward without introducing too many heuristics, the class attributes used to state style sheet classes could be added to the tagpath. The class attribute which is a further specification of the tag would then not be ignored.

Clustering: Since term clustering yielded the best results, it is desirable to incorporate a clustering algorithm with possible multi cluster membership to overcome a weakness of term clustering that terms can belong to only one cluster which is a hard limitation that is to be avoided. Here a term should be able to belong truly to many clusters, also referred to as non exclusive clustering. While allowing terms to belong to more than one sibling group it can account for the circumstance that there are concepts that can have multiple super-concepts and that then might have multiple sibling context and for homoymy/polysemy

[Cicurel et al., 2006] of terms. There are several clustering techniques with soft cluster membership such as Clustering by Committee [Pantel, 2003], Overlapping Pole-Based Clustering [Cleuziou et al., 2004] or Overlapping k-means [Cleuziou, 2007] which are considerable candidates.

While performing tagpath clustering, feature space reduction techniques such as principal component analysis [Joliffe, 1986] might be invoked to reduce dimensionality, especially while using bigger feature spaces of several thousand dimensions.

Processing Alternatives: The frequent itemsets alone yielded worse quality than just clustering. It might be interesting to investigate the merit of the approach of Ester et al [Beil et al., 2002, Fung et al., 2003] which is given by a process where first frequent item sets are discovered which are then clustered. Such a process has the benefit that the clustering can be performed on a smaller data set which will give a better overall time complexity. Also the application of Latent Semantic Analysis (LSA) [Deerwester et al., 1990] should be tested.

Domain Relevance enhanced Term Weighting: Applying and evaluating domain relevance term weighting on different data sets such as regular text data sets.

Ontology Engineering – Coordination Direction: Not much attention is paid to sibling relations within ontology engineering methods - though as the practically oriented tutorials suggest, it is useful. Since the methods described in this thesis allow for a relatively easy acquisition of sibling relations, this should be more explicitly regarded in ontology engineering and maintenance processes.

Integration of different results: Since there are the types of ontological knowledge which are obtained for ontology learning it is desirable to combine the different evidences of such approaches as for example done in [Cimiano et al., 2004c, David Manzano-Macho and Borrajo, 2008]. An important goal is to obtain candidate names for the parent concept of a sibling group as well. One possibility is to use one of the approaches which learn pairs of super-concept and sub-concepts. For all terms of a sibling group candidates of parent concepts can be obtained where a decision for common parent-concepts appropriate for all siblings has to be taken.

But another possibility is to acquire collocations which are characteristic of the terms of a sibling group in combination. Those terms which are characteristically used in the context of the terms of the sibling groups can contain the appropriate parent concept or they might give valuable clues. For example, for the terms “white shark” and “tiger shark” there will be terms such as “dangerous” or “fish” or “predator” as characteristic “common collocations”. But also interesting is the acquisition of the “attributes” which distinguish the sibling terms among each other. To do so, one could obtain collocations and contrast them so that the terms which

are characteristic for one sibling compared to another can be obtained. For example, for the concept "river", it could be possible to obtain terms such as "flow velocity" or "water quality". In contrast, for "mountain" there are likely collocations such as "high" or "minerals".

Tool Integration: Practically the engineering of ontologies is usually performed with the support of ontology learning environments such as Protégé, OntoStudio or TopBraid Composer. Making those methods available within such tools is a requirement for the practical applicability.

Evaluating Ontologies against Sibling Terms from the Web: This is not unrelated to the possibility of using an approach like XTREEM-SL for assessing the plausibility of existing ontologies. Even thought the acquired results are far from perfect, an inspection of existing ontologies regarding the support with sibling relations on the Web can reveal insights.

Incremental Application We had two extreme positions while acquiring siblings, on the one side, the entire vocabulary was used, on the other side only one term or one sibling group was used. A solution in between those extremes might be considered where a particular fraction of an ontology is to be learned. For practical applicability it is highly desirable to start with a small seed of one or more terms. For this terms matching Web documents should be obtained. From the Web documents sibling relations and/or a vocabulary should be extracted. The newly obtained terms should be used to retrieve more Web documents. The Web document collection, can therefore, be built up easily and the fit of Web documents to be learned ontology is ensured.

Application for learning and enhancing personal information models: The learning of ontologies is an ambitious goal. For learning more complex ontological constructs, appropriate methods for acquiring rules and axioms [Völker et al., 2007] are desirable. And indeed the realization of the semantic Web is still pending, but there is also demand for the acquisition of less rigorous formal knowledge as those to be captured in personal information models [Sauermann et al., 2007] envisioned for the users (social) semantic desktop [Ansgar Bernardi, 2008]. For the acquisition of sibling relations with XTREEM methods, one could start by using the user's bookmarks as seed for Web crawls. These bookmarks should already reflect the user's interests. The concepts represented in the users PIMO could be used to crawl Web documents by incorporating Web search engines as well. The automatic acquisition of terms and sibling relations can greatly facilitate the realization and maintenance of the users personal information model where a manual creation and maintenance can be expensive too, analogous to ontologies. The XTREEM methods are especially suited since they can acquire the necessary input data automatically from the Web and the amount of ontology learning

process knowledge in the form of to be applied parameters is rather small for the XTREEM methods compared to other methods. Applying ontology learning towards areas of personal interest can also be an mean to bring the user to provide the still required supervision, which for ontology learning is often hard to achieve [Siorpaes and Hepp, 2007, Siorpaes and Hepp, 2008].

A Exemplary Ontology Structure

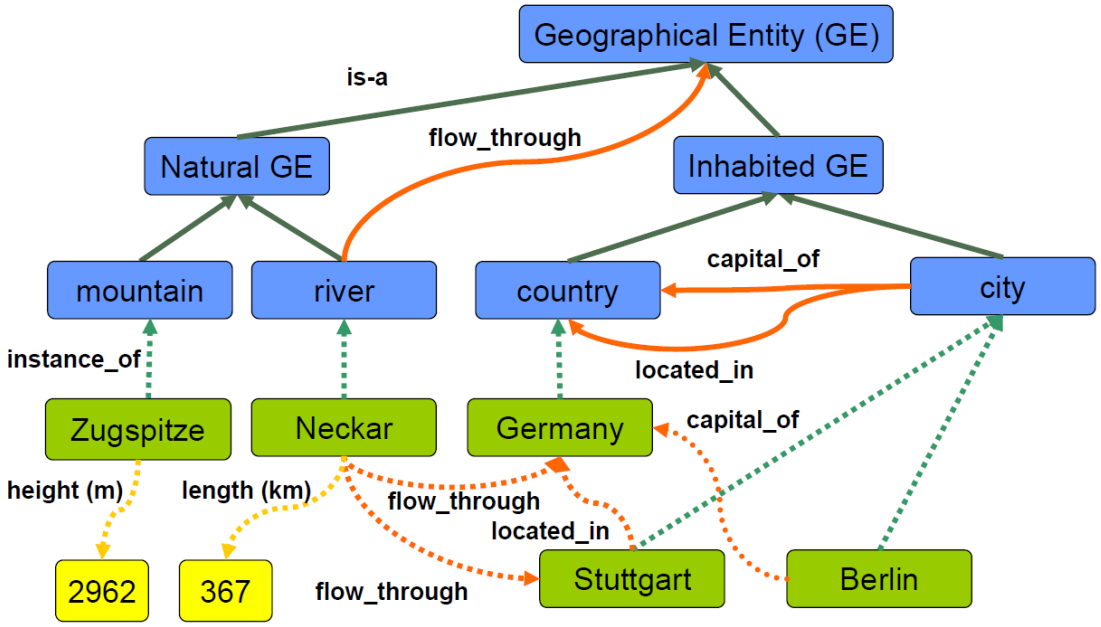


Figure A.1: Ontology from geography domain [Buitelaar and Cimiano, 2007]

B Reference Sibling Groups from Gold Standard Ontologies

balcony	address	beach	badminton_court
bed	brochure	city	basketball_ground
drier	category	country	beach_volleyball_field
electronic_device	conference_folder	local_recreation_area	bow_shooting_installation
kitchenette	date	nature_reserve	bowling_alley
minibar	email	place	crazy_golf_course
shower	house_description	region	cross_country_ski_run
terrace	route_description	ski_run	football_field
whirlpool	adventure_holiday	town	golf_course
daytime	camping	walking_trail	gym
holiday_time	cure	fitness_course	indoor_swimming_pool
night	journey	football_game	shooting_gallery
period	short_trip	art_exhibition	skittle_alley
public_holiday	sport_holiday	cabaret	sports_hall
starting_point	ausblick	concert	squash_court
accommodation	beach_view	musical	swimming_pool
inn	panorama	presentation	tennis_court
service	sea_view	theatre	trimmdichpfad
touristic_installation	booking	town_sightseeing_tour	volleyball_field
rowing_boat	excursion	living_thing	volleyball_ground
sailing_boat	sport	organization	weights_room
first_class_hotel	educational_journey	animation	hair_dryer
middle_class_hotel	pilgrimage	fango	iron
heritage_town	bicycle	hair_dresser	radio
port	boat	massage	telephone
driving_license	bus	moor_therapy	tv
id_card	car	mud_therapy	videorecorder
aerobic	carriage	shop	washing_machine
badminton	ferry	shuttle_service	cinema
billiard	ship	appartement	concert_house
golf	yacht	beauty_farm	exhibition
kayak	bill	castle	gallery
riding_crop	cheque	club	museum
swimming	contract	farm	musical_theatre
tennis	currency	health_club	open_air_theatre
agreement	law	holiday_appartment	opera_house
cultural_installation	passport	holiday_village	theatre_house
culture	price_list	hotel	banquet
event	visum	motel	buffet
group	kiosk	pension	main_meal
information	rental	sanatorium	menu
money	sport_shop	seminar_house	ballroom
time	masseur	tent	barbecue_area
view	tour_operator	youth_hostel	casino
holiday_equipment	tourist	caravan	diving_station
room_equipment	bar	transport_vehicle	fitness_studio
equipment	billiard_room	accommodation_equipment	ice_hall
meal	conference_room	non_material_thing	library
personal_thing	disco	qualitative_time_concept	moor_bath
recreational_installation	elevator	situation	park
room	fitness_room	spatial_concept	playground
sight	library	thing	promenade
sport_equipment	parking_lot	non_private_accommodation_equipment	sauna
vehicle	recreational_installation	room_equipment	shopping_center
ausblickturm	solarium	beer_garden	sport_installation
basilica	steam_bath	cafe	squash_field
castle	turkish_bath	casino	sun_studio
city_wall	wellness_installation	jazz_club	thermal_bath
ruin	ball	lounge	thermal_spring
afternoon	climbing_wall	night_cafe	bull_fight
autumn	fishing_equipment	pub	business_event
day	football	wine_tavern	cultural_event
early_season	sledge	city_port	day_tour
morning	table_tennis_table	yacht_port	day_trip
off_season	conference	airport	holiday
spring	congress	bank	human_activity
summer	presentation	change_office	party
time_interval	seminar	harbour	sport_event
winter	animal	kindergarden	trip
chimney_room	person	station	wedding
single_room	plant	tourism_center	

