TaxoLearn: a Semantic Approach to Domain Taxonomy Learning

Emmanuelle-Anna Dietz, Damir Vandic, Flavius Frasincar
Econometric Institute
Erasmus University Rotterdam
the Netherlands
dietz@iccl.tu-dresden.de, vandic@ese.eur.nl, frasincar@ese.eur.nl

***Abstract*—Building domain taxonomies is a crucial task in the domain of ontology construction. Domain taxonomy learning keeps getting more important as a form of automatically obtaining a knowledge representation of a certain domain. The alternative of manually developing domain taxonomies is not trivial. The main issues encountered when manually developing a taxonomy are the non-availability of a domain knowledge expert and the considerable amount of effort needed for this task. This paper proposes TaxoLearn, an approach to automatic construction of domain taxonomies. TaxoLearn is a new methodology that combines aspects from existing approaches, but also contains new steps in order to improve the quality of the resulted domain taxonomy. The contribution of this paper is threefold. First, we employ a word sense disambiguation step when detecting concepts in the text. Second, we show the use of semantics-based hierarchical clustering for the purpose of taxonomy learning. Third, we propose a novel dynamic labeling procedure for the concept clusters. We evaluate our approach by comparing the machine generated taxonomy with a manually constructed golden taxonomy. Based on a corpus of documents in the field of financial economics, TaxoLearn shows a high precision for the learned taxonomic concept relationships.**

***Keywords*-taxonomy learning; concept learning; word sense disambiguation**

I. Introduction

A taxonomy is the result of a classification task where categories are ordered in a hierarchical subclass structure. In recent years the extraction and the execution of domain specific taxonomies has become increasingly relevant. This is due to two main facts. First, it is a crucial process in information science, since it is usually a part of information extraction. Second, the manual construction of domain taxonomies is a time consuming task, which has to be done by experts of the considered domain. The purpose of taxonomy learning is to efficiently construct a taxonomy from a given corpus that identifies the main characteristics of the given data. The goal is that the constructed taxonomy contains all relevant concepts and their subclass relations for the domain of interest.

The main problem of constructing domain taxonomies is that it requires expert knowledge and that it is a time consuming task when done manually. This shows the need for an algorithm that can automatically deduce a taxonomy from existing data. According to literature on this topic [1], the process of taxonomy learning is based on many subtasks and issues that have to be addressed. One of the main issues is the data sparseness. Usually it is difficult to obtain enough data that covers the domain of interest thoroughly. However, taxonomy learning requires a large amount of data in order to build an appropriate domain taxonomy that covers the most important concepts. Another issue that needs to be addressed is whether only the syntactical structure of words or also their semantics should determine the concept hierarchy. We hypothesize that a domain taxonomy that is constructed with an algorithm that employs word sense disambiguation will show a more appropriate representation of the domain than an algorithm that does not use word sense disambiguation. Besides this, it is a challenge to select the most relevant concepts in an automatic way, as many unrelated concepts are also mentioned in texts that cover the domain of interest. Furthermore, the deduction of the relations between the concepts is recognized as a complex issue [2].

In this paper we propose TaxoLearn, a taxonomy learning methodology that addresses the previously mentioned issues. TaxoLearn is a corpus-based semantic approach to taxonomy learning that makes use of hierarchical clustering in order to construct a domain taxonomy. The primary focus of TaxoLearn are word sense disambiguation, concept similarity computation, and concept cluster labeling. To be able to evaluate the results, the constructed taxonomy is compared with a manually constructed golden taxonomy. For the evaluation, a large test corpus that covers the financial domain is collected as the test data.

II. Related work

Due to the urgency of the present issues, the research done in the field of taxonomy learning keeps increasing. In this section, we discuss three different approaches to taxonomy learning. We mention what aspects are used for the comparison in our performed evaluation and how TaxoLearn differs from each considered approach.

A. OntoGen

OntoGen [3] is a data-driven tool for semi-automatic topic ontology construction. The ontology construction process consists of the following steps: 'Document clustering',

978-0-7695-4880-7/12 $26.00 © 2012 IEEE
DOI 10.1109/WI-IAT.2012.129

'Term extraction', 'Term vocabulary extension', and 'Manual selection of relevant terms'. The first step concerns document clustering. For this purpose two different approaches are proposed. The first approach is the Latent Semantic Indexing and the second approach is the K-Means clustering.

