Word Sense Disambiguation for Automatic Taxonomy Construction from Text-Based Web Corpora

Jeroen de Knijff, Kevin Meijer,
Flavius Frasincar, and Frederik Hogenboom

Erasmus University Rotterdam
PO Box 1738, NL-3000 DR
Rotterdam, The Netherlands

{312470jk, 312177km}@student.eur.nl
{frasincar, fhogenboom}@ese.eur.nl

Abstract. In this paper, we propose the Automatic Taxonomy Construction from Text (ATCT) framework for building taxonomies from text-based Web corpora. The framework is composed of multiple processing steps. Firstly, domain terms are extracted using a filtering method. Subsequently, Word Sense Disambiguation (WSD) is optionally applied in order to determine the senses of these terms. Then, by means of a subsumption technique, the resulting concepts are arranged in a hierarchy. We construct taxonomies with and without WSD and we investigate the effect of WSD on the quality of concept type-of relations using an evaluation framework that uses a golden taxonomy. We find that WSD improves the quality of the built taxonomy in terms of the taxonomic F-Measure.

1 Introduction

Nowadays, an ever increasing amount of documents is digitally stored and readily available on the Web. A common issue with managing these documents is that many of these are organized in an unstructured manner. Organizing these documents in a structured way by creating a taxonomy representation can be beneficial, as it enhances the overview of available documents. A taxonomy is defined as a specific form of an ontology, which is a formal, explicit specification of a shared conceptualization [6] that provides users with insight into the type relations between domain concepts.

Currently, many taxonomies are manually created. While manually constructing taxonomies is usually more accurate because of the involvement of domain experts, automatically generating taxonomies is less costly and time consuming. Taxonomy construction on the Web is particularly of interest, as it enables inter-operability between Web sites, tools, etc. due to the knowledge aggregation into shared taxonomies. The Web fosters an incredible large amount of

information and knowledge that up until know is mostly not linked and even remains virtually invisible because of the lack of structure. Within the last decade, there has been a trend of connecting related data that has not been previously linked before. The well-know Linked Open Data cloud diagram[1], which aims to show the inter-connectivity of today's Web sites and their knowledge bases, grows by the year. However, this cloud could grow at an even higher rate and become increasingly more complex when widely applying automatic taxonomy construction. Hence, due to the need for structured information, as well as the complexity and considerable effort for manually creating taxonomies, automatic taxonomy construction is an interesting field to explore.

Although there is a substantial body of literature on automatic taxonomy construction, the amount of literature focussing on applying Word Sense Disambiguation (WSD) is limited, even though WSD is proven to be able to improve the results of clustering [8]. Hence, we evaluate the influence of a sophisticated WSD algorithm on an existing concept subsumption method. We propose a framework for Automatic Taxonomy Construction from Text (ATCT), which we use for the evaluation of the added value of WSD in taxonomy construction. The research presented in this paper is based on the analysis of 25,000 abstracts extracted from two online paper repositories, i.e., RePub (`http://repub.eur.nl/`) and RePEc (`http://repec.org/`), which are divided into a set of domain abstracts, as well as a contrastive set. We investigate the impact of WSD for the domain of economics and management, as well as the medical domain.

The main contribution of this paper is five-fold. Firstly, we alter an evaluation measure, so that the semantic concept representation is taken into account, allowing us to measure the influence of WSD properly. Secondly, we analyze the influence of WSD on taxonomy construction. Thirdly, we investigate the optimal parameters for the methods that comprise the ATCT framework. Fourthly, we modify the subsumption algorithm introduced in [14] to take the position of the ancestors with respect to the current node into account. Finally, we present an implementation of this approach for the domain of economics and management, as well as the medical domain. To our knowledge, taxonomy construction has been applied to other domains, e.g., finance and tourism [3], but not to these domains before.

The organization of our paper is as follows. Section 2 includes a review of related work in the area of taxonomy construction. Section 3 describes in detail the ATCT framework that generates the taxonomies. Subsequently, Sect. 4 discusses the implementation of our framework. Finally, the framework and its implementation are evaluated in Sect. 5 and conclusions are drawn in Sect. 6.