Figure B.1: Sibling groups from GSO1

countable_concept	horse_riding_lessons	aerobic	barbecue	beauty_day
intangible	tennis_lessons	badminton	wedding	beauty_relax_weekend
mass_concept	back_massage	balloon_trip	dancing_night	christmas_special
spatial_concept	body_massage	basketball	dancing_tee	cultural_offerings
thing	face_massage	billiard	bicycle_tour	event_offerings
event	beach_chair_rental	bowling	canoe_tour	family_celebration
time	bicycle_renting_agency	bungee_jumping	charter_excursion	flat_offer
view	boat_rental	climbing	cycling_tour	holiday_special
action	car_rental	cycling	day_trip	horse_riding_offer
adventure	fish_rod_rental	diving	fishing	overnight_stay_possibility
arrangement	horse_renting_agency	fishing	harbour_round_trip	program
business_event	motor_bicycle_renting_agency	fisticuffs	heath_tour	purge_day
cultural_event	surfboard_rental	fitness_training	hike	recreational_offer
culture	sailing_boat_rental	golf	horse_tour	regimen_offer
dance	yacht_rental	handball	island_round_trip	relax_weekend
educational_event	booking	hang_gliding	nature_experience	romantic_day
excursion	buy	hiking	round_trip	spare_time_possibility
holiday	camping	hunting	several_days_trip	sport_offer
offer	communication	ice_skating	sightseeing	testing_week
qualitative_time_concept	cultural_activity	mini_golf	sightseeing_flight	theater_weekend
sports_event	drive	parachuting	tour	tour_offer
action_affecting_an_object	eating	riding	walk	water_sports_offerings
arrival	informing	sailing	advent	weekend_special
device_state_change	recreate	skiing	adventure_holiday	wellness_offer
human_activity	relaxing	snowboarding	camping	advent
overnight_stay	shopping	soccer	christmas	adventure_holiday
produce	sleep	squash	club_holiday	camping
ski_run	spare_time	swimming	creativity_holiday	christmas
awaking_service	sport	table_tennis	cruise	club_holiday
baby_sitter_service	traveling	tennis	culture_tourism	creativity_holiday
breakfast_service	traveling_by_air	traveling_by_air	easter	cruise
care	visiting	volleyball	end_of_year	culture_tourism
cleaning_service	watching_tv	water_gymnastics	family_holiday	easter
cosmetic_care	working	water_hiking	graduation_travel	end_of_year
cosmetic_therapy	appetizer	water_sport	healthiness_holiday	family_holiday
dog_service	banquet	act	healthiness_tourism	graduation_travel
exchanging_money	brunch	contract	journey	healthiness_holiday
instruction	buffet	holiday_arrangement	regimen	healthiness_tourism
massage	business_dinner	invoice	shopping_tourism	journey
renting_agency	digestive	registration	short_holiday	regimen
reproducing_service	grilling	reservation	sightseeing_tour	shopping_tourism
secretary_service	light_diet	standardization	sports_holiday	short_holiday
shoeblack_service	lunch_packet	theater_arrangement	whit_sun	sightseeing_tour
shuttle_service	menu	weekend_arrangement	city_trip	sports_holiday
aroma_bath	organic_food	business_dinner	deluxe_journey	whit_sun
colour_light_therapy	picnic	conference	educational_holiday	city_trip
facial_therapy	principal_meal	congress	educational_journey	deluxe_journey
foot_care	snack	seminar	event_trip	educational_holiday
hair_cut	vegetarian_food	symposium	pilgrimage	educational_journey
hand_care	dinner	workshop	relaxing_holiday	event_trip
heat	farewell_dinner	ball	short_trip	pilgrimage
make_up	gala_dinner	ballet	sport_trip	relaxing_holiday
salt_bath	breakfast_buffet	concert	trip_for_singles	short_trip
type_advice	dinner_buffet	dancing	weekend_trip	sport_trip
vanishing_cream_pack	dinner	exhibition_opening	boat_round_trip	trip_for_singles
cleaning	gala_menu	festival	formula_one_tour	weekend_trip
face_mask	breakfast	gala	acquaintance_week	boat_round_trip
face_massage	dinner	matinee	bargain	formula_one_tour
peeling	dinner_buffet	musical	day_time	easter_holiday
skin_diagnosis	lunch	opera	holiday	summer_holidays
tightening_therapy	fast_food	pageant	holiday_time	whit_holidays
hand_peeling	vesper	performance	season	winter_holiday
manicure	autumn	puppet_theatre	week	ball_game
nail_design	early_season	talk	weekend	chess_tournament
eyebrow_correction	main_season	theatre	afternoon	ice_hockey
eyelashes_correction	off_season	base_ball	midday	race
permanent_make_up	shoulder_season	billiard_ball	midnight	basket_ball_game
permanent_eyebrow_make_up	spring	brenn_ball	morning	golf_tournament
permanent_lid_make_up	summer	football	night	handball_game
permanent_lip_make_up	summer_season	putting_the_shot_ball	beach_view	soccer_game
dog_care	winter	skiball	panorama	material_thing
dog_doctor	art	table_tennis_ball	panorama_view	partially_material_thing
dog_hair_cutter	music	tennis_ball	sea_view	sight
dog_psychologist	night_life	volleyball	side	sport_equipment

Figure B.2: Sibling groups from GSO2 - part 1 of 2

east_side	backwater	living_creature	catering_company	badminton_court
forest_side	bay	organization	cultural_institution	basketball_ground
lake_side	cape	plant	recreational_institution	beach_volleyball_ground
north_side	coast	animal	service	bow_shooting_range
road_lane	dune	person	beer_garden	bowling_alley
sea_side	embankment	bird	cafe	climbing_wall
south_side	hill	fish	disco	cross_country_ski_run
west_side	island	mammal	experience_gastronomy	diving_station
air	landscape_protection_area	crane	night_cafe	fitness_room
sand	moor	pigeon	wine_tavern	fitness_studio
snow	mountain	stork	animation	football_pitch
water	peninsula	saltwater_fish	art_gallery	gliding_field
fresh_water	river	sweet_water_fish	bullfight	golf_course
saltwater	salt_backwater	cat	cabaret	gym
area	sand	chicken	cinema	horse_riding_school
frontier	sea	dog	concert_house	horse_riding_yard
traffic_route	shore	dolphin	exhibition	ice_hall
city	spit_of_land	donkey	festival_house	mini_golf_area
city_centre	valley	elephant	gallery	shooting_range
continent	beach	giraffe	guided_tour	ski_lift
country	canyon	horse	jazz_club	ski_run
district	desert	monkey	library	skittle_alley
floor	lake	mouse	museum	sports_facilities
forest	basilica	ox	music_house	sports_hall
harbour_area	castle	pork	open_air_theater	squash_court
inner_city	castle_complex	rat	opera	surf_school
market_place	cathedral	actor	theater_house	tennis_court
mountain	church	adult	city_guided_tour	trimmdichpfad
nature	city_wall	agent	museum_guided_tour	volleyball_ground
nature_reserve	craft_work	baby	ball_room	walking_trail
old_town	excursion_goal	boy	beer_garden	water_sports_institution
park	fortress	brother	billiard_room	coastal_resort
pedestrian_area	hunting_castle	business_people	casino	indoor_swimming_pool
place	monastery	child	chimney_room	moor_bath
promenade	muenster	driver	circus	open_air_bath
recreation_area_close_to_a_town	museum	employee	disco	therm
region	old_town	female	garden	coastal_resort
rural_district	oratory	girl	grill_place	indoor_swimming_pool
sea_territory	ruin	gourmet	kursaal	moor_bath
state	stone_grave	grandchild	park	open_air_bath
town	vista_point	grandparents	playground	therm
town_centre	vista_tower	guest	pub	accommodation
village	work_of_art	holiday_maker	regimen_organization	agency
suburb	temple	male	sports_institution	airport
congress_city	ball	musician	swimming_pool	bank
cure_city	bat	organizer	wellness_institution	catering
hanseatic_city	billiard_equipment	pensioner	zoo	exchange_office
harbour_city	fishing_equipment	provider	hotel_garden	hairdresser
metropolis	kin	self_employed_person	winter_garden	harbour
town	racket	sibling	beauty_farm	kindergarden
village	skate	sportsman	beauty_temple	shop
attic	ski	teenager	cosmetician	tourist_information
ground_floor	sledge	tour_guide	fango	train_station
level	snowboard	tourist	fango_application	travel_organizer
upper_floor	surfboard	working_person	fango_therapy	apartment
east_shore	table_tennis_table	aunt	fitness_room	camp
north_shore	base_ball	daughter	hair_dresser	camping_ground
south_shore	billiard_ball	grandma	health_club	club
steep_bank	brenn_ball	mother	keep_fit_course	farm
stone_shore	football	airpark_guest	moor_therapy	guest_house
west_shore	putting_the_shot_ball	conference_guest	sauna	holiday_home
holiday_place	skiball	permanent_guest	solarium	holiday_village
recreation_location	table_tennis_ball	weekend_guest	steam_bath	hotel
sanatorium	tennis_ball	father	sun_studio	motel
forest_border	volleyball	grandfather	swimming_pool	pension
outskirts	billiard_ball	son	thermal_bath	registration
town_border	billiard_queue	uncle	turkish_bath	sanatorium
free_way	billiard_table	cycler	visagist	seminar_house
street	table_tennis_racket	hiker	whirl_pool	youth_hostel
avenue	tennis_racket	professional_sportsman	middle_class_hotel	kiosk
car_race	downhill_ski	water	top_hotel	mall
dog_race	mono_ski	fresh_water	city_harbour	sport_shop
horse_race	water_ski	saltwater	yacht_port	

Figure B.3: Sibling groups from GSO2 - part 2 of 2

Bibliography

[A. Ballester, 2002] A. Ballester, A. Martt'in-Municio, F. P. J. P.-Z. R. R.-U. F. S.-L. (2002). Combining statistics on n-grams for automatic term recognition. In *Proceedings of the 3th International Conference on Language Resources and Evaluation, LREC 2002*.

[Abiteboul, 1997] Abiteboul, S. (1997). Querying semi-structured data. In *Proceedings of 6th International Conference Database Theory, ICDT 1997*. Springer-Verlag.

