A Semantic Approach for Extracting Domain Taxonomies from Text

Kevin Meijer, Flavius Frasincar*, Frederik Hogenboom

Erasmus University Rotterdam, P.O. Box 1738, NL-3000 DR, Rotterdam, the Netherlands

Abstract

In this paper we present a framework for the automatic building of a domain taxonomy from text corpora, called Automatic Taxonomy Construction from Text (ATCT). This framework comprises four steps. First, terms are extracted from a corpus of documents. From these extracted terms the ones that are most relevant for a specific domain are selected using a filtering approach in the second step. Third, the selected terms are disambiguated by means of a word sense disambiguation technique and concepts are generated. In the final step, the broader-narrower relations between concepts are determined using a subsumption technique that makes use of concept co-occurrences in text. For evaluation, we assess the performance of the ATCT framework using the semantic precision, semantic recall, and the taxonomic F-measure that take into account the concept semantics. The proposed framework is evaluated in the field of economics and management as well as the medical domain.

Keywords: Taxonomy learning, word sense disambiguation, term extraction, subsumption method, semantic taxonomy evaluation

1. Introduction

In a world where the amount of digital data grows over more than 50% per year, any means to structure this data becomes increasingly relevant [19]. Knowledge management and decision making tasks more and more rely on such unstructured data and its derived, structured knowledge. One way to deal with the growing amount of data is by using taxonomies. A taxonomy is a concept hierarchy in which the broader-narrower relations between different concepts are stored. Taxonomies have proven useful for information search, classification, navigation, etc. [2], and hence can be exploited in decision support systems.

Manually creating a taxonomy, however, remains a difficult and time consuming process. In order to be able to construct high quality taxonomies, a massive amount of knowledge is required [5, 13]. Even if the required knowledge is available, it remains a tedious task to organize a high number of concepts in a proper manner. Therefore it is interesting to find ways to automatically build taxonomies [40]. Based on the availability of large text corpora one can investigate the construction of taxonomies from text, using techniques stemming from the closely-related field of terminology engineering [1, 10, 11, 24]. Such automatic taxonomy construction can greatly support the knowledge acquisition phase during the development of a knowledge-intensive decision support system. As the knowledge acquisition is fully automatic, it seamlessly provides up-to-date knowledge in a decision process, which can be of use in real-time business in a wide variety of tasks. For instance, taxonomies can support query formulation targeting at for instance finding articles on a certain theme [3], or can support recommendation systems [39]. Moreover, (automatically built) taxonomies can be employed in faceted search applica-

*Corresponding author; tel: +31 (0)10 408 1340; fax: +31 (0)10 408 9162
Email addresses: kmeijer@hotmail.com (Kevin Meijer), frasincar@ese.eur.nl (Flavius Frasincar), fhogenboom@ese.eur.nl (Frederik Hogenboom)

tions [43], or they can be used for summarizing information from different text-based data sources [7]. Last, a common application of taxonomies is in the filtering, enriching, or improving the quality of the data used in support systems [12, 25].

To automatically create a taxonomy from text corpora, first, terms need to be extracted. These extracted terms form the lexical representations of the concepts of the taxonomy that is to be built. After the concept lexical representations are determined, the concepts need to be stored in a concept hierarchy to form a taxonomy, which requires the use of clustering techniques. An intermediate step called word sense disambiguation (WSD) may also be applied to ambiguous simple terms. WSD is the process of deriving the sense in which terms are used in text. For example, the term 'return' may refer to a tennis stroke, but also to a return of money arising from economic transactions. By applying WSD, the taxonomy terms thus have associated a meaning which removes their possible ambiguity. This disambiguation allows for improved concept definition.

A common way of evaluating automatically built taxonomies is by applying a golden standard evaluation [21, 9], in which a constructed taxonomy is compared to a benchmark taxonomy. In the past such evaluation, however, only has taken place on a lexical level. As the terms in the benchmark taxonomy are ambiguous, evaluation is limited to comparing the lexical representations of taxonomy concepts. Because of these representations, it might occur that taxonomy concepts that are having the same lexical representations but that are semantically different, are considered to be the same. To prevent this situation, one can apply a semantic comparison of taxonomies. For this purpose, the concepts of the benchmark taxonomy first need to be disambiguated. In our current endeavors, we present an approach that enables the taxonomy evaluation on a semantic level. In order to be able to apply a semantic evaluation, on both the constructed taxonomy and the benchmark taxonomy, WSD is applied. The main focus of this work is on the use of WSD in the process of automatic taxonomy construction.

In this paper, we present a framework using a semantic approach for the automatic construction of domain taxonomies, called Automatic Taxonomy Construction from Text (ATCT). In the ATCT framework WSD is incorporated. The text corpora that are used to extract terms are the text corpus of RePub[1] and the text corpus of RePEc[2]. RePub is a repository of documents from the Erasmus University Rotterdam, the Netherlands. It contains documents from different domains such as economics, health, law, and psychology. RePEc is an online database which solely contains economic articles collected by volunteers from 76 countries. We have selected both corpora because they are tagged specifically for two domains of interest, i.e., economics and management, and health and medicine.

Two taxonomies are constructed, one for the domain of economics and management, and the other one for the domain of health and medicine. The taxonomy that is constructed for the domain of economics and management uses a total of 25,000 documents from RePub and RePEc. The medicine and health taxonomy uses a total of 10,000 documents from RePub only.

Furthermore, we introduce a new method for disambiguating taxonomy concepts. The application of this method allows for the semantic evaluation of the built taxonomy. The taxonomy for economics and management is semantically evaluated using the STW Thesaurus for Economics and Business Economics[3] as the benchmark taxonomy. The taxonomy for medicine and health on the other hand is evaluated using the MeSH taxonomy[4], which is a large ontology used for arranging medical subject headings.

The contributions of this paper are six-fold. First, we provide a semantic approach for taxonomy construction from text.

[1] Available at `http://repub.eur.nl/`
[2] Available at `http://repec.org/`
[3] Available at `http://zbw.eu/stw/`
[4] Available at `http://onto.eva.mpg.de/obo/mesh.owl`

Second, we define new evaluation measures, i.e., the semantic precision and semantic recall. Third, the framework as presented in the paper makes use of WSD for both the text corpus as well as the reference ontology (used for evaluation) in order to better define the meaning of concepts. Fourth, we investigate taxonomy construction from text corpora for the field of economics and management, a domain which has not been previously considered for this task in the literature. Fifth, we present a detailed evaluation of the different steps used in our taxonomy construction framework. Last, we refine an existing subsumption method [36] using concept semantics.

The rest of the paper is structured as follows. First, related work in the area of automatic taxonomy construction from text corpora is reviewed in Sect. 2. Then, the ATCT framework and its implementation is introduced in Sects. 3 and 4. Subsequently, the taxonomies built using our ATCT implementation are evaluated in Sect. 5. Last, we provide a summary of our research, as well as future work directions in the field of automatic taxonomy construction from text in Sect. 6.

2. Related work

In this section we discuss the current body of literature in the field of automatic taxonomy construction from text. A vast amount of research has been done in this area, and existing works differ in various ways. In general, three different aspects of taxonomy extraction can be distinguished, which are addressed in this section. For each of these aspects we infer the main approaches. First, various methods that have been applied to extract the terms used in taxonomies are described. Then, a review of methods to construct the broader-narrower relation between concepts is presented. Last, previous work concerning the evaluation of the built taxonomies is given. Also, we elaborate on word sense disambiguation techniques that can be applied in automatic domain taxonomy construction processes, and last, we summarize the section with a general discussion on existing extraction methods with respect to our proposed methodology.

2.1. Term Extraction

Several methods are available to extract terms from a set of documents. These methods can be broadly categorized into three different approaches: linguistic approaches, statistical approaches, and hybrid approaches.

Linguistic methods use natural language processing (NLP) for term extraction. A linguistic method is part-of-speech tagging [42]. A part-of-speech (POS) tagger labels the part-of-speech (e.g., adjective, noun, verb, etc.) of terms appearing in a text. Another technique is morphological analysis. This technique is used to derive a term's form, e.g., whether a term is used in singular or plural form, the term's inflection, etc. One can also extract terms by using lexico-syntactic patterns, which analyze relations between terms to possibly retrieve new terms [17]. An important feature of linguistic techniques is their ability to define the grammatical functions of terms in sentences. When extracting terms for a certain domain, they however do not consider the relevance of a term for that domain.

Cimiano et al. [6] propose a novel linguistic approach that specifically focuses on verbs. The authors assume that verbs limit the semantic content of their arguments, and hence can be exploited for building conceptual hierarchies by using the inclusion relations between the extensions of the verbs' selectional restrictions. The discussed method relies solely on generic NLP tools for determining the part-of-speech, and hence can be classified as a linguistic method.

Differently than the linguistic approaches, statistical methods do not use the linguistic characteristics of terms, but rely solely on statistical measures to extract terms. These statistical methods are applied to acquire the relevance of a term

for a domain. One popular statistical method is the term frequency - inverse document frequency (TF-IDF) [34] measure. This method uses the frequency of a term in a domain corpus document (the term frequency) and the inverse number of corpus documents in which the term appears (the inverse document frequency). The higher the term frequency is in comparison with the document frequency, the more relevant a term is according to the TF-IDF measure. It might occur that a relevant term appears often in corpus documents and thus might not be selected as a relevant term. To prevent such a situation a non-stopping word list can be used [15], on which terms are listed that should never be filtered out.

The authors of [45] provide an example of frequency-based taxonomy extraction for mining characteristic phrases (i.e., sequential patterns) that describe documents. In their extraction phase, meaningless sentences are removed, based on the amount of occurrences within the same paragraph. Another example of a statistical approach to term extraction that does not exploit linguistic characteristics is presented in [26]. Maedche and Volz extract terms from text using several statistical and data mining-based algorithms, mainly based on term frequencies. The outputs of these algorithms are subsequently used for creating concepts and their lexical representations, which can be used in following steps for deriving concept hierarchies. Alternatively, Google page counts can be used [27]. These page counts serve as a substitute for term frequencies, and appear to work well when used for calculating term dependencies, and subsequently adjacencies (resulting in a taxonomy).

Hybrid extraction techniques combine linguistic techniques and statistical measures. An example of a hybrid method is the term filtering method presented in [38]. First, linguistic processing takes place, after which terms are filtered on multiple criteria, e.g., domain pertinence, domain consensus, lexical cohesion, and structural relevance. C/NC-value is a hybrid method that calculates a score for each term based on their length and their context (words surrounding the term) [15]. This method is used for improving the extraction of compound terms (terms that consist of multiple words). However, as terms with a higher length have priority in the selection process, this technique might not select important short terms.

Another hybrid method computes a χ-square value for each term after linguistic processing has taken place [44]. Terms with a χ-square value above a certain threshold value are selected as representant for a certain domain. A last example is the work presented in [35]. The authors propose a method for automatic taxonomy extraction from Web sources by employing different types of linguistic patterns for finding hyponyms, and by additionally using statistical measures for inferring information relevance.

2.2. Hierarchy Creation

After terms or concepts have been identified using the previously described methods, various methods are available that are able to create a hierarchy. For instance, formal concept analysis groups objects with their attributes [5]. By identifying the similar attributes of multiple objects, the relations between objects can be defined. Determining the attributes of objects from text is achieved by linking terms with verbs.

Hierarchical clustering starts with one cluster and progressively merges clusters that are closest to each other [23]. Measures to determine the distance between clusters are: average linkage, minimum linkage, and maximum linkage. A problem with hierarchical clustering is the labeling of the clusters. To label clusters, one can use the most specific hypernym of the terms in a cluster, or use the centroid of the cluster as the label. In [18], the authors apply word sense disambiguation to clustering. To disambiguate a certain term that appears in a given document the authors use the concept vicinity, which is defined as the set of direct sub- and super-concepts. By counting the

amount of terms in the document that could express a concept from the concept vicinity a term is disambiguated. In vector space model representations, every document has associated a vector of weights corresponding to the occurrence of concepts. For building a hierarchy of clusters, bi-section k-means, which is a partitional clustering algorithm, is used.

Maedche and Volz [26] also make use of hierarchical clustering for generating a concept hierarchy, accessing background knowledge from existing ontological entities to label the extracted hierarchy. Additionally, the authors propose a heuristic, regular expression-oriented pattern-based approach, relying heavily on the Saarbruecken Message Extraction System (SMES) NLP tool for processing German texts. The system applies basic NLP procedures such as tokenization, stemming, part-of-speech tagging, etc., but also performs morphological analysis, grouping, and sentence pattern recognition. Moreover, it is able to link stemmed words to ontological concepts.