The second step focuses on the terminology extraction from document clusters using a tool called TermExtractor. TermExtractor [4] automatically extracts the most relevant terms from a specific domain based on the available documents of this domain. This approach is also used in TaxoLearn, but we apply this method to concepts (synsets) from WordNet [5] instead of syntactical terms. The term extraction algorithm of this tool is based on the following filters:

- Domain Pertinence: the pertinence is high if a term is frequent in the domain of interest and much less frequent in the other domains used for contrast;
- Domain Consensus: the consensus is high if a term has an even probability distribution across the documents chosen to represent the domain;
- Lexical Cohesion: the cohesion is high if the words composing the (compound) term are more frequently found within the (compound) term than alone in the text;
- Structural Relevance: if a term is highlighted in a document it is increased by a factor k;
- Miscellaneous: A set of heuristics is used to remove generic modifiers (e.g., "large knowledge database", where "large" is the generic modifier);

The third step is about increasing the term vocabulary. The approach presented in [3] shows different techniques to retrieve more information about the terms extracted in the previous step. One such technique uses the Google search engine. This technique consists of obtaining the Google snippets about the considered term. This is done by executing a query with the term and an extra keyword related to the topic. This keyword is attached to the term to limit the search to the Web pages that are related to this topic. The returned snippets are used as an information source for the following step.

The fourth step is about choosing the concept names. This step is manually done with the support of the results obtained by the TermExtractor and the information gained in the third step (e.g., using Google). The results are then proposed to the user, who has to choose the most appropriate suggestion.

The main difference between OntoGen and TaxoLearn is that TaxoLearn focuses on the semantics of the taxonomy learning process. Our approach (automatically) produces a taxonomy of concepts and not a taxonomy of terms, like OntoGen does. A consequence of this is that we cannot directly benchmark our approach with OntoGen, as there is no precise way of comparing terms with concepts.

B. Formal Concept Analysis

The authors of [2] present an approach that is based on Formal Concept Analysis (FCA). In order to derive the correct relationships between objects and their descriptions, the syntactic dependencies between the verbs appearing in the text collection and their objects or subjects are extracted. Based on the definition of a formal concept, the verbs are assigned to a group of attributes and the objects and subjects are assigned to a group of objects. Three different information measures have been applied to calculate the importance of the relations from the derived object/attribute dependency. Because of the issue of data sparseness, only a few object/attribute relationships were extracted from the text corpus. Consequently, the authors applied a smoothing method. First, the similarities between objects are defined by applying five different similarity measures. Next, objects that occur with an attribute promote this attribute to their similar objects (so that these similar objects also occur with the attribute). This smoothing method overcomes the data sparseness problem, since it provides more information about the object/attribute relationship. Based on this information, a concept lattice representing the taxonomy is automatically derived for the domain of interest.

C. Another semi-automated approach

In [6] a semi-automated strategy for the purpose of extracting domain-specific taxonomies from Web documents is presented. The authors implement this method by computing a taxonomy that characterizes a scientific Web community. The process of this strategy is based on a sequence of automatic and manual steps. The approach consists of the following steps:

1) Collection of the Web documents of the domain of interest (manual);
2) Extraction of the relevant information (automatic);
    a) Selection of a list of the most relevant terms of the domain of interest;
    b) Search for the candidate definitions for the relevant terms;
    c) Filtering of the candidates to reduce noise.
3) Arranging and ordering the relevant information (automatic);
4) Construction of the taxonomy (manual).

The first step is to collect a large number of documents of the domain. Since the approach is implemented to characterize a scientific Web community, the documents mainly consist of communications between the members of a specific Web community. The second step addresses the extraction of the most relevant terms. This is done by applying filters such as the Domain Consensus. After having selected a list of the most relevant terms of the domain of interest, another list is created, consisting of candidate definitions about the relevant terms. These candidate definitions will serve as additional

information resources in a later step. Next, the Domain Pertinence filter and a style filter are applied to obtain the relevant definitions. The third step is about arranging and ordering the obtained relevant information. This step results in terms being ordered in a forest of taxonomically ordered sub-trees. The last step is about creating the actual domain taxonomy. In order to obtain a taxonomy, the sub-trees are linked together manually by experts who construct the final taxonomy.