[Agirre et al., 2000] Agirre, E., Ansa, O., Hovy, E. H., and Martínez, D. (2000). Enriching very large ontologies using the www. In Staab, S., Maedche, A., Nedellec, C., and PeterWiemer-Hastings, editors, *Proceedings of ECAI Workshop on Ontology Learning*, volume 31 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[Agrawal et al., 1993] Agrawal, R., Imielinski, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 207–216.

[Alani et al., 2003] Alani, H., Kim, S., Millard, D. E., Weal, M. J., Lewis, P. H., Hall, W., and Shadbolt, N. (2003). Automatic extraction of knowledge from web documents. In *Workshop on Human Language Technology for the Semantic Web and Web Services, 2 nd Int. Semantic Web Conference*.

[Alfonseca and Manandhar, 2002] Alfonseca, E. and Manandhar, S. (2002). Improving an ontology refinement method with hyponymy patterns. In *Language Resources and Evaluation, Proceedings of the 3th International Conference on Language Resources and Evaluation, LREC 2002*.

[Alvarez et al., 2008] Alvarez, M., Pan, A., Raposo, J., Bellas, F., and Cacheda, F. (2008). Extracting lists of data records from semi-structured web pages. *Data & Knowledge Engineering (DKE)*, 64(2):491–509.

[Anderman and Rogers, 1998] Anderman, G. M. and Rogers, M. (1998). *Words Words Words*. Multilingual Matters.

[Ansgar Bernardi, 2008] Ansgar Bernardi, Stefan Decker, L. v. E. G. G.-T. G. S. H. M. J.-C. M. K. M. G. R. M. S. L. S. (2008). The social semantic desktop: A new paradigm towards deploying the semantic web on the desktop. In Cardoso, J.

and Lytras, M. D., editors, *Semantic Web Engineering in the Knowledge Society*, chapter XII, pages 290–312. IGI Global, Hershey, PA, USA.

[Aston and Burnard, 1998] Aston, G. and Burnard, L. (1998). *The BNC Handbook*. Edinburgh University Press, Edinburgh.

[Aussenac-Gilles and Jacques, 2006] Aussenac-Gilles, N. and Jacques, M.-P. (2006). Designing and evaluating patterns for ontology enrichment from texts. In Staab, S. and Svatek, V., editors, *Proceedings of International Conference on Knowledge Engineering and Knowledge Management, EKAW 2006*, volume 4248 of *Lecture Notes in Artificial Intelligence*, pages 158–165. Springer.

[Bagni et al., 2007] Bagni, D., Cappella, M., Pazienza, M. T., Pennacchiotti, M., and Stellato, A. (2007). Harvesting relational and structured knowledge for ontology building in the wpro architecture. In *Proceedings of the 10th Congress of the Italian Association for Artificial Intelligence, AI*IA 2007*, pages 157–169, Berlin, Heidelberg. Springer-Verlag.

[Baneyx et al., 2005] Baneyx, A., Charlet, J., and Jaulent, M.-C. (2005). Building medical ontologies based on terminology extraction from texts: Methodological propositions. In Miksch, S., Hunter, J., and Keravnou, E. T., editors, *Proceedings of 10th Conference on Artificial Intelligence in Medicine,AIME 2005*, volume 3581 of *Lecture Notes in Computer Science*, pages 231–235. Springer.

[Baroni and Bisi, 2004] Baroni, M. and Bisi, S. (2004). Using cooccurrence statistics and the web to discover synonyms in technical language. In *Proceedings of the 4th International Conference on Language Resources and Evaluation, LREC 2004*, pages 1725–1728.

[Beil et al., 2002] Beil, F., Ester, M., and Xu, X. (2002). Frequent term-based text clustering. In *Eighth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD 2002*, pages 436–442, New York, NY, USA. ACM.

[Bennacer and Karoui, 2005] Bennacer, N. and Karoui, L. (2005). A framework for retrieving conceptual knowledge from web pages. In Bouquet, P. and Tummarello, G., editors, *Proceedingsof the 2nd Italian Semantic Web Workshop, SWAP 2005 - Semantic Web Applications and Perspectives*, volume 166 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[Berners-Lee, 1998] Berners-Lee, T. (1998). Semantic web road map. http://www.w3.org/DesignIssues/Semantic.html.

[Berners-Lee, 2007] Berners-Lee, T. (2007). Giant global graph. http://dig.csail.mit.edu/breadcrumbs/node/215.

[Berners-Lee et al., 1992] Berners-Lee, T., Cailliau, R., Groff, J.-F., and Pollermann, B. (1992). World-Wide Web: The information universe. *Electronic Networking: Research, Applications and Policy*, 1(2):74–82.

[Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, 284(5):34–43.

[Biemann, 2005] Biemann, C. (2005). Ontology learning from text: A survey of methods. *LDV Forum*, 20(2):75–93.

[Biemann et al., 2004a] Biemann, C., Bordag, S., and Quasthoff, U. (2004a). Automatic acquisition of paradigmatic relations using iterated co-occurrences. In *Proceedings of the 4th International Conference on Language Resources and Evaluation, LREC 2004*, Lisboa, Portugal.

[Biemann et al., 2004b] Biemann, C., Bordag, S., and Quasthoff, U. (2004b). Lernen paradigmatischer relationen auf iterierten kollokationen. *LDV Forum*, 19(1/2):103–111.

[Biemann, 2003] Biemann, C.; Bordag, S. H. G. Q. U. (2003). Local ontology engineering - automatic generation of ontological categories from text. AIS Workshop Ontology Learning, Ulm, Dezember 2003.

[Bodenreider et al., 2002] Bodenreider, O., Rindflesch, T. C., and Burgun, A. (2002). Unsupervised, corpus-based method for extending a biomedical terminology. In *Proceedings of the ACL-02 workshop on Natural language processing in the biomedical domain*, pages 53–60, Morristown, NJ, USA. Association for Computational Linguistics.

[Bogdan and Sacaleanu, 2002] Bogdan, P. B. and Sacaleanu, B. (2002). Extending synsets with medical terms. In *Proceedings of the First International Conference on Global WordNet, Mysore, India*, pages 21–25.

[Bos et al., 2007] Bos, B., Çelik, T., Hickson, I., and Lie, H. W. (2007). Cascading style sheets level 2 revision 1 (css 2.1) specification. W3c candidate recommendation, W3C.

[Bourigault and Jacquemin, 1999] Bourigault, D. and Jacquemin, C. (1999). Term extraction + term clustering: An integrated platform for computer-aided terminology. In *Proceedings of Conference of the European Chapter of the Association for Computational Linguistics, EACL 99*, pages 15–22.

[Bray et al., 1998] Bray, T., Paoli, J., and Sperberg-McQueen, C. M. (1998). Extensible Markup Language (XML) 1.0, W3C recommendation 10 february 1998. http://www.w3.org/TR/1998/REC-xml-19980210.

[Brewster et al., 2003] Brewster, C., Ciravegna, F., and Wilks, Y. (2003). Background and foreground knowledge in dynamic ontology construction. In *Proceedings of the SIGIR Semantic Web Workshop*.

[Brin and Page, 1998] Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. In *Proceedings of WWW*, pages 107–117.

[Brunzel, 2007] Brunzel, M. (2007). Learning of semantic sibling group hierarchies - k-means vs. bi-secting-k-means. In Song, I. Y., Eder, J., and Nguyen, T. M., editors, *Proceedings of 9th International Conference on Data Warehousing and Knowledge Discovery, DaWaK 2007*, volume 4654 of *Lecture Notes in Computer Science*, pages 365–374. Springer.

[Brunzel, 2008] Brunzel, M. (2008). The XTREEM methods for ontology learning from Web documents. In Buitelaar, P. and Cimiano, P., editors, *Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, volume 167 of *Frontiers in Artificial Intelligence and Applications*, pages 3–26. IOS Press, Amsterdam,The Netherlands.

[Brunzel and Spiliopoulou, 2005] Brunzel, M. and Spiliopoulou, M. (2005). Extracting domain specific semantics from the WWW with XTREEEX. Workshop des Arbeitskreises Knowledge Discovery (AKKD), Karlsruhe 2005.

[Brunzel and Spiliopoulou, 2006a] Brunzel, M. and Spiliopoulou, M. (2006a). Discovering multi terms and co-hyponymy from XHTML documents with XTREEM. In Nayak, R. and Zaki, M. J., editors, *Proceedings of First International Workshop on Knowledge Discovery from XML Documents, KDXD 2006*, volume 3915 of *Lecture Notes in Computer Science*, pages 22–32. Springer.

[Brunzel and Spiliopoulou, 2006b] Brunzel, M. and Spiliopoulou, M. (2006b). Discovering semantic sibling associations from Web documents with XTREEM-SP. In Tjoa, A. M. and Trujillo, J., editors, *Proceedings of 8th International Conference, DaWaK 2006*, volume 4081 of *Lecture Notes in Computer Science*, pages 469–480. Springer.

[Brunzel and Spiliopoulou, 2006c] Brunzel, M. and Spiliopoulou, M. (2006c). Discovering semantic sibling groups from Web documents with XTREEM-SG. In Staab, S. and Svátek, V., editors, *Managing Knowledge in a World of Networks, Proceedings of 15th International Conference, EKAW 2006*, volume 4248 of *Lecture Notes in Computer Science*, pages 141–157. Springer.

[Brunzel and Spiliopoulou, 2007a] Brunzel, M. and Spiliopoulou, M. (2007a). Acquiring semantic sibling associations from Web documents. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(4):83–98.

[Brunzel and Spiliopoulou, 2007b] Brunzel, M. and Spiliopoulou, M. (2007b). Domain relevance on term weighting. In Kedad, Z., Lammari, N., Métais,

E., Meziane, F., and Rezgui, Y., editors, *Proceedings of 12th International Conference on Applications of Natural Language to Information Systems, NLDB 2007*, volume 4592.

[Brunzel and Spiliopoulou, 2008] Brunzel, M. and Spiliopoulou, M. (2008). Discovering groups of sibling terms from Web documents with XTREEM-SG. *Journal on Data Semantics*, 5383:126–155.

[Buitelaar and Cimiano, 2007] Buitelaar, P. and Cimiano, P. (2007). Ontologies and lexical semantics in natural language understanding. Course at the ESSLLI Summer School - August 2007, Dublin, Ireland.

[Buitelaar et al., 2009] Buitelaar, P., Cimiano, P., Haase, P., and Sintek, M. (2009). Towards linguistically grounded ontologies. In *Proceedings of the 6th European Semantic Web Conference on The Semantic Web (ESWC 2009)*, pages 111–125, Berlin, Heidelberg. Springer-Verlag.