The subsumption method constructs the concept broader-narrower relations based on the co-occurrence of concepts [36, 37]. If a concept co-occurs frequently with another concept, a parent-child relationship is created between the concepts. As only co-occurrence values are calculated and computations remain simple, the subsumption method allows for a fast creation of broader-narrower relations between concepts.

Classification methods have also been proposed that add concepts to already existing concept hierarchies. In the tree-descending algorithm a term is added to a hierarchy by descending the hierarchy from the root to the leaf. The term is added as a child of the leaf node that is on the path with the highest sum of similarities with the to-be-classified node [32].

In the tree-ascending algorithm a combination of distributional similarity measures (similarity of two concepts in text corpora) and taxonomic similarities (similarity of concepts in a taxonomy structure) is computed to give a node in the hierarchy a certain number of votes. The taxonomic similarity between a to-be-added term and a hierarchy concept label is computed by first retrieving the lowest common subsumer of the two terms. The closer the lowest common subsumer is to the two terms, and the further it is away from the hierarchy root node, the higher the taxonomic similarity. After computing for each node the amount of votes using a distributional similarity measure and a taxonomic similarity, a to-be-added concept is inserted as a child of the node with the highest number of votes [32].

2.3. Evaluation

In order to determine the quality of a constructed taxonomy, two types of evaluation may be applied. One could use the golden standard approach [5, 9, 36]. In this approach the created taxonomy is compared to a benchmark taxonomy, which is usually a manually built taxonomy made by one or more experts. The two taxonomies are lexically compared based on concept representation using the lexical precision and lexical recall measures. The constructed taxonomy and benchmark taxonomy can also be compared based on the broader-narrower relations present in the taxonomies by applying the taxonomic precision and taxonomic recall measures. To determine the similarity of the broader-narrower relations of the two taxonomies, specific semantic evaluation measures are used. Examples of specific semantic evaluation measures are the semantic cotopy (SC) (used in, e.g., [6]) and common semantic cotopy (CSC) [5]. The SC is a collection of a concept and all its sub- and super-concepts. The CSC is the collection of a concept and the sub- and super-concepts that are shared by the built taxonomy and the benchmark taxonomy.

When no benchmark domain taxonomy is available to evaluate a constructed domain taxonomy, one can also use several domain experts to manually evaluate the built taxonomy. By averaging their individual judgments the quality of the built tax-

onomy can be determined without the use of a benchmark taxonomy. A problem with this sort of evaluation is that it might be difficult to find a group of domain experts for judging the constructed taxonomy.

2.4. Word Sense Disambiguation

Although to our knowledge, word sense disambiguation (WSD) is generally not employed for automatic taxonomy generation, we argue that it is a crucial step in the generation of taxonomies, as for extracted terms, word sense disambiguation may be applied for identifying term meaning. This could help in identifying duplicate terms that represent the same concepts, or by distinguishing between multiple concepts that have the same lexical representation. Although such procedures are very useful, it should be noted that this only holds for non-compound terms, as compound terms usually have only one meaning. Next, we discuss four different WSD methods, of which one is an unsupervised method and the other three are supervised methods. The difference between supervised and unsupervised methods is that supervised methods use training data to train classifiers and subsequently disambiguate terms from a test set by using these classifiers, whereas unsupervised methods do not require this information.

2.4.1. Methods

An unsupervised WSD method is Structural Semantic Interconnections (SSI) [30]. This method disambiguates terms by computing what sense of the term has the highest similarity with its context, the senses of the terms surrounding the current term. The context of a term can be the sentence the term appears in, but also the paragraph or document in which the term is used. A context list is initialized by taking the senses of monosemous terms (terms with only one meaning). If no monosemous terms are available in the context, the most common sense of the least ambiguous term is selected.

In the supervised method named GAMBL [8] first a training text is analyzed linguistically. For the terms that appear in the training text the method checks if it is a term with multiple senses and if the term has a frequency in the training text above a specified threshold value. If a term has a frequency below the specified threshold value, or if the term is monosemous, the most frequent or the unique sense is assigned to the term. If a term has multiple senses and has a frequency above the threshold value, so-called expert modules, which are classifiers specialized in assigning the proper sense to an ambiguous term, are trained and used for disambiguating terms.

The supervised Naive Bayes method uses feature probabilities and prior probabilities to select a term's sense [46]. A feature probability is the proportion of times the sense and a feature associated with the sense, such as a word that often appears close to the sense in the text, have been found together in the training data. The prior probability is the proportion of times a sense was associated with a term in the training data. The sense maximizing the product of feature probabilities and the prior probability is selected as the sense of a term.

Another supervised WSD method is SenseLearner [28]. This method is minimally supervised, as it only uses a relatively small amount of training data and makes generalizations of concepts learned from the training data by using the semantic network of a semantic lexicon to also be able to disambiguate terms in the test data set that did not appear in the training data. Therefore this algorithm does not require as much data as the other considered supervised WSD methods.

2.4.2. Similarity measures

Many similarity measures are available to determine the similarity between terms. For instance, Resnik's similarity measure uses the information content of terms [33]. The information content is defined as the degree to which terms share information. If a term has a high probability of appearing in a specific

corpus, then the information content of that term is low. The reasoning behind this is that the more often a term appears, the more general it is, and therefore the less informative it is. The similarity between two terms is determined by taking the information content of the lowest common subsumer of the two terms. As only the information content of the lowest common subsumer is calculated, Resnik's measure proves to be a fast to compute similarity method.

Jiang and Conrath's similarity measure goes one step further [20]. This measure does not only take the information content of the lowest common subsumer of two terms into account, but it also uses the information content of the two terms for which the similarity is measured. The results of this measure are more accurate than those of Resnik's similarity measure, while it also is a fast to compute similarity method [4].

Another similarity measure is a window-based similarity. This measure uses the frequency and probability of two terms appearing in a window consisting of words that appear in sequence. The higher the probability that the two terms appear in a window, the higher their similarity [31].

The work presented in [31, 16] is a Web-based method, which utilizes a search engine to determine the similarity between two terms by counting the number of retrieved pages for the two terms. The higher the returned number of pages with two terms, the higher the similarity between the two terms. The advantage of this method is that it has access to a vast amount of data, as it uses information available on the Web as its data source.

2.5. Discussion

We have introduced various common methods for three aspects of automatic taxonomy extraction from text, i.e., for term extraction, for relation extraction and hierarchy construction, and for evaluating the generated taxonomies. As demonstrated, most works make use of either linguistic (lexical) or statistical methods for term extraction, and some prefer a hybrid method. However, to our knowledge, none of them seems to be able to deal with semantic representations. Furthermore, word sense disambiguation is generally not employed for term extraction, although a disambiguation procedure could improve the taxonomy quality by merging synonyms and by distinguishing homonyms. Last, the evaluation seems to be weak in most of the investigated cases, as it is generally done on a lexical level, and not on a semantic level.

Despite the proven advantages of linguistic, statistical, and hybrid methods in terms of performance and interpretability of the results, given the issues pointed out above, we propose a semantic approach to taxonomy construction. As most linguistic methods, we rely heavily on a set of initial NLP procedures. Our method differs from the existing (linguistic) works in that we aim to better define the meaning of concepts by employing an additional disambiguation step (based on an improved version of the SSI algorithm) for both the extracted taxonomy and the reference ontology used for evaluation. For hierarchy creation, we adopt the subsumption method due to its proven performance, yet we refine the method using concept semantics which improves accuracy, as found in [7]. In terms of evaluation, we make use of the commonly used golden standard approach and the taxonomic precision and recall measures (based on common semantic cotopy scores). Additionally, we propose new evaluation measures, i.e., the semantic precision and recall, in order to better take into account the concept semantics.

Because of the changes made in both the term extraction and the hierarchy creation steps, it would be interesting to compare our method to other related works. However, a comparison easily becomes rather problematic, as the state-of-the-art approaches differ in required inputs and target domains, thwarting a fair comparison. Moreover, as we propose additional semantics-based measures (next to the commonly used taxo-

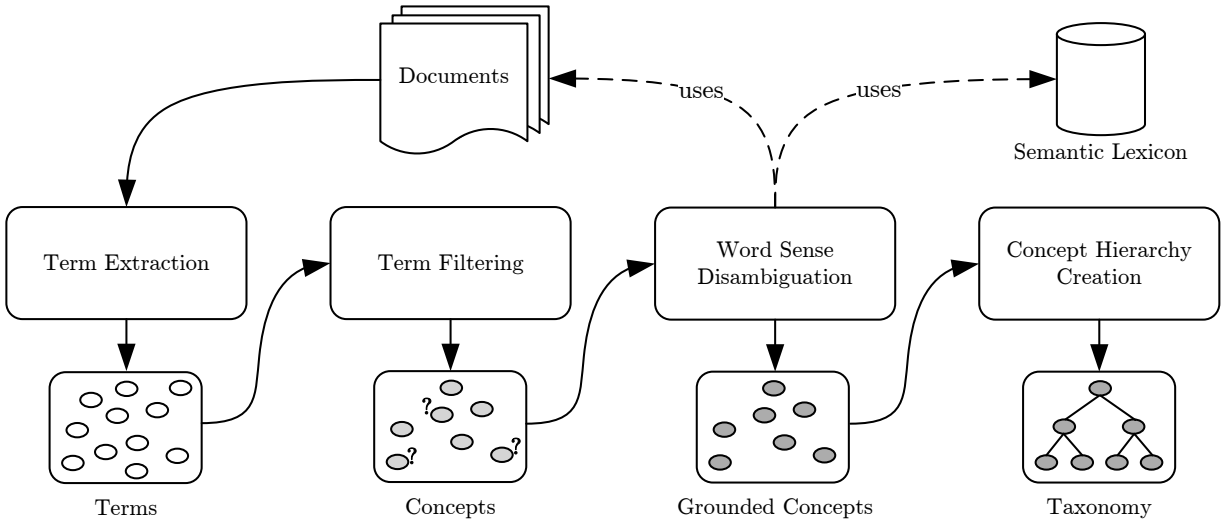

Figure 1: Overview of the ATCT framework

nomic precision and recall), we are unable to compare our proposed method against other works, due to the fact that in the literature, evaluation has taken place on a lexical level rather than on a semantic level.

3. ATCT Framework

In this section we present our framework for automatically constructing a domain taxonomy, i.e., Automatic Taxonomy Construction from Text (ATCT). An overview of the framework is shown in Fig. 1, which depicts the sequence of the different data processing steps, as well as the input and output of these steps.

The ATCT framework consists of four main steps. First, *term extraction* takes place to extract terms from a corpus of text. These terms are then filtered on multiple criteria and selected in the *term filtering* step. The selected terms are stored as labels of concepts. The concepts however do not have a meaning yet. Therefore the concept labels are disambiguated by subsequently applying *word sense disambiguation* (WSD) based on the senses gathered from a semantic lexicon. To be able to perform WSD the context of terms needs to be known. For this purpose the corpus of text is used again. After disambiguation has taken place, the broader-narrower relations between the concepts are determined to create the *concept hierarchy*. This hierarchy is finally stored in an ontology with a SKOS [29] vocabulary to form the taxonomy.

This section continues by providing a detailed description of each of the previously introduced framework components. First, the term extraction is discussed. Then, the process of term filtering will be explained. Subsequently, we describe how WSD is applied. Last, the approach used for creating a broader-narrower concept hierarchy is presented.