2 Related work

Extracting terms from text corpora can be done by means of linguistic methods, statistical methods, and hybrid methods. Linguistic methods generally use Nat-

[1] The diagram is based on data from the W3C SWEO Linking Open Data Community Project and is regularly updated at `http://richard.cyganiak.de/2007/10/lod/`.

ural Language Processing (NLP) techniques, such as Part-Of-Speech (POS) tagging, morphological analysis, and lexico-syntactic patterns [7]. Linguistic methods are very well capable of defining the function of a word in a sentence, but they do not take into consideration the importance of a term. Statistical methods, such as the Term Frequency – Inverse Document Frequency (TF-IDF), only use statistical techniques to extract terms from text. A problem with statistical methods is that they can filter out less frequently occurring important terms, as linguistic functions are ignored. To cope with this, hybrid methods are being developed, which use for instance stopping lists, chi-square measures, and term lengths [13, 15, 16].

Several similarity measures exist for Word Sense Disambiguation (WSD), such as the fast (yet possibly inaccurate) Resnik's similarity [2], which calculates similarity values between terms by analyzing the degree of information they share. Jiang and Conrath's similarity measure [9] is more accurate, as it takes into account the information content of the lowest common subsumer as well as of the terms themselves.

Multiple techniques can be applied to construct hierarchical relations between terms. For example, one could employ (hierarchical) clustering techniques using various similarity measures, such as window-methods and co-occurrences. Labeling clusters can be done by selecting the centroid of the cluster or the lowest hypernym of terms in a cluster as a label [3]. Another approach to taxonomy construction is the usage of a classification method. It is possible to combine a domain corpus, a general corpus, and a named-entity tagger to extract and arrange terms in a taxonomy using additional sources that provide more information about these terms in a tree-ascending or tree-descending way [16]. Classification methods can provide accurate results, but they require a large training set, which makes this method difficult to use when a large training set is not available. A final approach to hierarchical relation creation is the usage of lexical-syntactic patterns, by creating taxonomic relations through pattern matching [7]. In the subsumption method, based on co-occurrences, a term subsumes another term if they co-occur frequently. This method is simple and it does not have any labeling issues, but it is weak in arranging terms that do not occur frequently in documents [14]. Formal concept analysis, a conceptual clustering technique that groups attributes and their attributes, can be applied in text methods by linking terms with verbs [3].

There are many approaches to taxonomy evaluation [14, 3, 4]. Often, taxonomies are evaluated by comparing them to a golden taxonomy, which is made by domain experts. Common techniques for comparing a taxonomy with a golden taxonomy are the lexical recall and lexical precision, which show how well the terms reflect the target domain. The taxonomic precision and taxonomic recall analyze common concepts in both taxonomies. By using the semantic cotopy it is possible to only consider the super- and sub-concepts of a node. The common semantic cotopy is similar, but it only considers nodes that both taxonomies have in common. The F-Measure is a harmonic mean of the taxonomic recall and taxonomic precision that indicates how similar the taxonomies are in terms

of the hierarchical relations. It is also possible to express the quality of the taxonomy in terms of a harmonic mean of the F-Measure and the lexical recall or precision. This measure takes both the quality of the found terms and the relevance of these terms into account. When a golden taxonomy is not available, one could ask several experts to rate the taxonomic relations in the generated taxonomy, and average their judgments afterwards.

3 ATCT Framework

Figure 1 gives an overview of the architecture of our four-step ATCT framework for automatically constructing a domain-specific taxonomy. First, the terms are extracted from the documents, which subsequently get processed in our term filtering step. Here, terms are filtered on lexical cohesion and domain pertinence, after which the most relevant terms are selected on the basis of a score, which is determined by domain pertinence, domain consensus, and structural relevance. The selected terms are processed into concepts as concept labels. The next optional step is *Word Sense Disambiguation* (WSD), which is used to derive the sense of a concept term and to find synonyms of the disambiguated term. Lastly, a *concept hierarchy* is created by constructing the type relations between concepts. The resulting hierarchy is represented in SKOS [1], a commonly used domain taxonomy representation format.