[Buitelaar et al., 2005] Buitelaar, P., Cimiano, P., and Magnini, B. (2005). *Ontology Learning from Text: Methods, Evaluation and Applications*, volume 123 of *Frontiers in Artificial Intelligence and Applications Series*. IOS Press, Amsterdam.

[Burner, 1997] Burner, M. (1997). Crawling towards eternity: Building an archive of the World Wide Web. *Web Techniques Magazine*, 2(5).

[Buttler, 2004] Buttler, D. (2004). A short survey of document structure similarity algorithms. In Arabnia, H. R., Droegehorn, O., Arabnia, H. R., and Droegehorn, O., editors, *Proceedings of the International Conference on Internet Computing (IC '04)*, pages 3–9. CSREA Press.

[Buyukkokten et al., 2001] Buyukkokten, O., Garcia-Molina, H., and Paepcke, A. (2001). Seeing the whole in parts: text summarization for web browsing on handheld devices. In *Proceedings of the 10th international conference on World Wide Web (WWW 2001)*, pages 652–662, New York, NY, USA. ACM.

[Cafarella and Cutting, 2004] Cafarella, M. and Cutting, D. (2004). Building nutch: Open source search. *Queue*, 2(2):54–61.

[Cafarella et al., 2008] Cafarella, M. J., Halevy, A., Wang, D. Z., Wu, E., and Zhang, Y. (2008). Webtables: exploring the power of tables on the web. In *Proceedings VLDB Endow.*, volume 1, pages 538–549. VLDB Endowment.

[Caraballo, 1999] Caraballo, S. A. (1999). Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics (ACL'99)*, pages 120–126, Morristown, NJ, USA. Association for Computational Linguistics.

[Cattuto et al., 2008] Cattuto, C., Benz, D., Hotho, A., and Stumme, G. (2008). Semantic grounding of tag relatedness in social bookmarking systems. In *Proceedings of the 7th International Conference on The Semantic Web (ISWC 2008)*, pages 615–631, Berlin, Heidelberg. Springer-Verlag.

[Cederberg and Widdows, 2003] Cederberg, S. and Widdows, D. (2003). Using lsa and noun coordination information to improve the precision and recall of automatic hyponymy extraction. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 111–118, Morristown, NJ, USA. Association for Computational Linguistics.

[Chakrabarti, 2002] Chakrabarti, S. (2002). *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann, San Francisco, CA, USA.

[Chakrabarti et al., 1999] Chakrabarti, S., van den Berg, M., and Dom, B. (1999). Focused crawling: a new approach to topic-specific web resource discovery. *Computer Networks*, 31(11-16):1623–1640.

[Chen et al., 2006] Chen, J., Yeh, C.-H., and Chau, R. (2006). Identifying multi-word terms by text-segments. In *Proceedings of the Seventh International Conference on Web-Age Information Management Workshops (WAIMW '06)*, page 19, Washington, DC, USA. IEEE Computer Society.

[Cho et al., 1998] Cho, J., Garcia-Molina, H., and Page, L. (1998). Efficient crawling through URL ordering. In *Proceedings WWW*, pages 161–172.

[Cho, 2005] Cho, S. (2005). Indexing for xml siblings. In Doan, A., Neven, F., McCann, R., and Bex, G. J., editors, *Proceedings of the Eight International Workshop on the Web & Databases, WebDB 2005*, pages 91–96.

[Choi et al., 2007] Choi, I., Moon, B., and Kim, H.-J. (2007). A clustering method based on path similarities of XML data. *Data & Knowledge Engineering (DKE)*, 60(2):361–376.

[Chung et al., 2002] Chung, C. Y., Gertz, M., and Sundaresan, N. (2002). Reverse engineering for web data: From visual to semantic structures. In Ellis, A. and Hagino, T., editors, *Proceedings of the 18th International Conference on Data Engineering, ICDE 2002*, Washington, DC, USA. IEEE Computer Society.

[Chung et al., 2006] Chung, S., Jun, J., and McLeod, D. (2006). A web-based novel term similarity framework for ontology learning. In Meersman, R. and Tari, Z., editors, *On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE, OTM Confederated International Conferences, CoopIS, DOA, GADA, and ODBASE 2006*, volume 4275 of *Lecture Notes in Computer Science*, pages 1092–1109. Springer.

[Chung, 2003] Chung, T. M. (2003). A corpus comparison approach for terminology extraction. Terminology 9:2 (2003), 221-246. John Benjamins Publishing Company.

[Church and Hanks, 1989] Church, K. W. and Hanks, P. (1989). Word association norms, mutual information, and lexicography. In *Proceedings of the 27th. Annual Meeting of the Association for Computational Linguistics*, pages 76–83, Vancouver, B.C. Association for Computational Linguistics.

[Cicurel et al., 2006] Cicurel, L., Bloehdorn, S., and Cimiano, P. (2006). Clustering of polysemic words. In *Advances in Data Analysis: Proceedings of the 30th Annual Conferenceof the German Classification Society (GfKl), Berlin, March 8-10, 2006*, Studies in Classification, Data Analysis, and Knowledge Organization. Springer.

[Cimiano, 2006] Cimiano, P. (2006). *Ontology Learning and Population from Text*. PhD thesis, Universität Karlsruhe, Institut für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB).

[Cimiano et al., 2004a] Cimiano, P., Handschuh, S., and Staab, S. (2004a). Towards the self-annotating web. In *Proceedings of the 13th international conference on World Wide Web (WWW 2004)*. ACM Press.

[Cimiano et al., 2004b] Cimiano, P., Hotho, A., and Staab, S. (2004b). Comparing conceptual, divise and agglomerative clustering for learningtaxonomies from text. In de Mántaras, R. L. and Saitta, L., editors, *Proceedings of the 16th Eureopean Conference on Artificial Intelligence,ECAI'2004, including Prestigious Applicants of Intelligent Systems,PAIS 2004*, pages 435–439. IOS Press.

[Cimiano et al., 2004c] Cimiano, P., Schmidt-Thieme, L., Pivk, A., and Staab, S. (2004c). Learning taxonomic relations from heterogeneous evidence. In *ECAI 2004 Ontology Learning and Population Workshop*.

[Cimiano and Staab, 2004] Cimiano, P. and Staab, S. (2004). Learning by googling. *SIGKDD Explorations*, 6(2):24–33.

[Cimiano and Staab, 2005] Cimiano, P. and Staab, S. (2005). Learning concept hierarchies from text with a guided agglomerative clustering algorithm. In Biemann, C. and Paas, G., editors, *Proceedings of the ICML 2005 Workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods*, Bonn, Germany.

[Clark and DeRose, 1999] Clark, J. and DeRose, S. (1999). XML Path Language (XPath), Version 1.0, W3C Recommendation 16 november 1999. http://www.w3.org/TR/1999/REC-xpath-19991116.

[Cleuziou, 2007] Cleuziou, G. (2007). A generalization of k-means for overlapping clustering. Université dŠOrléans, LIFORapport No RR-2007-15.

[Cleuziou et al., 2004] Cleuziou, G., Martin, L., and Vrain, C. (2004). Poboc: An overlapping clustering algorithm, application to rule-based classification and textual data. In *Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI 2004*, pages 440–444.

[Cochran, 1977] Cochran, W. G. (1977). *Sampling Techniques (Third ed.)*. Wiley, New York, USA.

[Cohen and Fan, 2000] Cohen, W. W. and Fan, W. (2000). Web-collaborative filtering: recommending music by crawling the web. In *Proceedings of the 9th international World Wide Web conference on Computer networks : the international journal of computer and telecommunications netowrking*, pages 685–698, Amsterdam, The Netherlands, The Netherlands. North-Holland Publishing Co.

[Cohen et al., 2002] Cohen, W. W., Hurst, M., and Jensen, L. S. (2002). A flexible learning system for wrapping tables and lists in html documents. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, pages 232–241, New York, NY, USA. ACM.

[Cooper et al., 2001] Cooper, B., Sample, N., Franklin, M. J., Hjaltason, G. R., and Shadmon, M. (2001). A fast index for semistructured data. In *Proceedings of the 27th International Conference on Very Large Data Bases (VLDB '01)*, pages 341–350, San Francisco, CA, USA. Morgan Kaufmann.

[Costa et al., 2004] Costa, G., Manco, G., Ortale, R., and Tagarelli, A. (2004). A tree-based approach to clustering xml documents by structure. In *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2004)*, pages 137–148, New York, NY, USA. Springer.

[Cruse, 2004] Cruse, A. (2004). *Meaning in language: An introduction to semantics and pragmatics (2nd ed.)*. Oxford University Press, New York, USA.

[Cruz and Nicolle, 2008] Cruz, C. and Nicolle, C. (2008). Ontology enrichment and automatic population from xml data. In *Proceedings of the 4th International VLDB Workshop on Ontology-based Techniques for DataBases in Information Systems and Knowledge Systems, ODBIS 2008, Co-located with the 34th International Conference on Very Large Data Bases*, pages 17–20.

[Cutting, 2005] Cutting, D. (2005). Nutch: an open-source platform for Web search. Workshop on Open Source Web Information Retrieval, OSWIR 2005, in association with the 2005 IEEE/WIC/ACM International Conferences on Web Intelligence & Intelligent Agent Technology.

[Dalamagas et al., 2006] Dalamagas, T., Cheng, T., Winkel, K.-J., and Sellis, T. (2006). A methodology for clustering XML documents by structure. *Information Systems*, 31(3):187–228.

[Dalamagas et al., 2004] Dalamagas, T., Cheng, T., Winkel, K.-J., and Sellis, T. K. (2004). Clustering XML documents using structural summaries. In Lindner, W., Mesiti, M., Türker, C., Tzitzikas, Y., and Vakali, A., editors, *Current Trends in Database Technology - EDBT 2004 Workshops*, volume 3268 of *Lecture Notes in Computer Science*, pages 547–556. Springer.

[Damerau, 1993] Damerau, F. J. (1993). Generating and evaluating domain-oriented multi-word terms from texts. *Information Processing and Management*, 29(4):433–447.

[David Manzano-Macho and Borrajo, 2008] David Manzano-Macho, A. G.-P. and Borrajo, D. (2008). Unsupervised and domain independent ontology learning: Combining heterogeneous sources of evidence. In *Proceedings of the 6th International Language Resources and Evaluation, LREC 2008*, Marrakech, Morocco.

[Davidsson, 1997] Davidsson, P. (1997). Integrating models of discrimination and characterization for learning fromexamples in open domains. In *Proceedings International Joint Conferences on Artificial Intelligence, IJCAI 1997*, pages 840–845. Morgan Kaufmann Publishers Inc.

[Deane, 2005] Deane, P. (2005). A nonparametric method for extraction of candidate phrasal terms. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 605–613, Ann Arbor, Michigan. Association for Computational Linguistics.