3.1. Term Extraction

The first step in building a taxonomy is to extract the terms from a text corpus. The terms that are extracted are the nouns that appear in the text documents. We choose to extract nouns as many existing taxonomies consist of concepts that are labeled by nouns [15] and to properly evaluate the built taxonomy with the two reference ontologies, i.e., the earlier introduced STW taxonomy and MeSH taxonomy, which also use nouns for labeling concepts. To be able to acquire nouns from

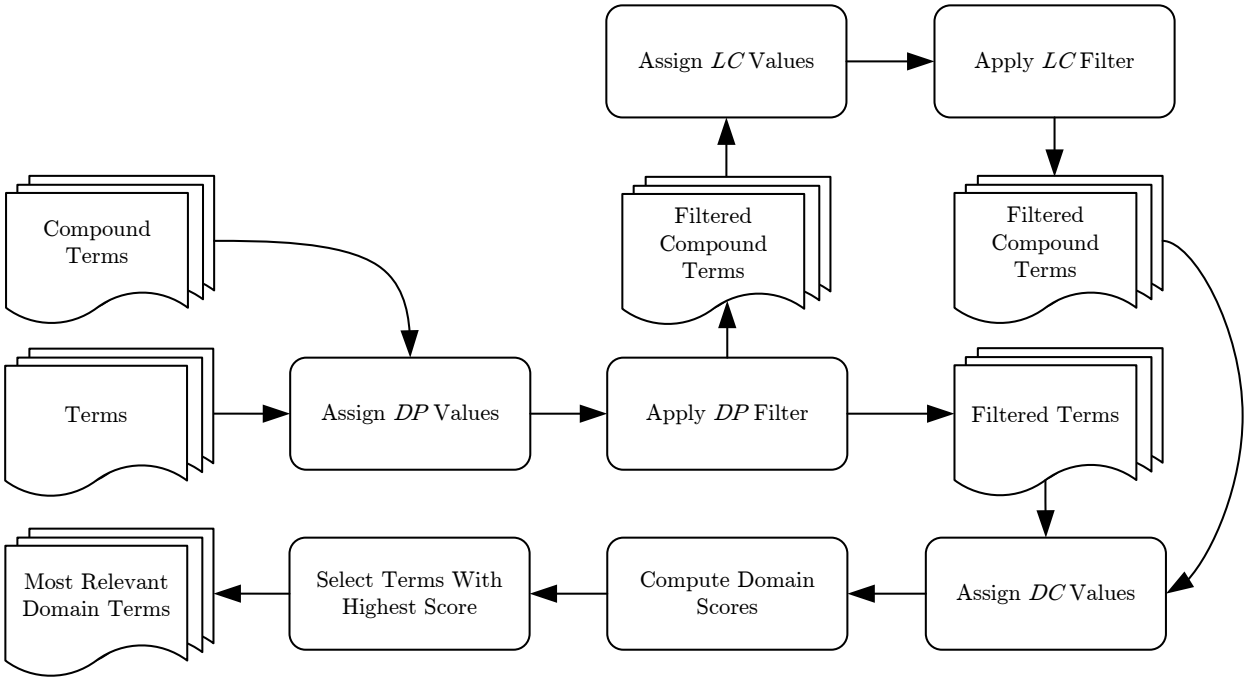

Figure 2: Overview of the term filtering process

text, we make use of a part-of-speech tagger that tags words that appear in the text [22].

3.2. Term Filtering

The most relevant terms for a specific domain need to be selected from the previously extracted terms. In order to achieve this goal, we apply a filtering approach in which filters are organized in a pipeline. We use four measures, each of which is based on one of the filters defined in [38]. On the basis of values obtained from these measures, a score is computed. This score is used to determine the relevance of a term. An overview of the term filtering process is depicted in Fig. 2.

The first measure that is applied is the domain pertinence (DP) measure. The domain pertinence is a measure that is used to acquire terms that are representative for a certain domain corpus, and not for other contrastive corpora. It is defined as follows:

$$DP_{D_i}(t) = \frac{freq(t/D_i)}{\max_j(freq(t/D_j)}\ , \tag{1}$$

where $freq(t/D_j)$ denotes the count of term t in domain corpus D_i and D_j represents a contrastive corpus. The more frequently a term appears in the domain corpus, and the less frequently a term appears in the contrastive corpus, the higher the domain pertinence value.

The second measure that is used is the lexical cohesion (LC) measure. This measure is solely applied on compound terms, terms that consist of more than one word. The lexical cohesion measure is used to determine how well the combination of individual compound term words represent a compound term. This is achieved by examining the compound term itself, as well as the separate words in the compound term. The definition of the lexical cohesion measure is as follows:

$$LC_{D_i}(t) = \frac{n \cdot freq(t/D_i) \cdot \log(freq(t/D_i))}{\sum_{w_j \in t} freq(w_j/D_i))}\ , \tag{2}$$

where n is the amount of words in compound term t, w_j is a word within the compound term, and D_i is a domain corpus. The frequency of the compound term in the domain corpus is compared to sum of the frequencies of the individual words in

the domain corpus. The higher the frequency of the compound term is in comparison with the sum of frequencies of the separate words, the higher the lexical cohesion value.

The third measure that is applied is the domain consensus (DC) measure. The domain consensus measure is used to determine if a term appears frequently across the domain corpus documents. It is defined as follows:

$$DC_{D_i}(t) = -\sum_{d_k \in D_i} n\_freq(t, d_k) \cdot \log(n\_freq(t, d_k)) \,, \quad (3)$$

where $n\_freq(t, d_k)$ is the normalized frequency of term t in document d_k, which is a document in domain corpus D_i, and $\log(n\_freq(t, d_k))$ is used to relatively decrease higher document frequencies compared to low document frequencies. To obtain a normalized frequency, the calculated frequency of term t is divided by the maximum frequency of term t in any domain corpus document.

The fourth measure is the structural relevance. Terms that appear in a domain corpus document's title are generally more representative for the domain. To take this into account in selecting the most relevant domain terms, each term that appears in a title of a domain corpus document is considered more important when determining the domain terms.

Two of the previously described measures are used for filtering out terms. After the domain pertinence value is computed for each term, a certain percentage of terms with the lowest domain pertinence values is filtered out. The domain pertinence is a relatively good measure for retrieving terms representative for a specific domain. Terms with a low domain pertinence value are not relevant for the domain, and are therefore filtered out.

A percentage of compound terms that have a low lexical cohesion value is also filtered out. When a compound term has a low lexical cohesion, the compound term is not a widely used and meaningful compound term. As we do not allow such terms in our taxonomy, we filter out the compound terms with the lowest lexical cohesion values.

Based on three of the previously described measures a score is obtained. The domain score of term t that appears in domain corpus D_i is acquired as follows:

$$score(t, D_i) = \alpha \frac{DP_{D_i}(t)}{max_t(DP_{Di}(t))} + \beta \frac{DC_{D_i}(t)}{max_t(DC_{Di}(t))} + k \,, \quad (4)$$

where α and β are weights that add more emphasis on either $DP_{D_i}(t)$ or $DC_{D_i}(t)$, k represents the structural relevance, $max_t(DP_{Di}(t))$ and $max_t(DC_{Di}(t))$ are the highest domain pertinence value and the highest domain consensus value found in domain corpus D_i, respectively. The latter two values are used to normalize the domain pertinence value and domain consensus value of term t, so that high domain pertinence or domain consensus values have less influence on the score and thus balance DP and DC. The terms with the highest scores are selected as concept labels that appear in the constructed domain taxonomy.

3.3. Word Sense Disambiguation

The terms that are selected in the previous term filtering step are possibly ambiguous. By applying word sense disambiguation (WSD), the terms are disambiguated and the meaning of the terms is derived. The ATCT framework uses WSD in two different ways: for disambiguating terms from a text corpus, and for disambiguating concepts in the reference taxonomy that is used for evaluation. First we discuss the WSD approach that is applied for disambiguating terms selected from a text corpus. Subsequently the WSD approach for concepts of a taxonomy is described.

3.3.1. WSD on Text Corpora

In order to be able to disambiguate terms, we must know the possible meanings of a term. For this we use a semantic lexicon, which is a dictionary that contains terms with their possible meanings and synonyms. A semantic lexicon also contains other relations than synonyms between different meanings like antonyms, hyponyms, and hypernyms. These relations can be used to compute the similarity between two meanings or synsets (as will be explained later).

To disambiguate the terms selected by the previously applied term filtering method, an approach that is based on the SSI algorithm is applied [30]. For each selected term the sense is determined by using a context list that consists of senses of terms that form the surrounding context of the current term. The sense of term t from the set of possible senses S_t is computed as follows:

$$sense_t = \max_{s_i \in S_t} \sum_{c_j \in C_t} sim(s_i, c_j) \ , \tag{5}$$

where s_i is a sense in the set of possible senses S_t, c_j is a context sense in the set of context senses C_t, and $sim(s_i, c_j)$ denotes the similarity between s_i and c_j.

The context list is initialized by taking the senses of monosemous terms. When a term is disambiguated, it is added to the context list. As more and more terms are disambiguated the context list thus becomes larger. Our approach differs from the SSI method when no monosemous terms are available to initialize the context list. If no monosemous terms are available the SSI method initializes the context list by taking the most common sense of the least ambiguous term. The most common sense however might not be the correct sense, especially when building taxonomies for a specific domain, therefore we use a different approach. We initialize the context list with each sense of the least ambiguous term. The context list is then built using Formula 5. The selected sense of the least ambiguous term is the one that results in the context list with the highest sum of pair-wise sense similarities.

The measure we use for computing similarities is the similarity measure proposed by Jiang and Conrath [20]. Research has pointed out that this measure provides better results than other measures such as Resnik's similarity measure [4].

As most of the terms that are disambiguated appear more than once in the text corpus, a term's sense is determined multiple times. We only allow one sense per term, the sense that is most representative for a specific domain. Therefore, the most frequently appearing sense is selected as the sense of a term. The reason for allowing only one sense per term is based on the fact that domain taxonomies (e.g., the reference taxonomy or benchmark taxonomy) have only one sense per term.

In the built taxonomy a sense is represented by only one concept. If a term is disambiguated and its retrieved sense is already represented by another term, the disambiguated term will only be present in the built taxonomy as an alternative label. For example, let us consider two concepts that appear to have the same sense: concept a with label x, and concept b with label y. If label y was assigned a lower domain score in the term filtering process than label x, concept b (that contains label y) is removed from the list of concepts. Label y is then only represented in the taxonomy as an alternative label of concept a. The documents in which concept b occurs are added to the documents of concept a. In this way, the shared meaning of labels x and y is now well presented in one concept.

3.3.2. WSD on Existing Taxonomies

Next to applying WSD on terms from text corpora, we also employ WSD for disambiguating existing taxonomies. By disambiguating a benchmark taxonomy, the evaluation of a constructed taxonomy is not limited to comparing lexical representations, but allows for comparing semantic representations as well.

As for the disambiguation performed for text corpora, for existing taxonomies we use the similarity measure proposed by Jiang and Conrath [20]. For disambiguating concepts we also apply the same algorithm, which is based on the SSI method [30]. The taxonomy concepts contain labels. These labels are disambiguated by using, for each label, a context list that contains previously disambiguated labels. The context list is initialized in the same way as for WSD applied on text corpora.

As the context of taxonomy concepts is not obtainable from a text corpus, the disambiguation of the concepts from a taxonomy requires a slightly different approach. Because the SSI algorithm is applied on a concept hierarchy with broader-narrower relations, the surrounding context of a lexical representation does not consist of other lexical representations that appear in the same document or sentence, but rather of other concepts in the hierarchy. The concepts that form the context list of an ambiguous concept are the disambiguated concepts that are closest to the ambiguous concept in the concept hierarchy. We define this collection of close concepts as the concept neighbourhood. The concept neighbourhood of concept x consists of ancestor concepts within a certain number of layers from x and descendant concepts within a certain number of layers from x.

Figure 3 shows an example of what the concept neighbourhood of 'labour market' is composed of. In this example the ancestors concepts that are within a distance of two layers from the ambiguous concept (with respect to a semantic lexicon) form the ancestor neighbourhood. The descendant concepts that are within a distance of two layers from the undisambiguated concept form the descendant neighbourhood. In the hierarchy depicted in Fig. 3 the concepts that form the concept neighbourhood are coloured blue (grey for black and white printing). The nodes (depicted by their labels) that form the ancestor neighbourhood nodes are 'economics' and 'labour', as these nodes are within a range of two layers upwards from 'labour market'. The nodes that form the descendant neighbourhood nodes are 'labour market theory', 'job-search the-

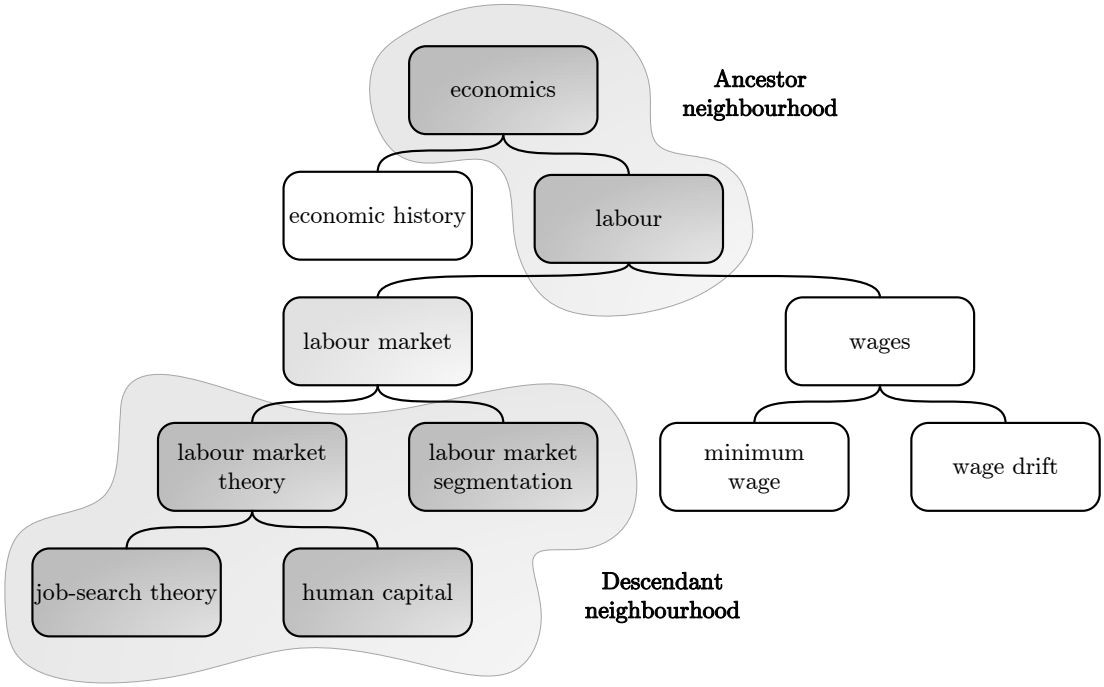

Figure 3: The concept neighbourhood of 'labour market'

ory', 'human capital' and 'labour market segmentation'. These nodes are within a range of two layers downwards from 'labour market'. The concept neighbourhood of 'labour market' thus is: {'economics', 'labour', 'labour market theory', 'job-search theory', 'human capital', 'labour market segmentation'}.