3.1 Term Extraction

The first step within the ATCT framework is to extract the terms from a set of documents. We perform this task by tagging all the words that appear in these documents and extracting those terms that are tagged as a noun. The choice for nouns is motivated by the fact that concepts are usually represented by nouns rather than other types of words, which is also the case for the golden taxonomies we use in our evaluation.

3.2 Term Filtering

Extracted terms are filtered on multiple criteria, i.e., their domain pertinence and their lexical cohesion value. The most relevant terms are selected by calculating a

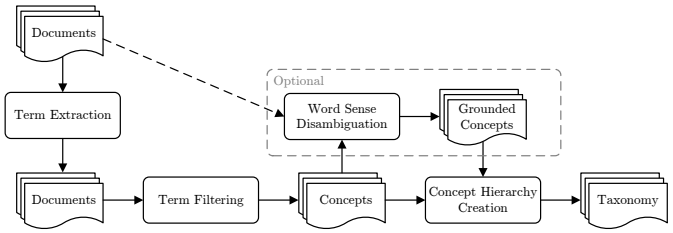

Fig. 1. Overview of the ATCT framework.

score, which is based on the domain pertinence, domain consensus and structural relevance of the term [15]. The four used filters are as follows:

1. **Domain pertinence** measures whether a term is relevant for the target domain. We define domain pertinence as

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

   where D_i represents the domain corpus and D_j denotes a contrastive corpus. The more frequently a term appears in the domain corpus and the less frequently the term appears in the contrastive corpus, the higher the value of the domain pertinence. The 30% terms with the lowest values are filtered out.
2. **Lexical cohesion**, which is used to measure the cohesion among words in a term, is only used for compound nouns. This filter is defined as

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

   where n is the amount of words in compound term t, $freq(t/D_i)$ is the count of the term in domain corpus D_i, and $freq(w_j/D_i)$ is the count of word j from term t in the domain corpus. The 30% terms with the lowest values are filtered out.
3. **Domain consensus** checks whether a term is important, i.e., it appears in several documents. The definition of this filter is

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

   where $n\_freq(t, d_k)$ is the normalized count of term t in document d_k, which is a document in domain corpus D_i. Normalization is performed by dividing the calculated count with the maximum count of term t in any document in the domain corpus.
4. **Structural relevance** is used for measuring importance of specific terms. We assume that whenever a term appears in a title, that term is more likely to be of importance, and thus we should take it into account. Hence, the scores of all the terms that occur in the title of a domain corpus document get a score increment with constant value (k).

Finally, the filters are combined and a score is calculated by using the following formula on each remaining term. The score is defined as

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

where α and β are weights that respectively add more emphasis on either $DP_{D_i}(t)$ or $DC_{D_i}(t)$, and $max_t(DP_{Di}(t))$ and $max_t(DC_{Di}(t))$ denote the highest domain pertinence value and the highest domain consensus value found in

domain corpus D_i. 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 final score. Normalization has not been used for this method before, but we decided to use it, so that the influence of extreme values on the calculation of scores is reduced (for balancing 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 process of term extraction is optionally followed by a WSD procedure, which is used for deriving the sense of a concept term and for finding the synonyms of the concept term. In order to find the sense of a term, we follow an approach that is based on the SSI algorithm [12]. First, we retrieve the possible senses that are associated to the term. For this, we employ a semantic lexicon, which is a large lexical database that contains many words with their synsets (collections of synonyms). When a term is recognized as a noun in the semantic lexicon, we retrieve the possible synsets for that particular term. One of those synsets is selected as the sense for a term by following a number of steps. The best suited sense is determined 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_t is the set of possible senses for term t, C_t is the set of context senses, and $sim(s_i, c_j)$ is the similarity between possible term sense s_i and context sense c_j. The similarity measure we use is the one proposed by Jiang and Conrath [9], as this has proven to be an accurate and fast similarity measure [2].

After determining the proper sense, the disambiguated noun is added to a context list of disambiguated nouns for the document. This list of disambiguated nouns is initialized by either selecting all the monosemous nouns (nouns with only one sense) in the document or by applying Eq. 5 for the least ambiguous term and selecting the sense that yields the highest sum of pair-wise similarities for context terms (in case that there are no monosemous words).