[Debole and Sebastiani, 2003] Debole, F. and Sebastiani, F. (2003). Supervised term weighting for automated text categorization. In *SAC '03: Proceedings of the 2003 ACM symposium on Applied computing*, pages 784–788, New York, NY, USA. ACM.

[Deerwester et al., 1990] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407.

[Defays, 1977] Defays, D. (1977). An efficient algorithm for a complete link method. *The Computer Journal*, 20:364–366.

[Dias et al., 2000] Dias, G., Guilloré, S., Bassano, J.-C., and Lopes, J. P. (2000). Combining linguistics with statistics for multiword term extraction: A fruitful association? In *Proceedings of Recherche d'Informations Assistee par Ordinateur 2000, RIAO 2000*.

[Doorenbos et al., 1997] Doorenbos, R. B., Etzioni, O., and Weld, D. S. (1997). A scalable comparison-shopping agent for the world-wide web. In *Proceedings of the First International Conference on Autonomous Agents*, pages 39–48. ACM Press.

[Dorow, 2006] Dorow, B. (2006). *A graph model for words and their meanings*. PhD thesis, University of Stuttgart.

[Dowdall et al., 2004] Dowdall, J., Elleman, J., Hess, M., Lowe, W., and Rinaldi, F. (2004). The role of MultiWord Terminology in Knowledge Management. In *Proceedings of 4th International Conference on Language Resources and Evaluation, LREC 2004*.

[Drouin, 2004] Drouin, P. (2004). Detection of domain specific terminology using corpora comparison. In *Proceedings of the 4th international Conference on Language Resources and Evaluation, LREC 2004*.

[Duda et al., 2000] Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Classification*. Wiley-Interscience, 2nd edition.

[Ehrig, 2006] Ehrig, M. (2006). *Ontology Alignment - Bridging the Semantic Gap*. PhD thesis, Universität Karlsruhe (TH).

[Ehrig and Maedche, 2003] Ehrig, M. and Maedche, A. (2003). Ontology-focused crawling of Web documents. In *Proceedings of ACM symposium on Applied computing*, pages 1174–1178, New York, NY, USA. ACM Press.

[English and Nirenburg, 2007] English, J. and Nirenburg, S. (2007). Ontology learning from text using automatic ontological - semantic text annotation and the web as the corpus. Technical report.

[Etzioni et al., 2004] Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A. (2004). Webscale information extraction in knowitall: (preliminary results). In *Proceedings of the 13th international conference on World Wide Web (WWW 2004)*, pages 100–110, New York, NY, USA. ACM Press.

[Etzioni et al., 2005] Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A. (2005). Unsupervised named-entity extraction from the web: an experimental study. *Artificial Intelligence*, 165(1):91–134.

[Evert, 2005] Evert, S. (2005). *The statistics of word cooccurrences - word pairs and collocations*. PhD thesis, Uni Stuttgart, Philosophisch-historische Fakultät, Institut für Maschinelle Sprachverarbeitung, Fachrichtung Computerlingustik.

[Faatz and Steinmetz, 2002] Faatz, A. and Steinmetz, R. (2002). Ontology enrichment with texts from the WWW. In *Proceedings of the First International Workshop on Semantic Web Mining at the ECML 2002*, Helsinki, Finland.

[Faure and N'edellec, 1998] Faure, D. and N'edellec, C. (1998). A corpus-based conceptual clustering method for verb frames and ontology acquisition. LREC workshop on Adapting lexical and corpus resources to sublanguages and applications.

[Fellbaum, 1998] Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, USA.

[Frantzi et al., 2000] Frantzi, K. T., Ananiadou, S., and Mima, H. (2000). Automatic recognition of multi-word terms: the c-value/nc-value method. *International Jounral on Digital Libraries*, 3(2):115–130.

[Freitag et al., 2005] Freitag, D., Blume, M., Byrnes, J., Chow, E., Kapadia, S., Rohwer, R., and Wang, Z. (2005). New experiments in distributional representations of synonymy. In *Proceedings of the Ninth Conference on Computational Natural Language Learning, CoNLL-2005*, pages 25–32, Ann Arbor, Michigan. Association for Computational Linguistics.

[Fung et al., 2003] Fung, B. C. M., Wang, K., and Ester, M. (2003). Hierarchical document clustering using frequent itemsets. In Barbará, D. and Kamath, C., editors, *International Conference on Data Mining 2003, SDM 2003*. SIAM.

[Gamallo et al., 2005] Gamallo, P., Agustini, A., and Lopes, G. P. (2005). Clustering syntactic positions with similar semantic requirements. *Computational Linguistics*, 31(1):107–146.

[Ghahramani and Heller, 2005] Ghahramani, Z. and Heller, K. (2005). Bayesian sets. In *Advances in Neural Information Processing Systems 18, NIPS 2005*.

[Gillam and Tariq, 2004] Gillam, L. and Tariq, M. (2004). Ontology via terminology? In *Proceedings of Workshop on Terminology, Ontology and Knowledge Representation, Termino 2004, Lyon, France*.

[Gómez-Pérez and Manzano-Macho, 2003] Gómez-Pérez, A. and Manzano-Macho, D. (2003). A survey of ontology learning methods and techniques. deliverable 1.5, OntoWeb project, universidad polytecnica de madrid. Technical Report OntoWeb Deliverable D1.5, Universidad Polytecnica de Madrid.

[Gottgtroy et al., 2003] Gottgtroy, P., Kasabov, N., and MacDonell, S. (2003). An ontology engineering approach for knowledge discovery from data in evolving domains. In *Proceedings of the 4th International Conference on Data Mining, Data Mining IV*, pages 43–52. WIT Press, Federal University of Rio de Janeiro.

[Gou and Chirkova, 2007] Gou, G. and Chirkova, R. (2007). Efficiently querying large xml data repositories: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 19(10):1381–1403.

[Grefenstette, 1994] Grefenstette, G. (1994). Explorations in automatic thesaurus construction. Kluwer.

[Groh and Toni, 2005] Groh, G. and Toni, K. (2005). Ontologyengineering part 4 - general guidelines withprotégé. Reviewed 4 hour tutorial at the IADIS International Conference WWW/Internet 2005, Lisbon, Portugal, 19-22 October 2005.

[Group, 2008] Group, W. W. (2008). Rdfa primer: Bridging the human and data webs. http://www.w3.org/TR/xhtml-rdfa-primer/.

[Gruber, 1993] Gruber, T. R. (1993). Towards principles for the design of ontologies used for knowledge sharing. In Guarino, N. and Poli, R., editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Deventer, The Netherlands. Kluwer Academic Publishers.

[Halevy et al., 2009] Halevy, A., Norvig, P., and Pereira, F. (2009). The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12.

[Harris, 1968] Harris (1968). *Mathematical Language*. Wiley, New York, USA.

[Harris, 1954] Harris, Z. S. (1954). Distributional structure. *Word*, 10(23):146–162.

[Hartigan and Wong, 1979] Hartigan, J. A. and Wong, M. A. (1979). A K-means clustering algorithm. *Applied Statistics*, 28:100–108.

[Hazman et al.,] Hazman, M., Reclamation, L., Giza, E., El-Beltagy, S. R., Rafea, A., and Cairo, E. Ontology learning from textual web documents. In *Proceedings of the 6th International Conference on Informatics and Systems (INFOS'2008).*

[Hearst, 1992] Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics*, pages 539–545, Morristown, NJ, USA. Association for Computational Linguistics.

[Henze, 2004] Henze, N. (2004). Semantic web lecture, short introduction to ontology engineering. http://www.kbs.uni-hannover.de/ henze/semweb04/skript/slides/14_06_2004.pdf.

[Herbelot and Copestake, 2006] Herbelot, A. and Copestake, A. (2006). Acquiring ontological relationships from wikipedia using rmrs. In *Proceedings of the ISWC 2006 Workshop on Web Content Mining with Human Language Technologies.*

[Heyer et al., 2001] Heyer, G., Läuter, M., Quasthoff, U., Wittig, T., and Wolff, C. (2001). Learning relations using collocations. In *Proceedings of IJCAI'2001 Workshop on Ontology Learning (OL'2001)*, volume 38 of *CEUR Workshop Proceedings*, Seattle, USA. CEUR-WS.org.

[Hirst, 1995] Hirst, G. (1995). Near-synonymy and the structure of lexical knowledge. In *In AAAI Symposium on Representation and Acquisition of Lexical Knowledge: Polysemy, Ambiguity, and Generativity*, pages 51–56.

[Inkpen, 2007a] Inkpen, D. (2007a). Near-synonym choice in an intelligent thesaurus. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 356–363, Rochester, New York. Association for Computational Linguistics.

[Inkpen, 2007b] Inkpen, D. (2007b). A statistical model for near-synonym choice. *ACM Transactions on Speech and Language Processing*, 4(1):2.

[Inkpen and Hirst, 2006] Inkpen, D. and Hirst, G. (2006). Building and using a lexical knowledge base of near-synonym differences. *Computational Linguistics*, 32(2):223–262.

[Inkpen and Hirst, 2002] Inkpen, D. Z. and Hirst, G. (2002). Acquiring collocations for lexical choice between near-synonyms. In *Proceedings of the ACL-02 workshop on Unsupervised lexical acquisition*, pages 67–76, Morristown, NJ, USA. Association for Computational Linguistics.

[Inkpen and Hirst, 2003] Inkpen, D. Z. and Hirst, G. (2003). Near-synonym choice in natural language generation. In *Proceedings of the International Conference RANLP-2003 (Recent Advances in Natural Language Processing*, pages 4–3. John Benjamins Publishing Company.

[Iyer and Simovici, 2007] Iyer, S. and Simovici, D. A. (2007). Multisets and clustering xml documents. In *Proceedings of 19th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2007*, pages 267–274. IEEE Computer Society.

[Jackendoff, 1997] Jackendoff, R. (1997). *The Architecture of the Language Faculty*. MIT Press, Cambridge, MA, USA.

[Jacquemin and Bourigault, 2003] Jacquemin, C. and Bourigault, D. (2003). Term extraction and automatic indexing. In *The Oxford Handbook of Computational Linguistics*, chapter 33.

[Jain and Dubes, 1988] Jain, A. K. and Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, NJ.

[Jannink and Wiederhold, 1999] Jannink, J. and Wiederhold, G. (1999). Thesaurus entry extraction from an on-line dictionary. In *Proceedings Second International Conference Information Fusion*.

[Joachims, 1997] Joachims, T. (1997). A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML 97)*, pages 143–151, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Joliffe, 1986] Joliffe, I. T. (1986). *Principal Component Analysis*. Springer-Verlag, Berlin Heidelberg.

[Jones, 1973] Jones, K. S. (1973). Index term weighting. *Information Storage and Retrieval*, 9(11):619–633.