The sense of a taxonomy concept is selected by taking the sense which yields the highest sum of similarities with the concept senses in the concept neighbourhood. After a concept in the hierarchy is disambiguated, the next concept that will be disambiguated is the one with the most disambiguated concepts in its neighbourhood. The process continues until all ambiguous concepts with a lexical representation present in the used semantic lexicon are disambiguated.

The possible meanings of a term are retrieved from a semantic lexicon. As the semantic lexicon does not contain all domain terms with their associated meanings, not every term can be disambiguated. However, each term that is disambiguated can be better taken into account than non-disambiguated ones in the hierarchy creation since text data in which a term appears but in which the sense is different than the domain one is not considered anymore.

3.4. Concept Hierarchy Creation

The next step is to establish the broader-narrower relations between the disambiguated concepts. To construct these relations the subsumption method is used [36]. The subsumption method creates relations between concepts by calculating the co-occurrence of different concepts. The co-occurrence between concept x and concept y is measured as follows:

$$P(x \mid y) \geq t, P(y \mid x) < t \ , \tag{6}$$

where t is a co-occurrence threshold value. If x appears in at least the proportion t of all documents in which y appears and if y appears in less than the proportion t of all documents in which x appears, then x is considered a subsumer of y.

A single concept can have multiple subsumers if just Formula 6 would have been used. As only one subsumer for a single concept is allowed in our built taxonomy, we need to select one of these potential parent concepts as the subsumer of a concept. The reason for this restriction is that learnt taxonomies have a tree representation where one concept is subsumed by at most one other concept. For the subsumer selection, we introduce the following formula:

$$score(p, x) = P(p \mid x) + \sum_{a \in A_p} w(a, x) \cdot P(a \mid x) \ , \tag{7}$$

where p is a potential parent concept of x, A_p is the list of ancestors of p, and $w(a, x)$ is a weight value with which the conditional probability $P(a \mid x)$ of ancestor a given x is multiplied. This weight value is influenced by the distance between node x and ancestor a. The definition of the weight is as follows:

$$w(a, x) = \frac{1}{d(a, x)} \ , \tag{8}$$

where $d(a, x)$ is the distance (amount of layers) between node x and ancestor node a. The more distant ancestor node a is from node s, the lower the resulting weight $w(a, x)$ is. The nodes that are closest to node x should have the most impact on which potential parent node is selected. Therefore ancestor nodes that are distant have a lower weight value to reduce their influence in the parent selection procedure.

When a concept has multiple potential parent concepts, we calculate a score for each of these parent concepts. The concept that is selected as the parent is the potential parent concept that yields the highest score. Figure 4 depicts a concept hierarchy to which a concept labeled 'pricing behaviour' will be added.

The concept labeled 'pricing behaviour' has two potential parent concepts, 'pricing' and 'trading'. To select which concept should be the parent of 'pricing behaviour', a score is calculated for both potential parents by applying Formula (7).

In order to compute *score*('pricing', 'pricing behaviour') the following probabilities are required: *P*('pricing' | 'pricing behaviour'), *P*('price' | 'pricing behaviour'), and *P*('market' | 'pricing behaviour'). To calculate *score*('trading', 'pricing behaviour') we need to compute: *P*('trading' | 'pricing behaviour'), and *P*('market' | 'pricing behaviour'). Let us assume that in this example *P*('pricing' | 'pricing behaviour') = 0.6, *P*('price' | 'pricing behaviour') = 0.4, *P*('trading' | 'pricing behaviour') = 0.7, and *P*('market' | 'pricing behaviour') = 0.3. If we use these probabilities to compute the two scores, we obtain *score*('pricing' | 'pricing behaviour') = $0.6 + \frac{1}{2} \cdot 0.4 + \frac{1}{3} \cdot 0.3 = 0.9$, which is higher than *score*('trading' | 'pricing behaviour'), i.e., $0.7 + \frac{1}{2} \cdot 0.3 = 0.85$. Hence, the selected parent for 'pricing behaviour' is 'pricing'.

After the broader-narrower relation of each concept is set, the concept hierarchy is created. The main advantage of the subsumption method is its processing speed in combination with the ability to provide good concept broader-narrower relations. The subsumption algorithm can categorize concepts in a relatively short amount of time because of its simplicity and therefore is a good method to apply on large data sets. As the algorithm is able to handle large text inputs, it is suitable to create appropriate concept broader-narrower relations based on concept co-occurrence in text.

4. ATCT Implementation

This section discusses the implementation of the proposed ATCT framework. For this implementation the Java programming language is used. For nouns extraction from a text corpus we use the tagger created by Stanford [22], and word senses are disambiguated using the WordNet [14] semantic lexicon. The built taxonomy is exported as a SKOS [29] file. Furthermore, we utilized the Jena [41] framework for manipulating RDF representations in our implementation.

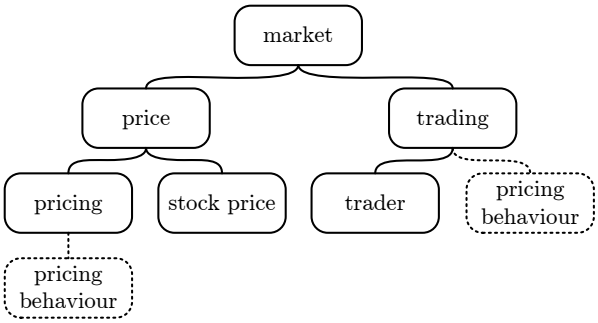

Figure 4: Hierarchy to which 'pricing behaviour' is added

The implementation of each main component of the ATCT framework is discussed below. First, the process of term extraction is explained. Then, the term filtering used for selecting the most relevant domain terms is discussed. Subsequently, the word sense disambiguation approaches for both text corpora and taxonomies are reviewed. Last, the built taxonomy for which the relations are created by the subsumption method is described. The taxonomy that is constructed in this implementation is for the domain of economics and management. As contrastive corpus we use medicine and health documents. To build the taxonomy we use 25,000 documents, of which 10,000 are from RePub, and 15,000 are from RePEc. Both RePub and RePEc suit our domains very well, as they have documents tagged specifically for these two domains. RePub is a repository from Erasmus University Rotterdam containing scientific papers written by academics from several domains, including economics and management, as well as medicine and health. RePEc consists of economic articles (i.e., research papers collected from journals as well as working paper series) collected by volunteers from 76 countries. We use 5,000 out of 25,000 documents from RePub as the contrastive corpus, while the remaining 20,000 documents are used as the domain corpus.

4.1. Term Extraction

For the taxonomy for economics and management we extract a total of 2,000 terms from the text corpora of RePub and RePEc. The terms we extract are the nouns that appear in the abstracts and titles of the documents from RePub and RePEc. To achieve this we use the Stanford part-of-speech tagger [22]. This part-of-speech tagger creates a tree from input text. This tree contains, among other things, nouns and compound nouns appearing in the input text. A drawback of the tagger is that it is relatively slow as it needs to create a detailed tree structure containing all the different types of words (e.g., verb, noun, adjective, etc.). The slow processing time however is compensated by the quality of the retrieved terms. The part-of-speech tagger performs well with respect to the extraction of compound terms, while faster methods often have difficulties with these terms [42].

4.2. Term Filtering

From the previously extracted terms we select the terms that are relevant for the domain of economics and management. The selected terms are represented in the to-be-built taxonomy as concept labels. To retrieve the most relevant domain terms we use multiple measures and assign a score to each term based on the values gained from these measures. First, we discuss the optimal parameters for retrieving the most relevant terms. Then, we show the results of the term filtering procedure based on the acquired optimal parameters.

4.2.1. Choosing the Parameters

There are multiple parameters in the term filtering process. A domain score is computed to determine whether a term is representative for a certain domain or not. But before the scores are computed for the extracted terms, terms are filtered out based on their domain pertinence (*DP*) and lexical cohesion (*LC*). All the extracted terms go through the *DP* filter. As the *DP* is a good measure to depict whether a term belongs to a specific domain and as the *DP* also plays a critical role in determining the score for a term, we filter out the 30% terms with the lowest calculated *DP* values.

Compound terms are also filtered on their *LC*. The *LC* performs well in determining compound terms. If a compound term has a low *LC* value that usually means that it is an unusual combination of words rather than a compound term. We do not allow such terms in our constructed taxonomy and therefore filter out the 30% compound terms with the lowest *LC* values.

For the remaining terms that are not filtered out we compute a score to determine their relevance for the domain of economics and management. The weight value for domain pertinence (α), the weight value for domain consensus (β), and the structural relevance (k) have an influence on what terms are selected as domain terms. To select the optimal values for these weights, we have examined the influence of variations in α, β, and k on the output of a set containing 25,000 documents. We have experimented with α and β values from 0 to 1 and 1 to 0, respectively, with a step of 0.1, and k from 0 to 0.5 with a step of 0.01. To investigate the quality of the selected domain

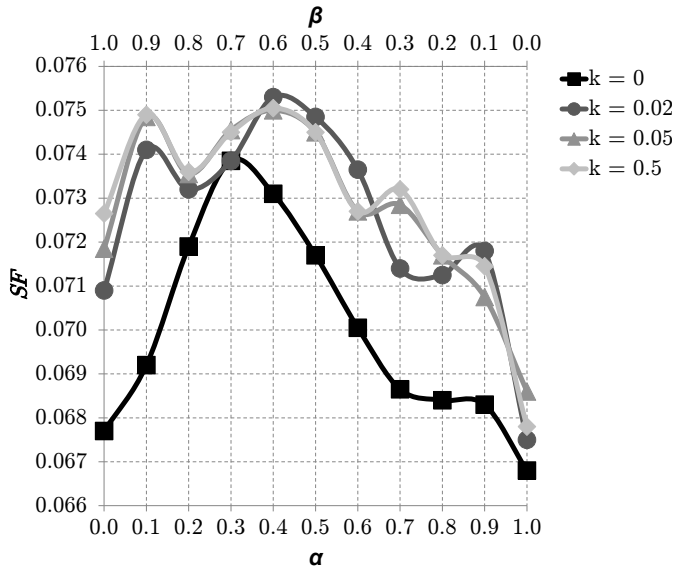

Figure 5: Harmonic mean of SP and SR for different α, β, and k values

terms we use the harmonic mean of the semantic precision and the semantic recall and define this as the semantic F-measure (SF). We use the STW Thesaurus for Economics and Business Economics as a reference ontology to compute this harmonic mean. The semantic precision (SP) is the proportion of concepts from the constructed taxonomy that is semantically present in the built taxonomy and reference ontology, while the semantic recall (SR) is the proportion of concepts from the reference ontology that appear in both ontologies. The results are depicted in Fig 5.