Similar to the OntoGen approach, this approach does not focus on the semantics of the taxonomy learning process. The algorithm uses the terms in a document to construct a taxonomy, while our approach finds concepts from WordNet. Therefore, we are not able to provide a comparison between TaxoLearn and this approach.

III. TaxoLearn

In this section, the TaxoLearn methodology is explained by means of a running example, i.e., learning a financial taxonomy. TaxoLearn requires a corpus of documents that cover the domain of interest, but also corpora of documents that are unrelated to the domain of interest. These corpora of unrelated documents are used in certain parts of the process to help determine the relevant concepts. The following sections describe the main steps of the TaxoLearn framework.

A. Finding candidate concepts

In order to construct the domain taxonomy, we first need to find all noun phrases (NPs), these are part of potential taxonomy concepts. Many terms consist of more than one word, e.g., terms like "stock market". These terms could be NPs and would not be taken into account when searching only for NPs consisting of only one word. It can occur that NPs that are part of a taxonomy are composed of two or more words. To avoid that these relevant NPs would not be represented in the taxonomy, we also select NPs containing more than one word in the NPs extraction process.

Each NP has a defined meaning, which is not always only specified by its syntactical structure. Many NPs are syntactically identical but have different meanings depending on the context. TaxoLearn considers this issue and aims to disambiguate NPs in order to find the correct meaning. For this purpose, a local word sense disambiguation method that uses the Key Player Problem measure is applied [7]. KPP is considered as one of the best local word sense disambiguation measures. The trade-off between a local and a global word sense disambiguation method is that the global method is more accurate but the local method is faster. Due to the large data set that is used, we prefer a faster method. The definition of the KPP formula is as follows:

$$\mathrm{KPP}(v, V) = \frac{\sum_{u \in V : u \neq v} \frac{1}{d(u,v)}}{|V| - 1} \quad (1)$$

where u and v are synsets from WordNet [5], V is the set of every synset of every NP in the current sentence, $d(u, v)$ is the distance between synsets u and v. We compute $\mathrm{KPP}(v)$ for each synset of every NP in the sentence to determine the correct sense of each NP. Using the KPP measure, we apply the disambiguation procedure described in Algorithm 1.

Algorithm 1 Word Sense Disambiguation for NPs

Require: S is the set of sentences
Require: $\mathrm{getNPs}(s)$ returns all NPs for sentence s
Require: $\mathrm{getSynsets}(np)$ returns all synsets for noun phrase np
Require: $\mathrm{getAllSynsets}(s)$ returns every synset for every noun phrase in sentence s

$R = \emptyset$
for all $s \in S$ **do**
  $V = \mathrm{getAllSynsets}(s)$
  $NPs = \mathrm{getNPs}(s)$
  for all $np \in NPs$ **do**
    $kpp = 0$
    $chosenSense = \emptyset$
    for all $syn \in \mathrm{getSynsets}(np)$ **do**
      $newKpp = \mathrm{KPP}(syn, V)$
      if $newKpp > kpp$ **then**
        $kpp = newKpp$
        $chosenSense = syn$
      end if
    end for
    $R = R \cup \{(s, np, chosenSense)\}$
  end for
end for
return R

The main idea of this method is that every possible meaning of every NP is considered after which the distance between a specific meaning of this NP and all meanings of all the other NPs is computed. The sense of the NP that has the highest KPP value is likely to be the most correct sense in the context of the examined sentence. Based on these results, each NP in each sentence is assigned to a disambiguated sense from WordNet. After all the NPs in every sentence are disambiguated, the most relevant concepts can be identified, which is the subject of the next section.

B. Finding relevant concepts

Based on the information gained in the previous step, the most relevant concepts of the domain are detected. In this context a concept is defined as a NP with a disambiguated sense. Usually it is not sufficient to only check the frequency of a concept in order to determine its relevance. In general, many concepts would then be considered as relevant while they might be quite broad and not represent the specific domain at all. Words such as "paper" appear often in research documents while this word is not specific enough to be relevant for the domain of interest (e.g., the topic of

the research). In order to get the most relevant concepts, we evaluate two techniques that are proposed in [4].