After determining the sense of the concepts, the synonyms of each concept term are derived and stored. The semantic lexicon provides a list of terms that belong to a certain synset. Thus, when the term sense is known, the synonyms are extracted from the semantic lexicon. These terms are stored as alternative labels of a concept.

WSD is also used for removing concepts that have exactly the same synset as another concept from the list of concepts. For example, let us consider that there are 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 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, which is now merged with concept b.

3.4 Concept Hierarchy Creation

In order to build the actual concept hierarchy, the subsumption algorithm discussed in [14] is employed. In order to improve the performance of this algorithm we take the position of the ancestors with respect to the current node into account, which has not been done before. This model is based on concept co-occurrence. For each concept, the potential parent concepts (or subsumers) are determined. We can say that x potentially subsumes y if $P(x \mid y) \geq t, P(y \mid x) < t$, where t is a co-occurrence threshold. 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 added as a potential parent of y. When the potential parents are found, we have to select one of them as parent. The parent is determined by calculating a score for each potential parent:

$$score(p, x) = P(p \mid x) + \sum_{a \in A_p} w(a, x) \cdot P(a \mid x) \ , \tag{6}$$

where p is the potential parent node of x, A_p represents the list of ancestors of p, and $w(a, x)$ denotes a weight value with which each co-occurrence probability of ancestor a and x is multiplied. This weight is influenced by the amount of layers between ancestor a and node x. The weight value is defined as

$$w(a, x) = \frac{1}{d(a, x)} \ , \tag{7}$$

where $d(a, x)$ is the length of the path between node x and ancestor a. The reasoning behind this weighting is that the closer a parent node is to the concept node, the more influence it should have on the total score and thus on whether the potential parent is the parent that should be selected. When the scores for all the potential parent concepts are calculated, the potential parent with the highest score is chosen as the parent of x.

The main advantage of using the previous subsumption algorithm is its simplicity and therefore the speed of the algorithm. The subsumption algorithm is able to form a concept hierarchy in a relatively short amount of time due to a low number of computations. A disadvantage of this method is that it requires a large set of documents, which might not always be available.

4 ATCT Implementation

We implement the ATCT framework as a Java-based tool, which parses nouns from texts by means of a parser created by Stanford [10]. For presenting RDF representations we employ the Jena framework [11]. Our domain taxonomies are exported as RDF files using a SKOS vocabulary [1], so that we can compare our taxonomy with other taxonomies in this area, which are also RDF files defined by means of a SKOS vocabulary.

4.1 Term Extraction

In our implementation, we make use of a domain corpus, containing 25,000 abstracts extracted from two paper repositories, i.e., RePub and RePEc, which are divided into a set of abstracts on economics and management, as well as a contrastive set. By means of a POS tagger [10], we select the words that are tagged as a noun. Initial experiments show that, even though the processing speed of the parser is slow, the accuracy of this parser is relatively good, and better than faster methods.

4.2 Term Filtering

In the term filtering process we select the most relevant terms from the extracted terms. For this step we use different filters, that are implemented in a pipeline. First, the terms are processed through a domain pertinence filter where a value for domain pertinence is assigned to the terms. The 30% terms that have the lowest domain pertinence value are filtered out. After this, the lexical cohesion filter is applied on the remaining terms, in order to remove the 30% least relevant compound terms. Finally, the remaining terms are passed through a domain consensus filter, which computes term relevance based on domain consensus. Terms that occur frequently and consistently in the domain corpus have a high domain consensus value and have a higher chance to become domain concepts.