The aforementioned figure illustrates SF for different values of α, β, and k. The sum of α and β, which are weight values for respectively the domain pertinence and domain consensus, is equal to 1 (as domain pertinence and domain consensus are subunitary, their weighted sum is also subunitary if we use this constraint). As α increases, β thus decreases. A change from k = 0 to k = 0.02 corresponds to a noticeable increase in the harmonic mean. Further increasing k only slightly improves SF. Low values for either α or β also cause low harmonic mean values. We found that the harmonic mean is highest when α = 0.4, β = 0.6, and k = 0.02, therefore we decided to use these values for our implementation. As we aim to build a taxonomy that covers a wide spectrum of the domain of economy and management, we select 2,000 (out of 3,875) terms with the highest scores. These terms will be represented in the built taxonomy as a concept label or as an alternative label of a taxonomy concept.

4.2.2. Implementation Results

As was described before we use α = 0.4, β = 0.6, and k = 0.02 for our implementation. These three variables are used to compute a domain score for each term, and thus identify the most relevant domain terms.

A sample of terms selected using the described values is: {'market', 'firm', 'revenue', 'forecast', 'enterprise', 'policy', 'data', 'model'}. Most of these terms have either a low *DC* value or low *DP* value, which implies that only using one of these measures would greatly influence the outcomes. Terms like 'revenue' and 'forecast' would not be present in the taxonomy if the *DP* was not used, while 'model', 'policy' and 'data' would not be selected if the *DC* did not play a role in the selection procedure. This illustrates that using both *DC* and *DP* is an adequate way to select domain terms.

4.3. Word Sense Disambiguation

The next step is to apply word sense disambiguation (WSD). In this step the sense of the selected domain terms will be determined. The disambiguation of word senses is only useful for simple (non-compound) terms, as complex (compound) terms usually have only one meaning. In our implementation, the possible senses of a single term are acquired from a semantic lexicon, i.e., WordNet [14]. As WordNet contains thousands of senses covering ten thousands of terms across many domains, it is suited for use in our implementation. Moreover, WordNet is a relevant resource for many domains, as it is general and, additionally, there are language-specific versions that are either released or still under development (e.g., Cornetto for Dutch[5] and GermaNet for German[6]). The availability of semantic lexicons for various languages contributes to the extendability of our approach, as non-English corpora can easily be analyzed after connecting different lexicons. In WordNet terms are assigned senses by synsets. A synset is a collection of terms that share the same meaning. WordNet also contains the broader-narrower relations between the different synsets. As we apply Jiang and Conrath's similarity measure, which uses the lowest common subsumer of terms to determine a term's sense, the relations contained in WordNet can be used to derive the sense of terms [20]. First we discuss our choices regarding some pa-

[5] Available at `http://staff.science.uva.nl/~mdr/Research/Projects/Cornetto/`

[6] Available at `http://arbuckle.sfs.uni-tuebingen.de/GermaNet/`

rameters in the WSD process. Then, results of our WSD implementation are presented.

4.3.1. Choosing the Parameters

The parameter that may be tuned in both WSD applied on text corpora and WSD applied on taxonomies is the content size for the context lists used. Concerning the WSD procedure for text corpora we manually evaluated the amount of correctly disambiguated terms for two different context types: the document in which a term appears or the sentence in which a term appears. Our experiments show that the percentage of terms disambiguated correctly using the document as the context, i.e., approximately 68.4%, is larger than the percentage of terms disambiguated correctly using the sentence as the context, which totals to 67.9%.

One would say however that a sentence provides a more accurate context for a term than a document, as the sentence contains the closest surroundings of a term. In our case, the fact that using sentences does not perform better can be easily explained. We only use the abstract and title of each document. As these titles and abstracts usually are not much larger than a few sentences (and usually convey one meaning), using a sentence as the context of a term does not differ much from using a document as the context of a term. Documents help disambiguating words better, because titles and abstracts are rather homogeneous in meaning and thus provide for more context than sentences alone.

We also examined what influence a change in the concept neighbourhood has on the WSD procedure for existing taxonomies. We manually investigated what percentage of ambiguous concept labels was disambiguated correctly. First, we explored how many concept labels are disambiguated correctly when we just select the most common sense of a term (the baseline). Then, we tested the percentage of terms disambiguated correctly for different concept neighbourhood sizes.

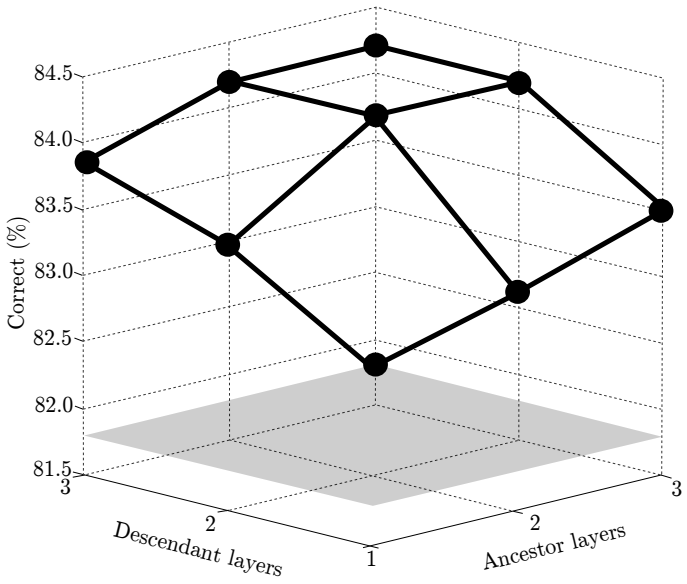

Figure 6: Percentage of concept labels that have been disambiguated correctly

Figure 6 illustrates that approximately 81.8% of the ambiguous concept labels are disambiguated correctly when selecting the most common sense of a term (as depicted by the gray plane). Applying the WSD approach for ontologies improves the percentage of correctly disambiguated labels (in comparison to using only the monosemous concept labels). We found that, as the concept neighbourhood becomes larger, the proportion of correctly disambiguated labels increases up until a certain maximum level is reached. Using a concept neighbourhood of two layers of ancestor concepts and two layers of descendant concepts yields a total of approximately 84.2% of the labels disambiguated correctly. This amount does not increase when further enlarging the concept neighbourhood. Therefore we decided to use a concept neighbourhood of two layers of ancestor concepts and two layers of descendant concepts to disambiguate taxonomy concepts.

4.3.2. Implementation results

For the previously selected domain terms we apply WSD using the text corpora of RePub and RePEc as the data sources to assess the context of each term. For each document in the domain corpus in which an ambiguous term appears, the term's

sense in that document is determined using the SSI method outlined in Sect. 3. The best overall sense of the term is retrieved by selecting the sense that appears in most domain corpus documents.

Table 1 illustrates the number of occurrences of the possible senses, represented by WordNet's synset notations, of the term 'payoff'. The sense notated as 'SID-12505332-N' was the sense assigned to 'payoff' in the majority of the documents. This sense is thus assigned to the concept labeled 'payoff' in the built taxonomy. The meaning of the sense is: 'the income arising from land or other property'. The meaning of the non-selected sense is: 'a recompense for worthy acts or retribution for wrongdoing', which indeed is outside the scope of our domain.

If other terms that were regarded as relevant domain terms in the previously applied term filtering step are assigned the same sense, they are added as an alternative label for the concept labeled 'payoff'. For example, the term 'return' was assigned the same sense. The computed term filtering score for this term was lower than the one for 'payoff'. The documents in which 'return' appeared are added to the documents in which 'payoff' occurred. Because of this merging of documents the meaning of both 'payoff' and 'return' is now well represented in one concept. This example illustrates the extra possibilities offered by WSD for the subsequent taxonomy construction steps. In total 142 of the 2,000 previously selected concept labels are merged.

4.4. Concept Hierarchy Creation

The last step is to store the previously constructed concepts in a concept hierarchy. To create the broader-narrower relations between the concepts we use the subsumption method [36]. On the basis of document co-occurrence the parent-child relationships of concepts are determined. In the previously applied WSD process, concepts with disambiguated concept labels that share the same meaning are merged. As the subsumption method uses the co-occurrence of concepts to establish the broader-narrower relation of a concept, this semantics-based merging should positively affect the resulting broader-narrower relations.

SynsetID	No. of occurrences
SID-12505332-N	160
SID-06847852-N	11
Total	171

Table 1: Number of occurrences of different synsets of the term 'payoff'.

First, we describe how we obtained the threshold value that plays a critical role in establishing the broader-narrower relation between concepts. Then, an excerpt of the obtained concept hierarchy is illustrated, as well as some of the concept hierarchy characteristics, such as the average depth and number of leaf nodes.

4.4.1. Choosing the parameters

To retrieve the broader-narrower relations of the concepts, a threshold value is used (as given in Formula 6). We have tested multiple threshold values to examine what impact different threshold values (from 0.1 to 1 with a step of 0.05) have on the constructed concept hierarchy. Figure 7 shows a diagram depicting the harmonic mean of the quality and the average depth of the built taxonomy, given certain threshold values. The quality is computed using the taxonomic F-measure (TF), which shows the degree of shared relations between the built taxonomy and a reference ontology [9].

Figure 7 illustrates the harmonic mean of the taxonomy quality and the average depth of the taxonomy. The higher the value for t is, the lower the average depth and the higher the quality of the built taxonomy. A higher threshold value corresponds to a more strict selection of parent concepts, and therefore also more accurate relationships. A trade-off thus has to be made between a higher average depth and a higher quality of the broader-narrower relations. By taking the harmonic mean be-

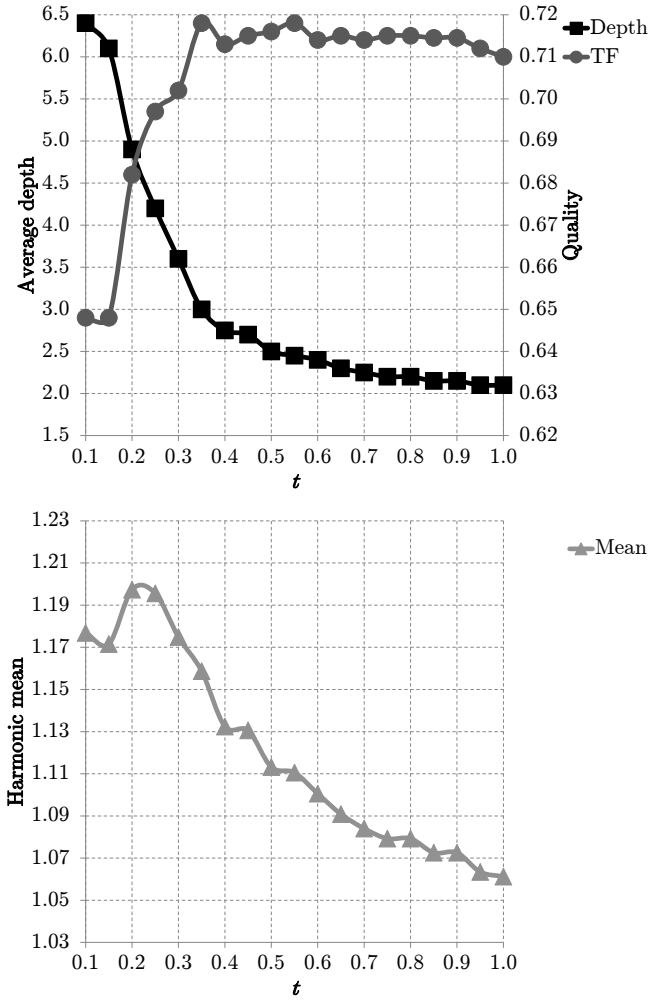

Figure 7: Quality and average depth (above) and the harmonic mean of the quality and average depth (below) of the created concept hierarchy given different threshold values

tween the quality and the average depth, we found that using t = 0.2 yields the best result.

4.4.2. Implementation results

For the concepts computed in the previous step, broader-narrower relations are determined using the subsumption method with a threshold value of t = 0.2. Using this threshold value results in a taxonomy with the characteristics displayed in Table 2.

With an average depth of approximately 4.84 and a maximum depth of 8 the constructed taxonomy is a taxonomy that is not too shallow. The average amount of child concepts of a concept is equal to approximately 1. As 1,436 of the 1,858 concepts are leaf concepts in the hierarchy and thus do not have child concepts, the average amount of children of the remaining 422 concepts is equal to approximately 4.40.

Characteristic	Value
Number of concepts	1,858
Number of leaves	1,436
Average depth	4.84
Maximum depth	8
Maximum number of children	77
Average number of children	1.00

Table 2: Characteristics of the built taxonomy

Figure 8 shows an excerpt of the constructed concept hierarchy. It depicts one of many examples of a collection of broader-narrower relations that are created by the subsumption method. The concepts are clearly representative for the domain of economics and management and also have appropriate broader-narrower relations.