[Jung and Kwon, 2006] Jung, S.-W. and Kwon, H.-C. (2006). A scalable hybrid approach for extracting head components from web tables. *IEEE Transactions on Knowledge and Data Engineering*, 18(2):174–187.

[Junichiro et al., 2004] Junichiro, M., Yutaka, M., Mitsuru, I., and Boi, F. (2004). Keyword extraction from the web for personal metadata annotation. In *Proceedings of the 4th International Workshop on Knowledge Markup and Semantic Annotation,(ISWC2004)*, pages 51–60, Hiroshima, Japan,.

[Karoui et al., 2004] Karoui, L., Aufaure, M.-A., and Bennacer, N. (2004). Ontology discovery from web pages: Application to tourism. In *KDO*.

[Karoui et al., 2007] Karoui, L., Aufaure, M.-A., and Bennacer, N. (2007). Contextual concept discovery algorithm. In Wilson, D. and Sutcliffe, G., editors, *Proceedings of the Twentieth International Florida Artificial Intelligence Research Society Conference*, pages 460–465. AAAI Press.

[Kathrin Eichler and Neumann, 2008] Kathrin Eichler, H. H. and Neumann, G. (2008). Unsupervised relation extraction from web documents. In *Proceedings of the 6th International Language Resources and Evaluation, LREC 2008*, Marrakech, Morocco.

[Kayed and Shaalan, 2006] Kayed, M. and Shaalan, K. F. (2006). A survey of web information extraction systems. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1411–1428. Member-Chia-Hui Chang and Member-Moheb Ramzy Girgis.

[Kilgarriff, 2001] Kilgarriff, A. (2001). Comparing corpora. *International Journal of Corpus Linguistics*, 6(1):97–133.

[Kilgarriff and Grefenstette, 2003] Kilgarriff, A. and Grefenstette, G. (2003). Introduction to the special issue on the Web as corpus. *Computer Linguistics*, 29(3):333–347.

[King, 1967] King, B. (1967). Step-wise clustering procedures. *Journal of the American Statistical Association*, 69:86–101.

[Krátký and Baca, 2006] Krátký, M. and Baca, R. (2006). A comparison of element-based and path-based approaches to indexing xml data. In Snásel, V., Richta, K., and Pokorný, J., editors, *Proceedings of the Dateso 2006 Annual International Workshop on DAtabases, TExts, Specifications and Objects*, volume 176 of *CEUR Workshop Proceedings*. CEUR-WS.org.

[Kruschwitz, 2001a] Kruschwitz, U. (2001a). Exploiting structure for intelligent Web search. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences, HICSS '01*, Washington, DC, USA. IEEE Computer Society.

[Kruschwitz, 2001b] Kruschwitz, U. (2001b). A rapidly acquired domain model derived from markup structure. In *Proceedings of the ESSLLI'01 Workshop on Semantic Knowledge Acquisition and Categorization*, Helsinki, Finland.

[Kushmerick et al., 1997] Kushmerick, N., Weld, D. S., and Doorenbos, R. B. (1997). Wrapper induction for information extraction. In *International Joint Conference on Artificial Intelligence, IJCAI 1997*, pages 729–737.

[Labský et al., 2005] Labský, M., Svátek, V., Sváb, O., Praks, P., Krátký, M., and Snásel, V. (2005). Information extraction from HTML product catalogues: From source code and images to rdf. In Skowron, A., Agrawal, R., Luck, M., Yamaguchi, T., Morizet-Mahoudeaux, P., Liu, J., and Zhong, N., editors, *2005 IEEE / WIC / ACM International Conference on Web Intelligence, WI 2005*, pages 401–404. IEEE Computer Society.

[Lan et al., 2005] Lan, M., Tan, C.-L., Low, H.-B., and Sung, S.-Y. (2005). A comprehensive comparative study on term weighting schemes for text categorization with support vector machines. In *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 1032–1033, New York, NY, USA. ACM Press.

[Landauer and Dumais, 1997] Landauer, T. K. and Dumais, S. T. (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition,induction, and representation of knowledge. *Psychological Review*, 104(2):211–240.

[Lee, 2007] Lee, L. (2007). Idf revisited: a simple new derivation within the robertson-spärck jones probabilistic model. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 751–752, New York, NY, USA. ACM.

[Leung et al., 2005] Leung, H.-P., Chung, F.-L., and Chan, S. C.-F. (2005). On the use of hierarchical information in sequential mining-based xml document similarity computation. *Knowledge and Information Systems*, 7(4):476–498.

[Lin et al., 2003] Lin, D., Zhao, S., Qin, L., and Zhou, M. (2003). Identifying synonyms among distributionally similar words. In Gottlob, G. and Walsh, T., editors, *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, IJCAI-03*, pages 1492–1493, San Francisco, CA, USA. Morgan Kaufmann.

[Liu and Yu, 2005] Liu, H. and Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(4):491–502. Senior Member-Liu, Huan and Student Member-Yu, Lei.

[Liu et al., 2004] Liu, Z., Ng, W. K., and Lim, E.-P. (2004). An automated algorithm for extracting website skeleton. In Lee, Y.-J., Li, J., Whang, K.-Y., and Lee, D., editors, *Database Systems for Advances Applications, 9th International Conference, DASFAA 2004*, volume 2973 of *Lecture Notes in Computer Science*, pages 799–811. Springer.

[Lloyd, 1957] Lloyd, S. (1957). Least squares quantization in pcm. Technical report, Bell Laboratories.

[Long et al., 2005] Long, J., Schwartz, D. G., and Stoecklin, S. (2005). An xml distance measure. In Arabnia, H. R. and Scime, A., editors, *Proceedings of The 2005 International Conference on Data Mining, DMIN 2005*, pages 119–125. CSREA Press.

[Lüdeling et al., 2007] Lüdeling, A., Evert, S., and Baroni, M. (2007). Using web data for linguistic purposes. In Hundt, M., Nesselhauf, N., and Biewer, C., editors, *Corpus Linguistics and the Web*, pages 7–24. Rodopi, Amsterdam.

[Lyons, 1977] Lyons, J. (1977). *Semantics*, volume 1. Cambridge University Press, New York, NY, USA.

[MacQueen, 1967] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *5th Berkley Symposium on Mathematics and Probability*, pages 281–297.

[Maedche and Staab, 2000] Maedche, A. and Staab, S. (2000). Discovering conceptual relations from text. In *Proceedings of the 13th European Conference on Artificial Intelligence, ECAI 2000*, pages 321–325, Amsterdam, The Nederlands. IOS Press.

[Maedche and Staab, 2001] Maedche, A. and Staab, S. (2001). Ontology Learning for the Semantic Web. *IEEE Intelligent Systems*, 16(2):72–79.

[Makagonov et al., 2005] Makagonov, P., A, R. F., Sboychakov, K., and Gelbukh, A. (2005). Learning a domain ontology from hierarchically structured texts. In Biemann, C. and Paas, G., editors, *Proceedings of the ICML 2005 Workshop*

on Learning and Extending Lexical Ontologies with Machine Learning Methods, Bonn, Germany.

[Manku et al., 1999] Manku, G. S., Rajagopalan, S., and Lindsay, B. G. (1999). Random sampling techniques for space efficient online computation of order statistics of large datasets. In *SIGMOD '99: Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, pages 251–262, New York, NY, USA. ACM.

[Manning et al., 2008] Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

[Manning and Schütze, 1999] Manning, C. D. and Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT Press, Cambridge, MA, USA.

[Maria Ruiz-Casado and Castells, 2005] Maria Ruiz-Casado, E. A. and Castells, P. (2005). Using context-window overlapping in synonym discovery and ontology extension. In *International Conference on Recent Advances in Natural Language Processing, RANLP 2005*, Sofia, Bulgaria.

[Mariam et al., 2005] Mariam, L. G., Gillam, L., Tariq, M., and Ahmad, K. (2005). Terminology and the construction of ontology. *Terminology*, 11:55–81.

[Markman, 1989] Markman, E. M. (1989). *Categorization and Naming in Children: Problems of Induction*. MIT Press, Cambridge, MA, USA.

[Markov and Larose, 2007] Markov, Z. and Larose, D. T. (2007). *Data Mining the Web: Uncovering Patterns in Web Content, Structure, and Usage*. Wiley-Interscience.

[Mervis and Rosch, 1981] Mervis, C. B. and Rosch, E. (1981). Categorization of natural objects. *Annual Review of Psychology*, 32(1):89–115.

[Metzler, 2008] Metzler, D. (2008). Generalized inverse document frequency. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 399–408, New York, NY, USA. ACM.

[Mitkov, 2003] Mitkov, R. (2003). *The Oxford Handbook of Computational Linguistics (Glossary)*. Oxford Handbooks in Linguistics. Oxford University Press, Oxford.

[Mladenic, 1998] Mladenic, D. (1998). Feature subset selection in text-learning. In *ECML '98: Proceedings of the 10th European Conference on Machine Learning*, pages 95–100, London, UK. Springer-Verlag.

[Moigno et al., 2002] Moigno, S. L., Charlet, J., Bourigault, D., Degoulet, P., and Jaulent, M.-C. (2002). Terminology extraction from text to build an ontology in surgical intensive care. In *Proceedings of the ECAI 2002 workshop on NLP and ML for Ontology Engineering*, Lyon.

[Morin and Jacquemin, 2004] Morin, E. and Jacquemin, C. (2004). Automatic acquisition and expansion of hypernym links. *Computer and the Humanities*, 38(4):343–362.

[Mukherjee et al., 2003] Mukherjee, S., Yang, G., Tan, W., and Ramakrishnan, I. V. (2003). Automatic discovery of semantic structures in html documents. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition, ICDAR 2003*, page 245, Washington, DC, USA. IEEE Computer Society.

[Murray and Reuter, 2005] Murray, G. C. and Reuter, K. (2005). Children's acquisition of categories and the implications for research in the development of classification schemes. Presented at the SIGCR workshop at the 2005 Annual Meeting of the American Society for Information Science and Technology.

[Murtagh, 1983] Murtagh, F. (1983). A survey of recent advances in hierarchical clustering algorithms. *Computer Journal*, 26(4):354–359.

[Muslea et al., 1999] Muslea, I., Minton, S., and Knoblock, C. (1999). A hierarchical approach to wrapper induction. In Etzioni, O., Müller, J. P., and Bradshaw, J. M., editors, *Proceedings of the Third International Conference on Autonomous Agents, Agents 1999*, pages 190–197, Seattle, WA, USA. ACM Press.

[Muslea et al., 2001] Muslea, I., Minton, S., and Knoblock, C. A. (2001). Hierarchical wrapper induction for semistructured information sources. *Autonomous Agents and Multi-Agent Systems*, 4(1-2):93–114.