After applying all filters, a term score is calculated. The higher the score, the more relevant the term. To compute the score for a term, the domain pertinence value, the domain consensus value, and the structural relevance are taken into account. We evaluated different values of domain pertinence (α), domain consensus (β) and structural relevance (k). We investigated the effect of different values for α, β and k on the lexical precision (LP) of the built taxonomy. The LP is the amount of concepts in the built taxonomy that is lexically present in a reference ontology as well. The reference ontology we used is the STW Thesaurus for Economics and Business Economics (`http://zbw.eu/stw/`). After evaluating the LP values for a selection of parameters (ranging between 0 and 1 with an increment of 0.05), we obtain the optimal weights of $\alpha = 0.95$ (domain pertinence), $\beta = 0.05$ (domain consensus) and $k = 0.5$ (structural relevance). The results are logical, since domain pertinence values are higher for more specific domain terms, while the values for domain consensus and structural relevance can also be higher for more general terms.

With the optimal values for α, β and k a total of 2,000 terms are selected based on their domain score. Examples of selected terms that are well representative for the domain of economics and management are: {revenue, enterprise, forecasting, interest rate, pricing, portfolio, credit, wage, business cycle, entrepreneur}. All of these terms have high domain pertinence values, which shows that using a high value for α (which corresponds to more emphasis on domain pertinence) yields desirable results.

4.3 Word Sense Disambiguation

After the concepts have been created, the next (optional) step is the disambiguation of the terms. In the WSD process, the concept terms are assigned a sense. The possible senses are extracted from a semantic lexicon, i.e., WordNet [5]. WordNet contains a large amount of words and synsets from which the sense and synonyms of a concept can be derived. Therefore, it is well suited for use in our implementation. Our implementation is based on the SSI algorithm [12]. A term is only assigned a single sense. This means that a single term will only appear once as a concept label in the built taxonomy.

4.4 Concept Hierarchy Creation

With the concepts created after term filtering (and possibly after disambiguation), we are able to build a concept hierarchy. Our implementation makes use of the framework's subsumption algorithm as proposed in Sect. 3. Firstly, the potential parent concepts of a concept are determined. We evaluated for several values of threshold t (0 to 1 with step 0.05) and we found that a value of 0.2 yields a good balance in the trade-off between depth and the quality of relations of the built taxonomies. After applying the subsumption algorithm for all concepts, the hierarchy with the parent-child relations is created. This hierarchy is stored as an RDF file with a SKOS vocabulary.

5 Evaluation

For evaluation purposes, we compare two taxonomies generated with and without applying WSD by our implementation, each consisting of 2,000 distinct concepts, with a golden taxonomy, i.e., a preprocessed version of STW Thesaurus for Economics (`http://zbw.eu/stw/`), which is a manually created taxonomy in the field of economics and business economics. During the preprocessing of this ontology, the German terms were translated into English and some non-relevant terms about other subjects (such as law) were removed from this ontology. Also, some irrelevant tags that were used specifically for the STW taxonomy were removed. The final preprocessed version of STW still is a large taxonomy with thousands of terms more than our automatically generated taxonomy.

We also construct two taxonomies for the medical domain to investigate the impact of WSD on concept type-of relations on another domain. The reference ontology we use for the domain of medicine and health is the MeSH ontology, (`http://onto.eva.mpg.de/obo/mesh.owl`), which is a large ontology used to arrange medical subject headings. The taxonomy consists of 1,000 concepts and is constructed from a total of 10,000 documents from RePub.

5.1 Experimental Setup

We evaluate the built taxonomies on two levels, using measures from literature [4], as well as modifications of these measures. We use the lexical precision

(LP) and (LR) to evaluate to what degree the concepts of our constructed taxonomy are lexically shared with the golden taxonomy. LP and LR are defined as follows:

$$LP(O_C, O_R) = \frac{|C_C \cap C_R|}{|C_C|} , \tag{8}$$

$$LR(O_C, O_R) = \frac{|C_C \cap C_R|}{|C_R|} , \tag{9}$$

where O_C and O_R are the core ontology and reference ontology, respectively, C_C is the collection of concepts of the core ontology, and C_R represents the concepts of the reference ontology.

We also measure the quality of the type-of relations by using the common semantic cotopy (*csc*), which is the collection of a concept and the concept's sub- and super-concepts that are shared (including the concept) between a core ontology and a reference ontology. The definition of *csc* is:

$$csc(c, O_C, O_R) = \{c_i | c_i \in C_C \cap C_R \wedge (c_i \leq_{C_C} \ \ c \vee c \leq_{C_C} \ \ c_i)\} , \tag{10}$$

where c is a concept, O_C is the core ontology and O_R is the reference ontology, and C_c is the order induced by the type-of relations in the O_c ontology.