5. Evaluation

In this section we evaluate the taxonomy for the domain of economics and management that is built using the settings proposed in the previously discussed implementation of our ATCT framework. First, the framework of our evaluation approach is described. Then, we report and discuss the evaluation results. We also briefly discuss a second taxonomy created for the domain of health and medicine.

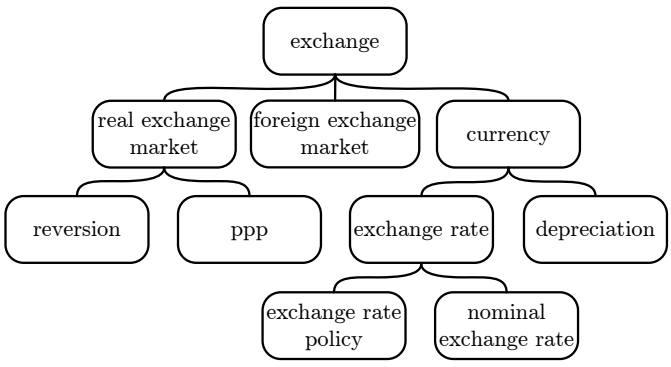

Figure 8: Excerpt of the built concept hierarchy

5.1. Evaluation framework

To evaluate the constructed taxonomy we apply a golden standard evaluation [9]. We compare our created taxonomy with a benchmark taxonomy using a number of measures. Measures from literature focus on two evaluation aspects. One aspect focuses on the precision and recall of the taxonomy concepts' lexical representations in the benchmark taxonomy. The other aspect focuses on the broader-narrower relations between concepts in the built taxonomy and the benchmark taxonomy.

Measures to evaluate the quality of the concepts present in a constructed taxonomy are the lexical precision and lexical recall. These measures compare the concept labels of the created taxonomy with the concept labels of a benchmark taxonomy. The lexical precision shows the proportion of concepts in the built taxonomy that is also present in the benchmark taxonomy. The lexical recall represents the proportion of benchmark taxonomy concepts that appear in the built taxonomy.

Figure 9 illustrates a small example of a core taxonomy and a reference taxonomy. The concepts that lexically appear in both taxonomies are: {'shipping', 'terminal', 'Rotterdam'}. Three out of seven concepts are thus lexically shared by the two ontologies. If one examines both taxonomies however, one can notice that semantically six of the seven concept labels are the same. For example, 'boat' has the same meaning as 'ship', and 'freight' is a synonym for 'cargo'. The only concept label from the built taxonomy that is not semantically present in the reference taxonomy is 'engine'.

Lexically comparing two taxonomies thus does not always give the right picture. Therefore, in order to cope with this problem we introduce the semantic precision and semantic recall as opposed to their lexical equivalents. Rather than using the lexical representations of concepts to compare taxonomies, we use the meaning of the concept labels. In this way, we prevent two situations that occur when applying lexical precision and lexical recall. In our computations, we prevent the situation that lexically different terms that share the same meaning are not considered as semantically the same (as illustrated by Fig. 9). We also prevent the situation that lexically the same, but semantically different terms are considered to be semantically the same. The definitions of the semantic precision (SP) and semantic recall (SR) are as follows:

$$SP(T_C, T_R) = \frac{|C_C \cap C_R|}{|C_C|}\,, \tag{9}$$

$$SR(T_C, T_R) = \frac{|C_C \cap C_R|}{|C_R|}\,, \tag{10}$$

where T_C and T_R are the core taxonomy and reference taxonomy, respectively, C_C represents the concepts of the core taxonomy, and C_R is the collection of concepts of the reference taxonomy. The concept intersection of these taxonomies, $C_C \cap C_R$, consists of the concepts that appear in both taxonomies.

By applying the WSD procedure to ontologies it may occur that some concept labels cannot be disambiguated as they are not recognized by the used semantic lexicon. This is due to the fact that a general semantic lexicon does not contain some

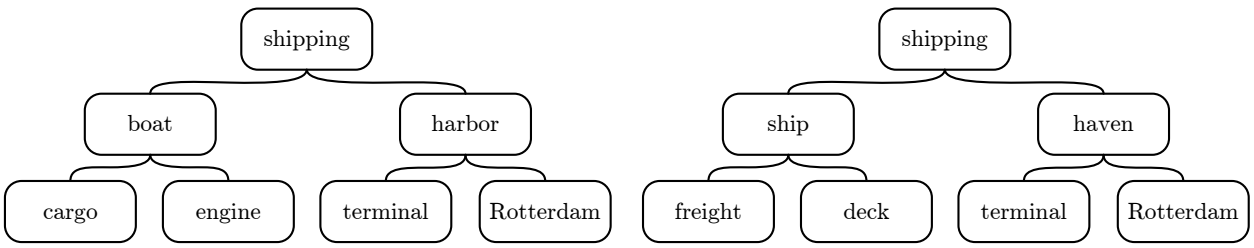

Figure 9: Small example core taxonomy (left) and reference taxonomy (right)

domain specific concepts. Therefore, senses cannot be found for some domain concepts, hence likely resulting in ambiguous concepts. This means that we are not able to compare such concepts semantically. However, we cannot ignore the concepts that could not be disambiguated and leave them out of the evaluation. Therefore, we apply a heuristic to determine whether an ambiguous concept label may semantically be present in the other taxonomy. If a concept label appears lexically the same in both the core taxonomy and reference taxonomy and they share, either lexically or semantically, a descendant concept or ancestor concept which is not the root node, they will likely have the same meaning. If this is the case, we add the concept to the taxonomy intersection.

For instance, assuming that in both the core taxonomy and reference taxonomy depicted in Fig. 9 there is a concept labeled 'terminal', in the right concept hierarchy the concept with label 'terminal' cannot be disambiguated. As the meaning of the term is not known, we cannot directly determine if the meaning appears in both taxonomies. We observe that the ancestor concepts of 'terminal' in the core ontology are 'harbor' and 'shipping'. We notice that the ancestor concepts of 'terminal' in the reference taxonomy are 'haven' and 'shipping'. The concepts labeled 'terminal' thus share an ancestor, a concept labeled 'shipping'. This means that they likely have the same meaning. The concept with label 'terminal' is added to the intersection of the core taxonomy and the reference taxonomy.

Next to evaluating the quality of concepts through semantic precision and recall, we additionally employ measures that aim at determining the quality of the concept relations in the built taxonomy [9]. To be able to compute the quality of the relations we use the common semantic cotopy (*csc*). The *csc* is the collection of a concept and its super- and sub-concepts that are shared by a core taxonomy and a reference taxonomy. It is defined as follows:

$$csc(c, T_C, T_R) = \{c_i | c_i \in C_C \cap C_R \wedge (c_i \leq_{C_C} c \vee c \leq_{C_C} c_i)\}, \quad (11)$$

where T_C and T_R are the core (built) taxonomy and reference taxonomy, respectively, C_C represents the concepts of the core taxonomy, C_R is the collection of concepts of the reference taxonomy, c is a concept, and $\leq C_C$ is the order induced by the broader-narrower relations in the T_C taxonomy. The *csc* of 'harbor' in the left taxonomy in Fig. 9 would consist of 'shipping', 'terminal', and 'Rotterdam'. These three concepts are either a super-concept or sub-concept of 'harbor' and appear in both taxonomies.

In our experiments, we use two measures, the global taxonomic precision (TP) and global taxonomic recall (TR), which employ the *csc* to compare the relations of the core taxonomy concepts and reference taxonomy concepts. To define the TP and TR, we must first define the local taxonomic precision (tp) and local taxonomic recall (tr). The definitions of tp and tr are as follows:

$$tp_{csc}(c, T_C, T_R) = \frac{|csc(c, T_C, T_R) \cap csc(c, T_R, T_C)|}{|csc(c, T_C, T_R)|} \quad (12)$$

$$tr_{csc}(c, T_C, T_R) = \frac{|csc(c, T_C, T_R) \cap csc(c, T_R, T_C)|}{|csc(c, T_R, T_C)|} \quad (13)$$

where T_C is the core taxonomy, T_R is the reference taxonomy, and c is a concept. Both tp and tr depict the quality of the relations of a single concept. This is measured by taking the intersection of the *csc* viewed from the core taxonomy's perspective and from the perspective of the reference taxonomy, respectively. In Fig. 9 the *csc* of 'boat' from the perspective of the core taxonomy is {'shipping', 'cargo'}. The *csc* of 'boat' ('ship') from the reference taxonomy's perspective is {'shipping', 'freight'}. The size of the intersection is then com-

pared to the size of the *csc* from the core taxonomy's perspective to determine *tp*, and compared to the size of the *csc* from the reference taxonomy's perspective to determine *tr*.

The global taxonomic precision (TP) and global taxonomic recall (TR) show the quality of relations of a taxonomy. They are defined as follows:

$$TP_{csc}(T_C, T_R) = \frac{1}{|C_C \cap C_R|} \cdot \sum_{c \in C_C \cap C_R} tp_{csc}(c, T_C, T_R) \ , \quad (14)$$

$$TR_{csc}(T_C, T_R) = \frac{1}{|C_C \cap C_R|} \cdot \sum_{c \in C_C \cap C_R} tr_{csc}(c, T_C, T_R) \ , \quad (15)$$

where T_C is the core taxonomy and T_R is the reference taxonomy. The TP is measured by first calculating the sum of tp values for each concept that is shared by the two taxonomies. This sum is then divided by the amount of shared concepts. For computing the TR the sum of tr values is used rather than the sum of tp values.

Next, we employ the taxonomic F-measure (TF), which describes the quality of the concept broader-narrower relations by taking the harmonic mean of TP and TR. The definition of TF is:

$$TF(T_C, T_R) = \frac{2 \cdot TP_{csc}(T_C, T_R) \cdot TR_{csc}(T_C, T_R)}{TP_{csc}(T_C, T_R) + TR_{csc}(T_C, T_R)} \ , \quad (16)$$

where T_C is the core taxonomy and T_R is the reference taxonomy. By applying the previously described measures we compare the built taxonomy with a benchmark taxonomy to determine the quality of the constructed taxonomy.

5.2. Evaluation results

We apply a golden standard evaluation using the measures described in the previous section and the STW Thesaurus for Economics and Business Economics as the benchmark taxonomy. This taxonomy contains a total of 3,875 concepts and their relations in the field of economics and business economics. As the STW Thesaurus also contains German labels, we manually translated the German terms to English (approximately 2% of the labels were adjusted). In this way the built taxonomy that contains solely English labels can be properly compared to the STW Thesaurus. Some characteristics of our built taxonomy and the STW taxonomy are depicted in Table 3. The latter table shows that the main difference between the built taxonomy and benchmark taxonomy is the size. The benchmark taxonomy is more than twice as large as the built taxonomy, and thus also contains a higher number of leaves and a higher average of child concepts.

A closer inspection of the taxonomy terms with respect to term ambiguity, leads to various insights. For the STW taxonomy, there are 3,875 concepts that have 3,874 unique term representations. From these concepts, 836 (21.6%) can be assigned a meaning. This low percentage (recall) can be explained by the fact that the STW ontology contains a high number of labels consisting of many words, such as 'computer-aided quality assurance', 'generally accepted auditing standards', and 'state participation in private enterprises'. Concepts with such labels are usually not recognized by a semantic lexicon and can therefore not be disambiguated. Of the 836 meaningful concepts, 217 (i.e., 26.0%) have 1 or more synonyms (2,138 in total), resulting in 6,013 unique terms. Also, 745 out of 3,874 terms are ambiguous. In other words, the number of ambiguous terms is 19.2%. These findings underline the importance of applying WSD procedures. In order to semantically evaluate the constructed taxonomy, we applied our WSD approach for taxonomies on the concepts of the benchmark STW ontology. This resulted in the correct disambiguation of approximately 84.2% of the terms that could be assigned a meaning.

Characteristic	Built tax.	Bench. tax.
No. Concepts	1,858	3,875
No. Leaves	1,436	2,541
Avg. Depth	4.84	6.04
Max. Depth	8	11
Max. Children	77	70
Avg. Children	1.00	1.85

Table 3: Characteristics of the built taxonomy and the benchmark taxonomy for economics and management

We also evaluated the performance of the WSD approach applied on text corpora, where we obtained a recall of approximately 62.9%, which is higher than the recall for the WSD approach applied on the concepts of a benchmark taxonomy. As the terms that are extracted from text corpora are usually shorter terms than the labels of the STW ontology, a larger number is recognized by the semantic lexicon, leading to a higher recall. The precision of the WSD approach applied on text corpora is equal to approximately 68.8%. This precision is lower than the precision of the WSD approach applied on the STW ontology. A larger set of terms is disambiguated by the WSD approach for text corpora than the WSD approach for taxonomies, likely causing lower precision. Also, the average number of senses the disambiguated terms have in the WSD approach for text corpora is also higher than the average number of senses of the disambiguated labels of the STW ontology. The average is approximately 4.37 meanings per disambiguated term extracted from the text corpora, while the average is approximately 3.67 for the disambiguated terms in the STW ontology, which is significantly lower. This higher average leads to a lower precision, because the probability that a correct term sense is selected becomes smaller as the number of possible senses increases.