[Najork and Heydon, 2001] Najork, M. and Heydon, A. (2001). High-performance web crawling. Technical Report 173, Compaq Systems Research Center.

[Najork and Heydon, 2002] Najork, M. and Heydon, A. (2002). High-performance web crawling. In James Abello, P. P. and Resende, M., editors, *Handbook of Massive Data Sets*, chapter 2, pages 25 – 45. Kluwer, Norwell, MA, USA.

[Nakagawa, 2001] Nakagawa, H. (2001). Automatic term recognition based on statistics of compound nouns. *Terminology*, 6:195–210.

[Nakagawa and Mori, 2003] Nakagawa, H. and Mori, T. (2003). Automatic term recognition based on statistics of compound nouns and their components. *Terminology*, 9(2):201–219.

[Navigli, 2005] Navigli, R. (2005). Ontology learning from a domain Web corpus. In Scime, A., editor, *Web Mining: Applications and Techniques*, pages 69–98. Idea Group Publishing, Hershey.

[Nierman and Jagadish, 2002] Nierman, A. and Jagadish, H. V. (2002). Evaluating structural similarity in XML documents. In *Proceedings of the Fifth International Workshop on the Web and Databases, WebDB 2002*, Madison, Wisconsin, USA.

[Ohshima et al., 2006] Ohshima, H., Oyama, S., and Tanaka, K. (2006). Searching coordinate terms with their context from the web. In Aberer, K., Peng, Z., Rundensteiner, E. A., Zhang, Y., and Li, X., editors, *Proceedings of 7th International Conference on Web Information Systems Engineering, WISE 2006*, volume 4255 of *Lecture Notes in Computer Science*, pages 40–47. Springer.

[Pantel and Lin, 2001] Pantel, P. and Lin, D. (2001). A statistical corpus-based term extractor. In *Proceedings of the 14th Biennial Conference of the Canadian Society on Computational Studies of Intelligence, AI 2001*, pages 36–46, London, UK. Springer-Verlag.

[Pantel, 2003] Pantel, P. A. (2003). *Clustering by committee*. PhD thesis, Edmonton, Alta., Canada. Adviser-Dekang Lin.

[Papineni, 2001] Papineni, K. (2001). Why inverse document frequency? In *NAACL '01: Second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies 2001*, pages 1–8, Morristown, NJ, USA. Association for Computational Linguistics.

[Pasca, 2004] Pasca, M. (2004). Acquisition of categorized named entities for web search. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 137–145, New York, NY, USA. ACM.

[Pasca, 2005] Pasca, M. (2005). Finding instance names and alternative glosses on the Web: Wordnet reloaded. In Gelbukh, A. F., editor, *Proceedings of the 6th Computational Linguistics and Intelligent Text Processing International Conference (CICLing 2005)*, volume 3406 of *Lecture Notes in Computer Science*, pages 280–292. Springer.

[Pasca, 2008] Pasca, M. (2008). Low-complexity heuristics for deriving fine-grained classes of named entities from web textual data. In *Proceedings of the Sixth International Language Resources and Evaluation , LREC 2008*, Marrakech, Morocco.

[Pekar and Krkoska, 2003] Pekar, V. and Krkoska, M. (2003). Weighting distributional features for automatic semantic classification of words. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing, RANLP 2003*, pages 369–373.

[Pekar et al., 2004] Pekar, V., Krkoska, M., and Staab, S. (2004). Feature weighting for co-occurrence-based classification of words. In *20th Conference on Computational Linguistics, COLING 2004*.

[Pemberton et al., 2000] Pemberton, S., Altheim, M., Austin, D., Boumphrey, F., Burger, J., Donoho, A. W., Dooley, S., Hofrichter, K., Hoschka, P., Ishikawa, M., ten Kate, W., King, P., Klante, P., Matsui, S., McCarron, S., Navarro, A., Nies, Z., Raggett, D., Schmitz, P., Schnitzenbaumer, S., Stark, P., Wilson, C., Wugofski, T., and Zigmond, D. (2000). XHTML 1.0: The Extensible HyperText Markup Language, a reformulation of HTML 4 in XML 1.0, W3C recommendation 26 january 2000. http://www.w3.org/TR/2000/REC-xhtml1-20000126/.

[Piao et al., 2003] Piao, S. S. L., Rayson, P., Archer, D., Wilson, A., and McEnery, T. (2003). Extracting multiword expressions with a semantic tagger. In *Proceedings of the ACL 2003 workshop on Multiword expressions*, pages 49–56, Morristown, NJ, USA. Association for Computational Linguistics.

[Pierre, 1980] Pierre, L. (1980). Sur la variabilité de la fréquence des formes dans un corpus. M.O.T.S, 1, pp. 127-165.

[Pivk et al., 2005] Pivk, A., Cimiano, P., and Sure, Y. (2005). From tables to frames. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(2-3):132–146.

[Plackett, 1983] Plackett, R. L. (1983). Karl pearson and the chi-squared test. *International Statistical Review / Revue Internationale de Statistique*, 51(1):59–72.

[Ponzetto and Strube, 2007] Ponzetto, S. P. and Strube, M. (2007). Deriving a large scale taxonomy from wikipedia. In *Proceedings of the National Conference on Artificial Intelligence*, volume 22, page 1440. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

[Quasthoff and Wolff, 2002] Quasthoff, U. and Wolff, C. (2002). The poisson collocation measure and its applications. In *Second International Workshop on Computational Approaches to Collocations*.

[Raggett et al., 1997] Raggett, D., Hors, A. L., and Jacobs, I. (1997). HTML 4.0 Specification, W3C recommendation 18 december 1997. http://www.w3.org/TR/REC-html40-971218/.

[Ravichandran et al., 2004] Ravichandran, D., Pantel, P., and Hovy, E. (2004). The terascale challenge. Proceedings of KDD Workshop on Mining for and from the Semantic Web (MSW-04). pp. 1-11. Seattle, WA.

[Rector et al., 2006] Rector, A., Noy, N., Drummond, N., and Musen, M. (2006). Ontology design patterns and problems: Practical ontology engineering using protege-owl. Tutorial of EKAW 2006.

[Reed et al., 2006] Reed, J. W., Jiao, Y., Potok, T. E., Klump, B. A., Elmore, M. T., and Hurson, A. R. (2006). Tf-icf: A new term weighting scheme for clustering dynamic data streams. In *Proceedings of the 5th International Conference on Machine Learning and Applications, ICMLA 2006*, pages 258–263, Washington, DC, USA. IEEE Computer Society.

[Rigau, 1994] Rigau, G. (1994). An experiment on automatic semantic tagging of dictionary senses. In *International Workshop the Future of the Dictionary*.

[Riloff and Shepherd, 1997] Riloff, E. and Shepherd, J. (1997). A corpus-based approach for building semantic lexicons. *The Computing Research Repository - CORR*, cmp-lg/9706013:117–124.

[Rinaldi et al., 2005] Rinaldi, F., Yuste, E., Schneider, G., Hess, M., and Roussel, D. (2005). Exploiting technical terminology for knowledge management. In Buitelaar, P., Cimiano, P., and Magnini, B., editors, *Ontology Learning from Text: Methods, Evaluation and Applications*, pages 140–154, Amsterdam: IOS Press (Frontiers in artificial intelligence and applications, edited by J. Breuker et al., volume 123).

[Roark and Charniak, 1998] Roark, B. and Charniak, E. (1998). Noun-phrase co-occurrence statistics for semiautomatic semantic lexicon construction. In *Proceedings of the 17th international conference on Computational linguistics*, pages 1110–1116, Morristown, NJ, USA. Association for Computational Linguistics.

[Robertson, 2004] Robertson, S. (2004). Understanding inverse document frequency: On theoretical arguments for idf. *Journal of Documentation*, 60:503Ű520.

[Rohit Khare, 2004] Rohit Khare, Doug Cutting, K. S. A. R. (2004). Nutch: A flexible and scalable open-source Web search engine. CommerceNet Labs Technical Report 04-04.

[Ruenes, 2007] Ruenes, D. S. (2007). *Domain Ontology Learning from the Web*. PhD thesis, Universitat Politècnica de Catalunya, Tarragona.

[Ruiz-Casado et al., 2006] Ruiz-Casado, M., Alfonseca, E., and Castells, P. (2006). Automatic extraction of semantic relationships for wordnet by means of pattern learning from wikipedia. In *Proceedings of 11th International Conference on Applications of Natural Language to Information Systems, NLDB 2006*, pages 67–79.

[Sag et al., 2002] Sag, I. A., Baldwin, T., Bond, F., Copestake, A. A., and Flickinger, D. (2002). Multiword expressions: A pain in the neck for nlp. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing, CICLing 2002*, pages 1–15, London, UK. Springer-Verlag.

[Salton and Buckley, 1987] Salton, G. and Buckley, C. (1987). Term weighting approaches in automatic text retrieval. Technical report, Cornell University, Ithaca, NY, USA.

[Sanchez and Moreno, 2006] Sanchez, D. and Moreno, A. (2006). A methodology for knowledge acquisition from the web. *International Journal of Knowledge-Based and Intelligent Engineering Systems*, 10(6):453–475.

[Sanchez and Moreno, 2008a] Sanchez, D. and Moreno, A. (2008a). Learning non-taxonomic relationships from web documents for domain ontology construction. *Data & Knowledge Engineering (DKE)*, 64(3):600–623.

[Sanchez and Moreno, 2008b] Sanchez, D. and Moreno, A. (2008b). Pattern-based automatic taxonomy learning from the web. *AI Communications*, 21(1):27–48.

[Sanderson and Croft, 1999] Sanderson, M. and Croft, B. (1999). Deriving concept hierarchies from text. In *Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval, SIGIR 1999*, pages 206–213, New York, NY, USA. ACM Press.

[Sarawagi, 2002] Sarawagi, S. (2002). Automation in information extraction and integration. Tutorial of The 28th International Conference on Very Large Data Bases (VLDB).

[Sauermann et al., 2007] Sauermann, L., van Elst, L., and Dengel, A. (2007). Pimo - a framework for representing personal information models. In Pellegrini, T. and Schaffert, S., editors, *Proceedings of I-Semantics'07*, pages pp. 270–277. JUCS.

[Savaresi et al., 2000] Savaresi, S. M., Boley, D. L., Bittanti, S., and Gazzaniga, G. (2000). Choosing the cluster to split in bisecting divisive clustering algorithms. Technical report.

[Sazedj and Pinto, 2007] Sazedj, P. and Pinto, H. S. (2007). Mining the web through verbs: A case study. In *Proceedings of the 4th European conference on The Semantic Web, ESWC 2007*, pages 488–502, Berlin, Heidelberg. Springer-Verlag.