Two measures that apply the *csc* to measure the quality of type-of relations of an ontology are the global taxonomic precision (TP) and the global taxonomic recall (TR). To provide the definitions of TP and TR we first define the local taxonomic precision (tp) and the global taxonomic recall (tr). They are defined as follows:

$$tp_{csc}(c, O_C, O_R) = \frac{|csc(c, O_C, O_R) \cap csc(c, O_R, O_C)|}{|csc(c, O_C, O_R)|} , \tag{11}$$

$$tr_{csc}(c, O_C, O_R) = \frac{|csc(c, O_C, O_R) \cap csc(c, O_R, O_C)|}{|csc(c, O_R, O_C)|} , \tag{12}$$

where c is a concept, O_C is the core ontology, and O_R is the reference ontology. Both tp and tr depict the quality of the relations of a single concept.

With tp and tr defined, the definitions of TP and TR are as follows:

$$TP_{csc}(O_C, O_R) = \frac{1}{|C_C \cap C_R|} \sum_{c \in C_C \cap C_R} tp_{csc}(c, O_C, O_R) , \tag{13}$$

$$TR_{csc}(O_C, O_R) = \frac{1}{|C_C \cap C_R|} \sum_{c \in C_C \cap C_R} tr_{csc}(c, O_C, O_R) , \tag{14}$$

where O_c and O_r are respectively the core ontology and the reference ontology. By first computing the sum of tp values for each concept that is in the intersection of O_c and O_r and then dividing this sum by the number of shared concepts, the TP is computed. The TR is measured using the sum of tr values instead of the sum of tp values. Thus, TP reflects how similar the relations in the intersection of both ontologies are with respect to the core ontology, while TR reflects how

similar the relations in the intersection of the ontologies are with respect to the reference ontology.

Last, we apply the taxonomic F-Measure (TF), which is the harmonic mean of TP and TR, to retrieve the overall quality of the concept type-of relations. TF is defined as:

$$TF(O_C, O_R) = \frac{2 \cdot TP_{csc}(O_C, O_R) \cdot TR_{csc}(O_C, O_R)}{TP_{csc}(O_C, O_R) + TR_{csc}(O_C, O_R)} \ . \tag{15}$$

5.2 Experimental Results

When comparing the generated taxonomies with the golden standards, we obtain relatively low lexical precision and recall. For the domain of economics and management, lexical precision and recall are merely 0.1769 and 0.0926, respectively. This can be explained by analyzing the reference taxonomy we used. The STW taxonomy mainly uses the categories (abstract) in the economics and management domain, while our taxonomy uses specific terms in this domain. For the domain of medicine and health, lexical precision is somewhat higher, i.e., 0.2388, while the lexical recall of 0.0156 is very low. The latter low value is explained by the fact that the MeSH ontology consists of 15,337 concepts, which is much higher than the size of our built taxonomy (1,000 concepts).

In order to be able to properly evaluate the quality of taxonomies that are built by applying WSD we use the semantically shared concepts of the core ontologies and reference ontologies rather than the lexically shared concepts to retrieve the quality of the type-of relations as concepts are now disambiguated.

We have applied a WSD approach specific for existing taxonomies on the golden taxonomies to disambiguate its concepts in order to be able to retrieve the semantically shared concepts. The approach for disambiguating is mostly the same as the one we used for disambiguating concepts from text corpora. The difference is that we now use the surrounding taxonomy concepts of a concept to disambiguate a concept rather than using text surroundings. We define the surrounding taxonomy concepts of a concept as the concept neighborhood. The concept neighborhood consists of the concepts that are ancestors of a concept, and the concepts that are descendants of a concept.

It may occur that none of the concepts surrounding a taxonomy concept can be disambiguated. If none of the labels of the surrounding concepts are recognized by the used semantic lexicon, the sense of a concept can simply not be determined since there is no context to derive the meaning of a concept label. In such situations we select the most common sense of the concept.