Measure	WSD	No WSD
SP	0.1163	–
SR	0.0557	–
TP	0.8190	0.7843
TR	0.5837	0.5011
TF	0.6816	0.6115

Table 4: Evaluation results for the built taxonomy for economics and management

With the concepts of the STW ontology disambiguated we apply a semantic evaluation of the built taxonomy. Table 4 depicts the results of the measures presented in the previous section. The results show that approximately 11.63% of the senses found in the constructed taxonomy also appear in the benchmark taxonomy. A total of 216 senses is shared by the two taxonomies. From these 216 senses a total of 11 senses was acquired by comparing the ancestors and descendants of non-disambiguated concepts that have the same lexical representation. Examples of such concepts that were added to the semantic intersection are ‘aggregate demand’, ‘exchange rate policy’ and ‘real estate market’. By manually examining these 11 concepts we acknowledge that they all have been correctly mapped.

The quality of the broader-narrower relations, captured by TF, equals approximately 68.16%. This roughly means that 68.16% of the broader-narrower relations of the semantically shared concepts are the same for the built taxonomy and the benchmark taxonomy. We found that TF significantly increases when using the extra heuristic that examines for non-disambiguated concepts the ancestor and descendant concepts in the WSD approach applied for taxonomies. Without the heuristic the TF decreases to approximately 44.35%, indicating that the non-disambiguated concepts that are added to the intersection of the built taxonomy and the benchmark taxonomy have better broader-narrower relations in general than the disambiguated ones. The quality of the broader-narrower relations is measured by examining the proportion of shared ancestor or descendant concepts: the higher this proportion, the higher the quality. As a non-disambiguated concept is added to the intersection if the associated concept in the built taxonomy and the associated concept in the benchmark taxonomy share an ances-

tor or descendant concept, this usually means that such added concepts contribute to a higher quality of broader-narrower relations.

Additionally, we evaluate the added value of the WSD procedure in the generated taxonomy. Therefore, we also measure the performance of the taxonomy construction framework without WSD. As we hence do not take into semantics, we are unable to evaluate the framework on semantic precision and recall, i.e., SP and SR, respectively, yet we can compute the taxonomic precision, recall, and F-measure (TP, TR, and TF).

The results in Table 4 clearly demonstrate the importance of disambiguating terms in a generated taxonomy. If the WSD procedure is omitted when generating a taxonomy from a text corpus, performance in terms of taxonomic precision, recall, and F-measure decreases by 4.2%, 14.2%, and 10.3%, respectively.

When analyzing the quality of disambiguated terms, we observe the following. First, 79.3% of the terms have the same meaning in both the generated and the STW reference taxonomy. These terms are mainly well-known and well-defined terms used specifically in economics and management, such as 'consumer price index', 'budget deficit', 'cash flow', 'tax system', and 'nash equilibrium', but also broader terms such as 'money' and 'economics'.

The other 20.7% of the terms that occur in both taxonomies, have different associated meanings. Generally, these terms are ambiguous terms that have an economic meaning, but also other interpretations that have only little to do with the domain of economics and management per se. For example, 'capital' is defined in the generated taxonomy as a seat of government, while in the STW taxonomy, it is defined as wealth in the form of money or property owned by a person or business, or human resources of economic value. The term 'inflation' is defined as a lack of elegance as a consequence of being pompous and puffed up with vanity, instead of a general and progressive increase in prices. Also, 'capitalization' is defined as writing in capital letters instead of an estimation of the value of a business.

Such differences are caused by the fact that the context used by the WSD method is different: the taxonomy-based WSD makes use of a context that is based on the taxonomy itself, forcing the disambiguation process to look for economic or managerial senses. The corpus-based WSD, on the other hand, uses text as a context, which not necessarily binds the disambiguation to the economics and management domain. However, in most cases the proper definition is found, due to the fact that the corpus is within the considered domain.

Differences in context can also result in milder cases of sense mismatch. There are a few terms that have a slightly different (economic) meaning. For instance, in the reference taxonomy, 'wealth' is defined as the state of being rich and affluent, whereas in the generated taxonomy, the term is described as property that has economic utility, i.e., a monetary or exchange value. Also, in the STW taxonomy, 'information' is assigned the meaning of a message received and understood, and in the generated taxonomy, it is defined as knowledge acquired through study, experience, or instruction. Hence in such cases, both definitions are within the correct domain, yet their interpretation differs slightly.

In order to investigate the performance of the ATCT framework when creating a taxonomy for a domain other than economics and management, we also constructed a taxonomy for medicine and health. The concept labels for this taxonomy are extracted from the text corpus of RePub. A total of 10,000 documents from RePub is used to extract the 1,000 most relevant domain terms to appear as concept labels in the built taxonomy. The taxonomy is created using the same parameters as the taxonomy built for economics and management. The benchmark taxonomy we used is the MeSH taxonomy, which

Characteristic	Built tax.	Bench. tax.
No. Concepts	959	15,337
No. Leaves	709	10,345
Avg. Depth	4.96	6.98
Max. Depth	9	13
Max. Children	39	162
Avg. Children	1.00	1.27

Table 5: Characteristics of the built taxonomy and the benchmark taxonomy for the domain of medicine and health

contains thousands of subject headings for the medical domain. The characteristics of the built taxonomy and the benchmark taxonomy are depicted in Table 5.

For the MeSH taxonomy we observe a different pattern with respect to term ambiguity when compared to the STW taxonomy. The taxonomy counts a total number of 15,336 concepts, represented by 15,285 unique terms. From these concepts, 4,316 (28.1%) can be assigned a meaning. Of these 4,316 concepts, there are no words that have 1 or more synonyms. However, 4,265 out of 15,285 terms (i.e., 27.9%) are ambiguous. Hence, similar to the STW taxonomy, the MeSH taxonomy also underlines the importance of using WSD procedures, and therefore we apply an additional disambiguation procedure to the MeSH taxonomy.

The benchmark taxonomy is notably bigger than our constructed taxonomy. The semantic recall value will therefore be low, as the number of shared concepts between the built taxonomy and benchmark taxonomy is divided by a large number. Table 6 shows that the taxonomy created for medicine and health is comparable to the one built for economics and management. The SP is higher, while the SR is lower, as the MeSH taxonomy is much larger than the STW taxonomy used before for the domain of economics and management. The overall quality of the broader-narrower relations, denoted by TF, is similar to the quality of the taxonomy constructed for economics and management. Also for the medicine and health domain, using WSD appears to beneficial for the results in terms of taxonomic recall and F-measure (with improvements of 6.6% and 2.9%, respectively), and yields approximately the same taxonomic precision as compared to taxonomy extraction from text without WSD. Although the differences caused by WSD are smaller than for the STW taxonomy (caused by a lack of improvement in precision which in turn is caused by the larger amount of ambiguous terms and the increased number of meanings per disambiguated term), overall, the results still underline the intuition that the ATCT framework can be successfully applied to other domains than economics and management.

Measure	WSD	No WSD
SP	0.1846	–
SR	0.0115	–
TP	0.6173	0.6182
TR	0.6900	0.6472
TF	0.6516	0.6324

Table 6: Evaluation results for the built taxonomy for medicine and health

Furthermore, when analyzing the disambiguation quality for medicine and health, similar patterns can be observed as for the domain of economics and management. For 85.1% of the terms, the meaning is identical in both in the MeSH and the generated taxonomy. This generally holds for techniques and procedures like 'biopsy', 'transplantation', 'echocardiography', and 'angiography', as well as for terms with many subcategories and related terms, such as 'virus' and 'enzyme'. Also, broad terms for complex systems, e.g., 'immune system' and 'central nervous system', and most general, body-related terms such as 'skin', 'muscle', 'intestine', and 'bone marrow' are disambiguated to the same senses.

The remaining 14.9% of the terms have a different meaning in both taxonomies. In most cases, these terms have a medical meaning in the reference taxonomy, but are assigned distinct, non-medical, interpretations in the generated taxonomy. For instance, 'axis' is defined as the second cervical vertebra that serves as a pivot for turning the head, but in the generated

taxonomy, it is a group of countries in special alliance. Next 'heart' is defined as the hollow muscular organ located behind the sternum and between the lungs, yet in the generated taxonomy it refers to the area that is approximately central within some larger region. A 'cell' is generally seen as the basic structural and functional unit of all organisms, but in the generated taxonomy it is disambiguated as a room where a prisoner is kept. Additionally, in the MeSH taxonomy, 'blood' is defined as the fluid (red in vertebrates) that is pumped by the heart, but in the generated taxonomy it is defined as the descendants of one individual. A last example is 'arm', which is a human limb in the domain of health and medicine, yet in the generated taxonomy, it is disambiguated as any instrument or instrumentality used in fighting or hunting, i.e., a weapon.

Some of these terms only have a slightly different meaning in both taxonomies. These small differences can be caused by different plausible meanings within the medical domain, e.g., 'medicine' refers to the branches of medical science that deal with nonsurgical techniques in the MeSH taxonomy, but in the generated taxonomy it is disambiguated as something that treats, prevents, or alleviates the symptoms of disease. Other terms are disambiguated in the benchmark taxonomy with a sense that pertains to medicine and health, yet in the generated taxonomy, they have a very similar (but non-identical) meaning that is defined in a more general way, and which does not necessarily connect to the considered domain. The terms 'risk', 'prognosis', and 'alcohol' are typical examples. In the MeSH taxonomy, the first term is defined as the probability of becoming infected given that exposure to an infectious agent has occurred, while in our generated taxonomy, it is a venture undertaken without regard to possible loss or injury. Hence, the meaning is similar, but lies within a different context. This also holds for the second term, 'prognosis', which is defined in the medical domain as a prediction of the course of a disease, whereas in the generated taxonomy it is disambiguated as a prediction about how something (as the weather) will develop. Last, in the MeSH taxonomy, 'alcohol' is defined as any of a series of volatile hydroxyl compounds that are made from hydrocarbons by distillation, but in the generated taxonomy it indicates a liquor or brew containing alcohol as the active agent.

6. Concluding Remarks

We presented an approach for the automatic construction of a taxonomy from a text corpus, which comprises four steps. First, we extract terms from a corpus using a part-of-speech tagger [22]. Subsequently, from these extracted terms the ones that are most relevant for a specific domain are selected using a filtering approach. Terms are selected on the basis of domain consensus, domain pertinence, and structural relevance. In order to build a concept hierarchy we applied the subsumption method. This method uses the co-occurrence of concepts in the used text corpora to establish the broader-narrower relation between concepts. Third, the selected terms are disambiguated by means of a word sense disambiguation technique and concepts are generated. In the final step, the broader-narrower relations between concepts are determined using a subsumption technique that makes use of concept co-occurrences in text.

We evaluated the constructed taxonomy by comparing it with a reference (benchmark) taxonomy using the golden standard evaluation approach. For this, we have also described new measures, i.e., semantic precision and semantic recall, which measure the quality of the concept representations of the created taxonomy. Moreover, we used existing measures such as the taxonomic precision and taxonomic recall to retrieve the quality of the broader-narrower relations in the built taxonomy. Instead of employing the lexical representations of taxonomy concepts we use the acquired semantic representations to retrieve the quality of the broader-narrower relations.

We constructed a taxonomy for the domain of economics and management. By semantically evaluating the built taxonomy we found that the senses present in the taxonomy had a semantic precision (SP) of approximately 11.63% and a semantic recall (SR) of approximately 5.57%. Additionally, we also tested the framework on the medicine and health domain, yielding similar results. The SP and SR are dependent on the size of the built taxonomy and the benchmark taxonomy, respectively. As the STW ontology, which was the benchmark ontology we used for evaluation, is more than twice as large as the built taxonomy the SR is lower than the SP.

The lower SP and SR values are explained by the content of the STW ontology. The STW ontology contains many concepts with labels used for broad categories, which are not easily extracted from text in an automatic manner. Examples of such terms are 'environmental and resource economics', 'public finance and finance research', and 'management science and operations research'. The majority of the concepts in the taxonomy intersection (computed and benchmark) have shorter labels like 'real estate', 'interest rate', and 'marginal cost', as these labels are recognized more often by a semantic lexicon.