1) Domain Pertinence: The first method is the so-called Domain Pertinence, which is high if the concept occurs significantly more frequently in the domain of interest than in other domains. This is computed by using the formula:

$$\mathrm{DP}(t, D^*) = \frac{\mathrm{freq}(c, D^*)}{\max_{j, D_j \neq D^*} (\mathrm{freq}(c, D_j))} \tag{2}$$

where c is the concept for which the pertinence is computed, and $\mathrm{freq}(c, D^*)$ is the number of times the concept occurs in the domain of interest D^*. That means that $\max_{j, D_j \neq D^*} (\mathrm{freq}(c, D_j))$ is the maximal number of times the concept occurs in one of the other domains. When a concept does not occur in any of the other domain, the denominator in Equation 2 would have the value 0. To avoid this, we increase the actual value of $\max_{j \neq i} (\mathrm{freq}(c, D_j))$ with 1 and use this in the computations. For example, consider the concept "portfolio", the WordNet definition of "a list of the financial assets held by an individual or a bank or other financial institution". This concept occurs 14 times in the domain of interest and 0 times in the other domains. $\mathrm{freq}(c, D^*)$ is computed then as 14 divided by 1. The Domain Pertinence of the concepts has been normalized with the following formula:

$$\mathrm{normDP}(\mathrm{c_i}, \mathrm{D}^*) = \frac{\mathrm{DP}(c_i, D^*)}{\max_j (\mathrm{DP}(c_i, D_j))} \tag{3}$$

where $\mathrm{DP}(c_i, D^*)$ is the Domain Pertinence of concept i in the corpus with the domain of interest and $\max_j \mathrm{DP}(c_i, D_j)$ is the highest value of the Domain Pertinence achieved in the set of all corpora. An empirically determined threshold is used to apply the Domain Pertinence as a filter to find the most relevant concepts.

2) Domain Consensus: The second method is the Domain Consensus, which is high if the concept is used consistently across the documents from the domain of interest. This is computed by the following formula:

$$\mathrm{DC}(c, D^*) = -\sum_{d_k \in D^*} \mathrm{norm\_freq}(c, d_k) \times \log\left(\mathrm{norm\_freq}(c, d_k)\right) \tag{4}$$

where c is a concept for which the consensus value is computed, and d_k is a document that covers the domain of interest D^*. The following normalization formula is used to compute $\mathrm{norm\_freq}(c, d_k)$:

$$\mathrm{norm\_freq}(c, d_k) = \frac{\mathrm{freq}(c, d_k)}{\max(\mathrm{freq}(c, D))} \tag{5}$$

where $\mathrm{freq}(c, d_k)$ is the frequency of times concept c occurs in document d_k from corpus D and $\max(\mathrm{freq}(c, D))$ is the highest frequency of times concept c occurs in general in a document from corpus D.

For example, consider the concepts "tax return" (the WordNet definition of "document giving the tax collector information about the taxpayer's tax liability") and "moral force" (the WordNet definition of "an efficient incentive"). It is clear that "tax return" is more likely to have an even probability distribution across the documents of the domain of finance. An empirically determined threshold is used to apply the Domain Consensus as a filter to determine the most relevant concepts.

C. Concept similarities

After having identified the most important concepts, the next step is to calculate the similarities between the chosen concepts. These similarities are then used to construct the taxonomy. In this paper, we evaluate three different approaches for the computation of concept similarities. The methods are called the WordNet method, the PMI method, and the Web method. The PMI and Web methods are both knowledge poor techniques that are only using the given input text corpus and the Web, respectively. The WordNet method is a knowledge rich method that uses WordNet, where all types of words are classified into sets of synonyms (synsets), each having a different meaning.

1) The WordNet method: The general idea of the WordNet method is to compute the similarities between two concepts by computing a distance in the WordNet graph. This is possible because WordNet is constructed as a network of synsets and in our case a concept corresponds to exactly one synset from WordNet. The similarity between two concepts is computed as follows [8]:

$$\mathrm{sim_{WN}}(c_i, c_j) = \frac{1}{d(c_i, c_j)} \tag{6}$$

The function $d(c_i, c_j)$ returns the smallest distance to the nearest common ancestor in WordNet of concepts i and j. If $\mathrm{sim_{WN}}(c_i, c_j)$ is close to 1, then c_i and c_j are similar to each other. If $\mathrm{sim_{WN}}(c_i, c_j)$ is close to 0, it means that the path in WordNet between c_i and c_j is large, and therefore these two concepts are not similar.