A problem that arises with the disambiguation of taxonomy concepts is that concept labels may not be recognized by the used semantic lexicon. We cannot simply ignore concepts that we are not able to disambiguate. Therefore, for concepts that are impossible to disambiguate we also examine if those concepts are lexically represented in both the built taxonomy and the golden taxonomy. If the lexical representations of concepts appear in both taxonomies that does not necessarily mean that the concepts are semantically the same as well. We introduce a heuristic to investigate if lexically equivalent concepts share semantics. If

a concept from the built taxonomy is lexically represented in the golden taxonomy and the lexically shared concepts have a descendant or ancestor concept in common, either lexically or semantically, they will likely have the same meaning and are added to the ontology intersection. We do not consider the root node in this heuristic, as root nodes often are lexically represented as 'root' and concepts may then be wrongfully added to the intersection of the built taxonomy and golden taxonomy.

We investigated what percentage of concepts were disambiguated correctly for the ontologies for different amounts of ancestor and descendant layers. We found that for disambiguating a concept the best results are obtained when using a concept neighborhood that consists of two layers of ancestor concepts and two layers of descendant concepts. Further increasing the size of the concept neighborhood does not improve the results, while decreasing the concept neighborhood size lowers the percentage of concepts disambiguated correctly.

Table 1 shows several quality measurements for the type-of relations. We distinguish between the two domains as well as between applying the framework with and without WSD. The table shows the TP, TR, as well as TF. Here, T_1 denotes the taxonomy without WSD and T_2 denotes the taxonomy with WSD. For the economics and management domain, both taxonomies have high TF values, implying that the taxonomic relations between concepts in both taxonomies are good. The taxonomy with WSD seems to perform better than the taxonomy without WSD on all three quality measures. WSD improves TF by approximately 12.12%. WSD thus improves the quality of the concept type-of relations in our experiment. For the medicine and health domain, TF is again higher when using WSD, but the increase is not as high as for the domain of economics and management. Nevertheless, for all built taxonomies WSD had a positive effect on the overall quality of the type-of relations.

We compared results regarding TF for other taxonomies as well, to analyze if the improvement in quality holds true for not just one taxonomy. We constructed taxonomies using subsumption threshold values ranging from 0.1 to 0.6, while keeping the other parameters constant. As further increasing the subsumption threshold value results in taxonomies with a low depth, we decided not to incorporate taxonomies built with higher threshold values in our experiment. The TF values for the economics and management domain that correspond to the different t values are depicted in Fig. 2.

Table 1. Quality measurements for resulting taxonomies with and without the use of WSD, within the domain of economics and management (E&M), as well as within the domain of medicine and health (M&H).

Domain	Taxonomy	TP	TR	TF
E&M	T_1 (without WSD)	0.7382	0.5082	0.6023
	T_2 (with WSD)	0.8056	0.5813	0.6753
M&H	T_1 (without WSD)	0.5681	0.6051	0.5860
	T_2 (with WSD)	0.5907	0.6016	0.5961

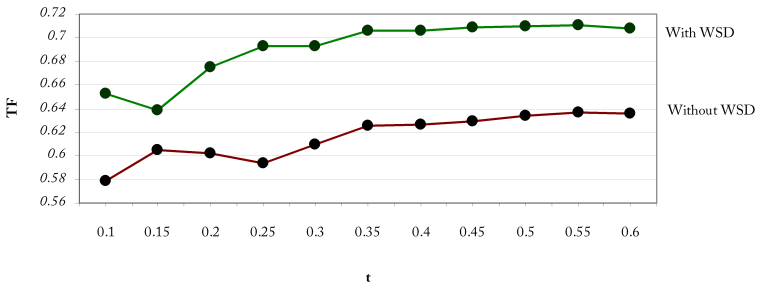

Fig. 2. Taxonomic F-Measure (TF) of taxonomies built with WSD compared to taxonomies constructed without WSD within the domain of economics and management.