[Schaal et al., 2005] Schaal, M., Müller, R. M., Brunzel, M., and Spiliopoulou, M. (2005). Relfin - topic discovery for ontology enhancement and annotation. In Gómez-Pérez, A. and Euzenat, J., editors, *The Semantic Web: Research and Applications, Proceedings of Second European Semantic Web Conference,*

ESWC 2005, volume 3532 of *Lecture Notes in Computer Science*, pages 608–622. Springer.

[Shamsfard and Barforoush, 2003] Shamsfard, M. and Barforoush, A. A. (2003). The state of the art in ontology learning: a framework for comparison. *The Knowledge Engineering Review*, 18(4):293–316.

[Shamsfard and Barforoush, 2004] Shamsfard, M. and Barforoush, A. A. (2004). Learning ontologies from natural language texts. *International Journal of Human-Computer Studies*, 60(1):17–63.

[Shinzato and Torisawa, 2004] Shinzato, K. and Torisawa, K. (2004). Acquiring hyponymy relations from Web documents. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2004, Boston, Massachusetts*, pages 73–80.

[Sibson, 1973] Sibson, R. (1973). Slink: An optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34.

[Siorpaes and Hepp, 2007] Siorpaes, K. and Hepp, M. (2007). Ontogame: Towards overcoming the incentive bottleneck in ontology building. In Meersman, R., Tari, Z., and Herrero, P., editors, *OTM Workshops (2)*, volume 4806 of *Lecture Notes in Computer Science*, pages 1222–1232. Springer.

[Siorpaes and Hepp, 2008] Siorpaes, K. and Hepp, M. (2008). Ontogame: Weaving the semantic web by online games. In Bechhofer, S., Hauswirth, M., Hoffmann, J., and Koubarakis, M., editors, *The Semantic Web: Research and Applications, 5th European Semantic Web Conference, ESWC 2008*, volume 5021 of *Lecture Notes in Computer Science*, pages 751–766. Springer.

[Smadja and McKeown, 1990] Smadja, F. A. and McKeown, K. (1990). Automatically extracting and representing collocations for language generation. In *roceedings of the 28th annual meeting on Association for Computational Linguistics*, pages 252–259.

[Smith and Mark, 1999] Smith, B. and Mark, D. (1999). Ontology with human subjects testing: An empirical investigation of geographic categories. *American Journal of Economics and Sociology*, 58(2):245–272.

[Sneath and Sokal, 1973] Sneath, P. H. and Sokal, R. R. (1973). *Numerical Taxonomy: The Principles and Practice of Numerical Classification.* W.H. Freeman, San Francisco.

[Staab and Hotho, 2005] Staab, S. and Hotho, A. (2005). Semantic web and machine learning tutorial. Tutorial at ICML 2005.

[Steinbach et al., 2000] Steinbach, M., Karypis, G., and Kumar, V. (2000). A comparison of document clustering techniques. In *Proceedings of the KDD International Workshop on Text Mining*, Boston, MA, USA.

[Suchanek et al., 2006] Suchanek, F., Ifrim, G., and Weikum, G. (2006). LEILA: Learning to extract information by linguistic analysis. In Buitelaar, P., Cimiano, P., and Loos, B., editors, *Proceedings of the 2nd Workshop on Ontology Learning and Population (OLP2) at COLING/ACL 2006*, pages 18–25, Sydney, Australia. Association for Computational Linguistics.

[Suh et al., 2006] Suh, S., Halpin, H., and Klein, E. (2006). Extracting common sense knowledge from wikipedia. In *Proceedings of the ISWC2006 Workshop on Web Content Mining with Human Language technology*.

[Sung et al., 2008] Sung, S., Chung, S., and McLeod, D. (2008). Efficient concept clustering for ontology learning using an event life cycle on the web. In *Proceedings of the 2008 ACM symposium on Applied computing, SAC 2008*, pages 2310–2314, New York, NY, USA. ACM.

[Sure et al., 2006] Sure, Y., Tempich, C., and Vrandecic, D. (2006). Ontology engineering methodologies. In Davies, J., Studer, R., and Warren, P., editors, *Semantic Web Technologies: Trends and Research in Ontology-based Systems*, chapter 9, pages 171–190. Wiley, Chichester, West Sussex, England.

[Surowiecki, 2004] Surowiecki, J. (2004). *The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations*. Doubleday.

[Suryanto and Compton, 2001] Suryanto, H. and Compton, P. (2001). Discovery of ontologies from knowledge bases. In *Proceedings of the 1st international conference on Knowledge capture, K-CAP 2001*, pages 171–178, New York, NY, USA. ACM.

[Tijerino et al., 2005] Tijerino, Y. A., Embley, D. W., Lonsdale, D. W., Ding, Y., and Nagy, G. (2005). Towards ontology generation from tables. *World Wide Web*, 8(3):261–285.

[Turney, 2001] Turney, P. D. (2001). Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the Twelfth European Conference on Machine Learning, ECML-2001*.

[van der Plas and Tiedemann, 2006] van der Plas, L. and Tiedemann, J. (2006). Finding synonyms using automatic word alignment and measures of distributional similarity. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 866–873, Sydney, Australia. Association for Computational Linguistics.

[Velardi et al., 2001a] Velardi, P., Fabriani, P., and Missikoff, M. (2001a). Using text processing techniques to automatically enrich a domain ontology. In *FOIS '01: Proceedings of the international conference on Formal Ontology in Information Systems*, pages 270–284, New York, NY, USA. ACM Press.

[Velardi et al., 2001b] Velardi, P., Missikoff, M., and Basili, R. (2001b). Identification of relevant terms to support the construction of domain ontologies. In *Proceedings of the workshop on Human Language Technology and Knowledge Management*, pages 1–8, Morristown, NJ, USA. Association for Computational Linguistics.

[Vidur Apparao, 1998] Vidur Apparao, Steve Byrne, M. C. S. I. I. J. A. L. H. G. N.-J. R. R. S. C. W. L. W. (1998). Document object model level 1, DOM, W3C recommendation, 1 october 1998. http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001/.

[Völker et al., 2007] Völker, J., Hitzler, P., and Cimiano, P. (2007). Acquisition of owl dl axioms from lexical resources. In Franconi, E., Kifer, M., and May, W., editors, *Proceedings of the 4th European conference on The Semantic Web, ESWC 2007*, volume 4519 of *Lecture Notes in Computer Science*, pages 670–685. Springer.

[Volz et al., 2003] Volz, R., Oberle, D., Staab, S., and Studer, R. (2003). Ontolift prototype, wonderweb deliverable d11. Technical report.

[Voorhees, 1986] Voorhees, E. M. (1986). Implementing agglomerative hierarchical clustering algorithms for use in document retrieval. *Information Processing and Management*, 22(6):465–476.

[Vuong et al., 2006] Vuong, L. P. B., Gao, X., and Zhang, M. (2006). Data extraction from semi-structured web pages by clustering. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2006*, pages 374–377, Washington, DC, USA. IEEE Computer Society.

[Wermter and Hahn, 2005a] Wermter, J. and Hahn, U. (2005a). Finding new terminology in very large corpora. In *Proceedings of the 3rd international conference on Knowledge capture, K-CAP 2005*, pages 137–144, Banff, Alberta, Canada. ACM Press.

[Wermter and Hahn, 2005b] Wermter, J. and Hahn, U. (2005b). Paradigmatic modifiability statistics for the extraction of complex multi-word terms. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT 2005*, pages 843–850, Morristown, NJ, USA. Association for Computational Linguistics.

[Wermter and Hahn, 2006] Wermter, J. and Hahn, U. (2006). You can't beat frequency (unless you use linguistic knowledge): a qualitative evaluation of

association measures for collocation and term extraction. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, ACL-44*, pages 785–792, Morristown, NJ, USA. Association for Computational Linguistics.

[Widdows and Dorow, 2002] Widdows, D. and Dorow, B. (2002). A graph model for unsupervised lexical acquisition. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics.

[Witschel, 2005] Witschel, H. F. (2005). Terminology extraction and automatic indexing – comparison and qualitative evaluation of methods. In *7th International Con-ference on Terminology and Knowledge Engineering, TKE 2005*, pages 363–Ű374.

[Witten and Frank, 2005] Witten, I. H. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, CA, USA, 2nd edition.

[Wu and Zhou, 2003] Wu, H. and Zhou, M. (2003). Synonymous collocation extraction using translation information. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, ACL 2003*, pages 120–127, Morristown, NJ, USA. Association for Computational Linguistics.

[Xiao and Rösner, 2004] Xiao, C. and Rösner, D. (2004). Detecting multiword verbs in the english sublanguage of medline abstracts. In *Proceedings of the 20th international conference on Computational Linguistics, COLING 2004*, page 861, Morristown, NJ, USA. Association for Computational Linguistics.

[Xiao and Rösner, 2004] Xiao, C. and Rösner, D. (2004). Finding high-frequent synonyms of a domain-specific verb in english sub-language of medline abstracts using wordnet. In *Proceedings of the 2nd International Conference of the Global WordNet Association, GWC 2004*, pages 242–247.

[Ye and Chua, 2006] Ye, S. and Chua, T.-S. (2006). Learning object models from semistructured web documents. *IEEE Transactions on Knowledge and Data Engineering*, 18(3):334–349.

[Yu et al., 2002] Yu, H., Hatzivassiloglou, V., Friedman, C., Rzhetsky, A., and Wilbur, J. W. (2002). Automatic extraction of gene and protein synonyms from medline and journal articles. In *Proceedings AMIA Symp*, volume 23, pages 919–923.

[Zhang et al., 2003] Zhang, Z., Li, R., Cao, S., and Zhu, Y. (2003). Similarity metric for XML documents. In Ralph B, M. S., editor, *Proceedings of Workshop on Knowledge Experience and Management, FGWM 2003)*, pages 255–261, Karlsruhe, Germany. AIFB Karlsruhe, GI.

[Zhao and Karypis, 2002] Zhao, Y. and Karypis, G. (2002). Evaluation of hierarchical clustering algorithms for document datasets. In *Proceedings CIKM*, pages 515–524, New York, NY. ACM Press.

[Zhou, 2007] Zhou, L. (2007). Ontology learning: state of the art and open issues. *Information Technology and Management*, 8(3):241–252.

[Zipf, 1949] Zipf, G. K. (1949). *Human Behavior and the Principle of Least Effort*. Addison Wesley, Cambridge MA.

[Ziqi Zhang and Ciravegna, 2008] Ziqi Zhang, Jose Iria, C. B. and Ciravegna, F. (2008). A comparative evaluation of term recognition algorithms. In *Proceedings of the Sixth International Language Resources and Evaluation , LREC 2008*, Marrakech, Morocco.