The quality of the broader-narrower relations between the concepts of the built taxonomy is given by the taxonomic F-measure (TF) which is 68.16%. A majority of the broader-narrower relations of the concept in the intersection of the built taxonomy and the benchmark taxonomy is thus shared by the two taxonomies. This shows that our approach works well in capturing the broader-narrower relation between concepts.

We have additionally evaluated the effect of the disambiguation procedure in the generated ontology. From our experiments, we conclude that this additional disambiguation step results in higher taxonomic precision, recall, and F-measure values. Last, all results obtained on the domain of economics and management have additionally been confirmed on another domain, i.e., the domain of medicine and health, which was benchmarked using the MeSH ontology.

In our endeavors, we constructed a taxonomy using a term filtering method to select the most relevant terms in the domain of economics and management. These terms are processed into labels of the concepts that ultimately form the taxonomy using the subsumption method. To take into account that terms might not be selected as they are not recognized by the used semantic lexicon, we applied a heuristic which adds lexically equivalent concepts to the semantic intersection of two taxonomies when the concepts have a common ancestor or descendant.

As future work we would like to improve the proposed algorithm as follows. One could enhance the concept mappings between the built taxonomy and the reference taxonomy by taking into account the semantic distances (e.g., path lengths) between concepts in a graph that combines the built taxonomy with the benchmark taxonomy. The smaller the distance between concepts, the higher the probability that the concepts are semantically the same. Also, we would like to investigate the use the proposed semantic approach on other methods than the subsumption method, e.g., hierarchical clustering, formal concept analysis, etc. This would allow us to compare the semantics-based implementation of several corpus-based taxonomy construction methods.

Acknowledgment

The authors are partially supported by the NWO Physical Sciences Free Competition project 612.001.009: Financial Events Recognition in News for Algorithmic Trading (FERNAT) and the Dutch national program COMMIT.

References

[1] Barrière, C., Agbago, A., 2006. TerminoWeb: A Software Environment for Term Study in Rich Contexts. In: 3rd International Conference on

Terminology, Standardisation and Technology Transfer (TSTT 2006). International Information Centre for Terminology, pp. 103–113.

[2] Berners-Lee, T., Hendler, J., Lassila, O., 2001. The Semantic Web. Scientific American 284 (5), 34–43.

[3] Borsje, J., Levering, L., Frasincar, F., 2008. Hermes: A Semantic Web-Based News Decision Support System. In: 23rd Annual ACM Symposium on Applied Computing. ACM, pp. 2415–2420.

[4] Budanitsky, A., Hirst, G., 2001. Semantic Distance in WordNet: An Experimental, Application-Oriented Evaluation of Five Measures. In: Workshop on WordNet and Other Lexical Resources at 2nd Meeting of the North American Chapter of the Assocation for Computational Linguistics (NAACL 2001). From: `ftp://ftp.cs.toronto.edu/pub/gh/Budanitsky+Hirst-2001.pdf`.

[5] Cimiano, P., Hotho, A., Staab, S., 2005. Learning Concept Hierarchies from Text Corpora Using Formal Concept Analysis. Journal of Artificial Intelligence research 24 (1), 305–339.

[6] Cimiano, P., Staab, S., Tane, J., 2003. Automatic Acquisition of Taxonomies from Text: FCA meets NLP. In: ECML/PKDD Workshop on Adaptive Text Extraction and Mining (ATEM 2003) at 14th European Conference on Machine Learning (ECML 2003) and the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2003). pp. 10–17, from: `http://staffwww.dcs.shef.ac.uk/people/F.Ciravegna/ATEM03/ATEM03-Proceedings.pdf`.

[7] de Knijff, J., Frasincar, F., Hogenboom, F., 2013. Domain Taxonomy Learning from Text: The Subsumption Method versus Hierarchical Clustering. Data & Knowledge Engineering 83 (1), 54–69.

[8] Decadt, B., Hoste, V., Daelemans, W., van den Bosch, A., 2004. GAMBL, Genetic Algorithm Optimization of Memory-Based WSD. In: 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (Senseval-3) at 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004). Association for Computational Linguistics, pp. 108–112.

[9] Dellschaft, K., Staab, S., 2006. On How to Perform a Gold Standard Based Evaluation of Ontology Learning. In: Cruz, I. F., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L. (Eds.), 5th International Semantic Web Conference (ISWC 2006). Vol. 4273 of Lecture Notes in Computer Science. Springer, pp. 228–241.

[10] Drouin, P., 2003. Term Extraction Using Non-Technical Corpora as a Point of Leverage. Terminology 9 (1), 99–115.

[11] Drouin, P., 2004. Detection of Domain Specific Terminology Using Corpora Comparison. In: Lino, M. T., Xavier, M. F., Ferreira, F., Costa, R., Silva, R. (Eds.), 4th International Conference on Language Resources and Evaluation (LREC 2004). European Language Resources Association, pp. 79–82.

[12] Du, H., Zhou, L., 2012. Improving Financial Data Quality Using Ontologies. Decision Support Systems 54 (1), 76–86.

[13] Du, T. C., Li, F., King, I., 2009. Managing Knowledge on the Web Extracting Ontology from HTML Web. Decision Support Systems 47 (4), 319–331.

[14] Fellbaum, C., 1998. WordNet: An Electronic Lexical Database. Computational Linguistics 25 (2), 292–296.

[15] Frantzi, K. T., Ananiadou, S., ichi Tsujii, J., 1998. The C-value/NC-value Method of Automatic Recognition for Multi-Word Terms. In: 2nd European Conference on Research and Advanced Technology for Digital Libraries (ECDL 1998). Springer, pp. 585–604.

[16] Gligorov, R., ten Kate, W., Aleksovski, Z., van Harmelen, F., 2007. Using Google Distance to Weight Approximate Ontology Matches. In: Williamson, C. L., Zurko, M. E., Patel-Schneider, P. F., Shenoy, P. J. (Eds.), 16th International World Wide Web Conference (WWW 2007). ACM, pp. 767–776.

[17] Hearst, M. A., 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In: 14th Conference on Computational Linguistics (COLING 1992). Vol. 2. pp. 539–545.

[18] Hotho, A., Staab, S., Stumme, G., 2003. WordNet improves Text Document Clustering. In: Semantic Web Workshop at 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2003). ACM, pp. 541–544.

[19] IDC, 2010. The Digital Universe Decade - Are You Ready? from: `http://www.emc.com/collateral/demos/microsites/idc-digital-universe/iview.htm`.

[20] Jiang, J. J., Conrath, D. W., 1997. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In: 10th International Conference on Research in Computational Linguistics (ROCLING 1997). pp. 19–33.

[21] Kaza, S., Chen, H., 2008. Evaluating Ontology Mapping Techniques: An Experiment in Public Safety Information Sharing. Decision Support Systems 45 (4), 714–728.

[22] Klein, D., Manning, C. D., 2002. Fast Exact Inference with a Factored Model for Natural Language Parsing. In: Becker, S., Thrun, S., Obermayer, K. (Eds.), Neural Information Processing Systems (NIPS 2002). Vol. 15 of Advances in Neural Information Processing Systems. MIT Press, pp. 3–10.

[23] Lance, G. N., Williams, W. T., 1967. A General Theory of Classificatory Sorting Strategies: 1. Hierarchical Systems. The Computer Journal 9 (4), 373–380.

[24] L'Homme, M.-C., 2004. A Lexico-semantic Approach to the Structuring

of Terminology. In: Ananiadou, S., Zweigenbaum, P. (Eds.), 3rd International Workshop on Computational Terminology (CompuTerm 2004) at 20th International Conference on Computational Linguistics (COLING 2004). Association for Computational Linguistics, pp. 7–13.

[25] Li, Y., Zhou, X., Bruza, P., Xu, Y., Lau, R. Y., 2012. A Two-Stage Decision Model for Information Filtering. Decision Support Systems 52 (3), 706–716.

[26] Maedche, A., Volz, R., 2001. The Ontology Extraction and Maintenance Framework Text-To-Onto. In: Workshop on Integrating Data Mining and Knowledge Management (DM-KM 2001), at the 2001 IEEE International Conference on Data Mining (ICDM 2001). From: `http://users.csc.calpoly.edu/~fkurfess/Events/DM-KM-01/Volz.pdf`.

[27] Makrehchi, M., Kamel, M. S., 2007. Automatic Taxonomy Extraction Using Google and Term Dependency. In: 6th IEEE/WIC/ACM International Conference on Web Intelligence (WI 2007). IEEE Computer Society, pp. 321–325.

[28] Mihalcea, R., Faruque, E., 2004. SenseLearner: Minimally Supervised Word Sense Disambiguation for All Words in Open Text. In: 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (Senseval-3) at 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004). Association for Computational Linguistics, pp. 155–158.

[29] Miles, A., Bechhofer, S., 2009. SKOS Simple Knowledge Organization System Reference – W3C Recommendation 18 August 2009. From: `http://www.w3.org/TR/2009/REC-skos-reference-20090818/`.

[30] Navigli, R., Velardi, P., 2005. Structural Semantic Interconnections: A Knowledge-Based Approach to Word Sense Disambiguation. IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (7), 1075–1086.

[31] Neshati, M., Alijamaat, A., Abolhassani, H., Rahimi, A., Hosseini, M., 2007. Taxonomy Learning Using Compound Similarity Measure. In: 6th ACM International Conference on Web Intelligence (WI 2007). IEEE Computer Society.

[32] Pekar, V., Staab, S., 2002. Taxonomy Learning - Factoring the Structure of a Taxonomy into a Semantic Classification Decision. In: 19th International Conference on Computational Linguistics (COLING 2002). From: `http://acl.ldc.upenn.edu/C/C02/C02-1090.pdf`.

[33] Resnik, P., 1995. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In: 14th Int. Joint Conf. on Artificial Intelligence (IJCAI 1995). Morgan Kaufmann, pp. 448–453.

[34] Salton, G., Buckley, C., 1988. Term-Weighting Approaches in Automatic Text Retrieval. Information Processing and Management 24 (5), 513–523.

[35] Sánchez, D., Moreno, A., 2008. Pattern-Based Automatic Taxonomy Learning from the Web. AI Communications 21 (1), 27–48.

[36] Sanderson, M., Croft, B., 1999. Deriving Concept Hierarchies from Text. In: 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1999). ACM, pp. 206–213.

[37] Schmitz, P., 2006. Inducing Ontology from Flickr Tags. In: Workshop on Collaborative Tagging at 15th International World Wide Web Conference (WWW 2006). IW3C2, pp. 206–209.

[38] Sclano, F., Velardi, P., 2007. TermExtractor: a Web Application to Learn the Shared Terminology of Emergent Web Communities. In: 7th Conference on Terminology and Artificial Intelligence (TIA 2007). From: `http://wwwusers.di.uniroma1.it/~velardi/TIA-termextractor.pdf`.

[39] Shambour, Q., Lu, J., 2012. A Trust-Semantic Fusion-Based Recommendation Approach for E-Business Applications. Decision Support Systems 54 (1), 768–780.

[40] Tam, K. Y., 1993. Applying Conceptual Clustering to Knowledge-Bases Construction. Decision Support Systems 10 (2), 173–198.

[41] The Apache Software Foundation, 2013. Apache Jena – A Free and Open Source Java Framework for Building Semantic Web and Linked Data Applications. From: `http://jena.apache.org/`.

[42] Toutanova, K., Manning, C. D., 2000. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. In: Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC 2000). ACM, pp. 63–70.

[43] Vandic, D., van Dam, J.-W., Frasincar, F., 2012. Faceted Product Search Powered by the Semantic Web. Decision Support Systems 53 (3), 425–437.

[44] Weber, N., Buitelaar, P., 2006. Web-based Ontology Learning with ISOLDE. In: Workshop on Web Content Mining with Human Language Technologies collocated with the 5th International Semantic Web Conference (ISWC 2006). From: `http://www.dfki.de/dfkibib/publications/docs/ISWC06.WebContentMining.pdf`.

[45] Wu, S.-T., Li, Y., Xu, Y., Pham, B., Chen, Y.-P. P., 2004. Automatic Pattern-Taxonomy Extraction for Web Mining. In: 3rd IEEE/WIC/ACM International Conference on Web Intelligence (WI 2004). IEEE Computer Society, pp. 242–248.

[46] Yuret, D., 2004. Some Experiments with a Naive Bayes WSD System. In: 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (Senseval-3) at 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004). Association for Computational Linguistics, pp. 265–268.