The figure shows that WSD improves the overall quality of the type-of relations for all constructed taxonomies. A larger proportion of the type-of relations is shared between the built taxonomies and the golden taxonomy when using WSD. Furthermore, a higher t corresponds to a higher quality of type-of relations. The average increase in quality for the built taxonomies when using WSD is approximately 12.15%. The minimum and maximum increase in quality between a taxonomy built without WSD and one constructed with WSD are respectively 5.50% and 16.67%. A one-sided paired t-test with a 95% confidence interval shows that the increase in quality when using WSD is significant.

6 Conclusions

We have presented the ATCT framework for the automatic generation of a domain taxonomy from text. The framework extracts potential taxonomy terms from a large corpus, resulting in a number of the most relevant terms, after having filtered the potential terms for domain pertinence, domain consensus, lexical cohesion, and structural relevance. The filtered terms represent the taxonomy concepts. When disambiguating term senses, a sense and alternative labels (synonyms) are added to the concept. Concepts with the same senses are removed from the concept list. Subsequently, taxonomic relations are created by means of a subsumption method, which arranges concepts in a taxonomy according to their co-occurrence in documents. After implementation, we found in our experiments that the usage of WSD in automatic taxonomy construction improves the performance measured in terms of the TF-value by 12.15% with respect to the method without WSD.

For future research it would be interesting to benchmark our method against other taxonomy creation methods, such as hierarchical clustering or classification methods. Furthermore, we would like to investigate the impact of WSD on these other methods. Exploring other term extraction methods such as lexico-syntactic patterns in combination with our framework is an interesting topic as well. Finally, we would like to investigate the application of our approach to other domains, e.g., law, chemistry, physics, or history.

References

[1] Bechhofer, S., Miles, A.: SKOS Simple Knowledge Organization System Reference - W3C Recommendation 18 August 2009 (2009), Available at: http://www.w3.org/TR/2009/REC-skos-reference-20090818/

[2] Budanitsky, A., Hirst, G.: Semantic Distance in WordNet: An Experimental, Application-Oriented Evaluation of Five Measures. In: Workshop on WordNet and Other Lexical Resources, 2nd Meeting of the North American Chapter of the Assocation for Computational Linguistics (NAACL 2001). pp. 29–34. Association for Computational Linguistics (2001)

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

[4] Dellschaft, K., Staab, S.: On How to Perform a Gold Standard Based Evaluation of Ontology Learning. In: 5th Int. Semantic Web Conf. (ISWC 2006). Lecture Notes in Computer Science, vol. 4273, pp. 228–241. Springer (2006)

[5] Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press (1998)

[6] Gruber, T.R.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition 5(2), 199–221 (1993)

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

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

[9] Jian, J.J., Conrath, D.W.: Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In: 10th Republic of China Computational Linguistics Conf. on Research in Computational Linguistics (ROCLING 1997). pp. 19–33. The Association for Compuational Linguistics and Chinese Language Processing (1997)

[10] Klein, D., Manning, C.D.: Fast Exact Inference with a Factored Model for Natural Language Processing. In: 16th Annual Conf. on Neural Information Processing Systems (NIPS 2002). Advances in Neural Information Processing Systems, vol. 15, pp. 3–10. MIT Press (2002)

[11] McBride, B.: Jena: Semantic Web Toolkit. IEEE Internet Computing 6(6), 55–59 (2002)

[12] Navigli, R., Lapata, M.: Graph Connectivity Measures for Unsupervised Word Sense Disambiguation. In: Veloso, M.M. (ed.) 20th Int. Joint Conf. on Artificial Intelligence (IJCAI 2007). pp. 1683–1688. AAAI Press (2007)

[13] Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, Inc. (1986)

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

[15] Sclano, F., Velardi, P.: TermExtractor: a Web Application to Learn the Shared Terminology of Emergent Web Communities. In: 7th Conf. on Terminology and Artificial Intelligence (TIA 2007). Presses Universitaires de Grenoble (2007)

[16] Weber, N., Buitelaar, P.: Web-based Ontology Learning with ISOLDE. In: Workshop on Web Content Mining with Human Language at the 5th Int. Semantic Web Conf. (ISWC 2006) (2006), Available at: http://www.dfki.de/dfkibib/publications/docs/ISWC06.WebContentMining.pdf