2) The PMI method: The Pointwise Mutual Information (PMI) method [9] is a method that takes, based on the text corpus, the frequencies of every word and the frequencies of every word pair in consideration. The PMI method computes a correlation measure. It is positive when words co-occur and negative otherwise. The similarity between two concepts is computed by the PMI method as follows:

$$\mathrm{sim_{PMI}}(c_i, c_j) = \log \frac{F_{c_i \cap c_j} / F_{all}}{(F_{c_i} / F_{all}) \times (F_{c_j} / F_{all})} \tag{7}$$

where F_{c_i} is the number of times concept i occurs in the text, F_{all} is the number of times all concepts occur in the text, and $F_{c_i \cap c_j}$ is the number of times concepts i and j occur together in the text. The main idea behind the PMI method is to use the concept co-occurrences as a measure

for their similarity. When the PMI value is positive then both concepts have a high co-occurrence and therefore seem to be related to each other. A negative PMI value indicates that the concepts are complementary to each other. When the PMI value is near 0 then there is no relationship between the two concepts.

3) The Web method: The Web method is implicitly a calculation based on the PMI method. The main difference is that it uses another source of knowledge. The Web method computes the similarity between two concepts as follows:

$$\mathrm{sim}_{\mathrm{WEB}}(c_i, c_j) = \log \frac{{}^{H_{c_i \cap c_j}}/_{H_{all}}}{\left({}^{H_{c_i}}/_{H_{all}}\right) \times \left({}^{H_{c_j}}/_{H_{all}}\right)} \qquad (8)$$

where H_{c_i} is the number of hits a search engine returns when searching for concept c_i, $H_{c_i \cap c_j}$ is the number of hits a search engine returns when searching for concept c_i and c_j, and H_{all} is the number of all pages indexed in English by a specific search engine. We set H_{all} to 10^{10}, as proposed in [10]. The evaluation of the different components is defined by the number of hits found by the search engine. So first both terms are searched separately, then they are searched together. The disadvantage of this method is that one cannot perform any word sense disambiguation on the concepts. This is due to the fact that it is not possible to specify the meaning of a word in a Web search engine.

D. Taxonomy construction

In order to construct the domain taxonomy, first a hierarchy based on the hierarchical clustering algorithm [11] is constructed. Hierarchical clustering eases the determination of the number of clusters as one can inspect the whole cluster tree (dendogram) and decide when to stop. Other algorithms, such k-means, force one to choose a priori the number of clusters, which is not known in advance.

The algorithm of constructing hierarchical clusters is applied separately for each similarity measure presented previously. To compute the distance between two clusters, the technique of average linkage clustering is employed. This means that the mean distance between two clusters is taken as the explicit distance to be considered for the following computations.

The other two techniques, the single or the complete linkage method, are not used. The main drawback of the single linkage method is that clusters that are not very similar can be put together if just two single entities in each of the clusters are very close to each other. The main drawback of the complete linkage method is that outliers have a high influence on the clustering process. We choose the average linkage clustering as it has shown to provide a good balance between these two extremes.

E. Taxonomy labeling

An important aspect when constructing hierarchical clusters for a representation of a taxonomy is the labeling process for each cluster. This can be done by using several methods. The authors of [2] and [12] propose hierarchy construction methods that use the hypernym information of concepts. In this case, one of the cluster labeling methods considers the hypernym information retrieved from WordNet. This method consists of taking the hypernyms of the concepts in the cluster. A hypernym represents a concept that has usually a general meaning and for which more specific concepts exist, the so-called hyponyms. Figure 1 shows a part of the constructed taxonomy where the two concepts "market_value" and "monetary_value"' are clustered together under the label of their common hypernym "worth". The second approach to this problem is to take the

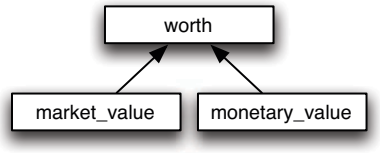

Figure 1: Cluster with hypernym label

centroid concept of a cluster as the label of that cluster. The centroid concept is the concept that is the closest to all other concepts.

The main advantage of the first approach (using WordNet) is that the hypernyms add information about the concepts in a cluster. Nevertheless, there are several problems with the first approach. First, the labels of the clusters can get too general when one uses the first approach. Second, one needs to check whether the chosen hypernym belongs to the domain of interest. For example, many concepts will have as common hypernym the concept "causality". This is due to the fact that similar concepts do not need to share the same specific hypernym, since this relation can be defined in various ways. Third, a hypernym is not a specific concept, but usually defines something that is related to a collection of more specific concepts. Obviously, this concept represents this individual cluster, but does not give any added value to the taxonomy as a whole.

The main advantage of the second approach (the centroid concept) is that there will always exist a centroid concept of the cluster that is specific enough. One disadvantage is that the chosen concept may not represent the sub-superclass relationship between the cluster label and the concepts in the cluster. Another disadvantage is that when the number of concepts in a cluster is large, the centroid concept does not have a representational function any more because the number of concepts in the cluster is too high to be represented appropriately by only one individual concept.

In order to improve the labeling of the clusters, TaxoLearn combines these two techniques into a hybrid approach. First, for every cluster we check whether a concept in the cluster

is already a hypernym concept of some other concept in this cluster. If this is the case, a threshold is used to calculate whether enough concepts in this cluster are hyponyms of this hypernym, the potential label concept. In order to get rid of the very general concepts, the depth of the considered hypernyms is limited. Only direct hypernyms and direct hypernyms of the hypernyms are taken into account.

For clusters that consist of only two concepts, we first check whether they have any hyperym in common. If this is the case, the hyperynm is set as the label of the cluster. This seems to be a reasonable decision, as the probability to find an appropriate hypernym for two concepts that already have been defined as similar is quite high. Additionally, it is not possible to find a centroid concept when only two concepts are representing the cluster. For all clusters consisting of two concepts, but not having any hypernym in common, a concatenation of the two concepts is used as label.

Clusters that contain more than two concepts, and do not pass the hypernym threshold (previously explained), are labeled using the second method, i.e., the centroid method. When the clusters consist of a large number of concepts, then more centroid concepts are considered as label of the cluster. This is done by first considering the 'most' centroid concept, then the next 'most' centroid one until enough centroids are found to label the cluster. The number of concepts labeling the cluster is determined by a threshold that is based on the total number of concepts in the cluster. This threshold is set such that for every three concepts in a cluster, one label is used. From this it follows that the number of labels used grows with the number of concepts in a cluster. This labeling process allows clusters that have many concepts to be labeled in such a way that the label adequately represents the meaning of the cluster.

IV. Evaluation

In this section, we evaluate the TaxoLearn framework. We first explain the measures that we used in the evaluation. Then, we present the data and domain that have been used to construct the taxonomy. Finally, we discuss the results of TaxoLearn with respect to the used evaluation measures.

A. Evaluation Measures

The main part of the evaluation is the comparison of each learned taxonomy with the golden taxonomy. The golden taxonomy is the taxonomy that is manually constructed using only knowledge from the corpus of the domain of interest. The comparison technique applied is based on the work proposed in [2].

It would be possible to compare the three taxonomies (each one based respectively on the WordNet method, the PMI method, and the Web method) with the labels that are given to the clusters. But since the manually built taxonomy is not labeled and based on the same concepts as the automatically constructed ones, another approach is needed. The comparison is done by comparing the concepts present in the clusters. We use the Jaccard similarity coefficient [13] with a threshold to determine when two clusters are equal to each other. The Jaccard similarity coefficient is defined as follows:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B - A \cap B|} \tag{9}$$

where, in our case, A and B are the clusters of concepts that are being compared to each other. The Jaccard similarity coefficient has a value between 0 and 1. The value is high when the clusters have many concepts in common and low when they do not share common concepts. For the evaluation, the clusters are considered to be equivalent when their Jaccard similarity is equal to or higher than 0.3. This value was obtained using a hill-climbing procedure on the used performance measures.

The first performance measure considers the lexical recall of the two taxonomies, i.e., calculate in how far the two taxonomies contain the same clusters. This is done by using the following formula:

$$LR(O_1, O_2) := \frac{|T_1 \cap T_2|}{|T_2|} \tag{10}$$

O_1 and O_2 are the core ontology's, i.e., the taxonomies that are considered, $|T_2|$ is the number of clusters in the golden taxonomy, and $|T_1 \cap T_2|$ is the number of clusters that are both in the golden and in the constructed taxonomy.

In order to compare the two hierarchies, we use the Semantic Cotopy (SC) presented in [14]. Given the concept $c \in O$, where c is a concept in the ontology O, the Semantic Cotopy is defined as follows:

$$SC(c, O) := \{c_i | c_i \in O \wedge (c_i \leq c \vee c \leq c_i)\} \tag{11}$$

where c_i is any concept in the ontology O. This means that c_i is either a sub-concept of c ($c_i < c$), i.e., a child, or the super-concept, i.e., the parent of c ($c < c_i$), or the equivalent of c ($c_i = c$).

The second performance measure is the so-called taxonomic overlap. This is defined with the following formula:

$$\overline{TO}(O_1, O_2) := \frac{1}{|O_1|} \times \sum_{c \in O_1} TO(c, O_1, O_2) \tag{12}$$

where

$$TO(c, O_1, O_2) := \begin{cases} TO^{'}(c, O_1, O_2), & c \in O_2 \\ TO^{''}(c, O_1, O_2), & c \notin O_2 \end{cases} \tag{13}$$

and $TO^{'}$ and $TO^{''}$ are defined as follows:

$$TO^{'}(c, O_1, O_2) := \frac{|SC(c, O_1) \cap SC(c, O_2)|}{|SC(c, O_1) \cup SC(c, O_2)|} \tag{14}$$

$$TO^{''}(c, O_1, O_2) := max_{c' \in C_2} \frac{|SC(c, O_1) \cap SC(c', O_2)|}{|SC(c, O_1) \cup SC(c', O_2)|} \tag{15}$$

where O_1 and O_2 are the two taxonomies that are compared to each other, and C_1 and C_2 are the concepts included in O_1 and O_2, respectively. More precisely, O_1 is the constructed taxonomy and O_2 is the golden taxonomy. The following step includes the calculation of the precision, which compares O_1 with O_2, the recall, which compares O_2 with O_1, and the F-Measure, which balances these two computations. This is done based on the $\overline{TO}(O_1, O_2)$ definition:

$$Precision : P(O_1, O_2) := \overline{TO}(O_1, O_2) \tag{16}$$

$$Recall : R(O_1, O_2) := \overline{TO}(O_2, O_1) \tag{17}$$

$$F\text{-}Measure : \tag{18}$$
$$F(O_1, O_2) := \frac{2 \times P(O_1, O_2) \times R(O_1, O_2)}{P(O_1, O_2) + R(O_1, O_2)}$$

The third and last performance measure is the harmonic mean of the lexical recall and the F-Measure, which is defined as follows:

$$F'(O_1, O_2) := \frac{2 \times LR(O_1, O_2) \times F(O_1, O_2)}{LR(O_1, O_2) + F(O_1, O_2)} \tag{19}$$

The range of the F-Measure is between 0 and 1. This can be interpreted as a percentage value that represents to what extent the two hierarchies are similar.

Based on the results that we obtain from the automatic comparison of all three computed hierarchical clusters with the golden taxonomy, it is possible to identify how the concept similarity methods affect the performance of the taxonomy construction algorithm.

B. Data

The source of the data collection has been the repository RePub [15], which provides access to the academic papers of the Erasmus University Rotterdam and makes them available online. With the help of the RePub personnel, 236 papers of the domain of Financial Economics were accessed. Furthermore all abstracts of the domain of medicine & health and law, culture & society have been provided by using the RSS feed from the RePub homepage.

TaxoLearn is evaluated by comparing the constructed taxonomies with a golden taxonomy that is manually constructed from the above mentioned corpus. At the moment there is no given golden taxonomy available related to the financial domain that contains approximately hundred terms and their related senses. This is the reason why we had to manually construct one for this purpose. This was done using only the knowledge obtained from the collected corpus, in order to keep the evaluation fair. After that a comparison takes place between the automatically constructed taxonomy and the golden taxonomy. As described in Section III-C, there are three alternative similarity measures used in order to construct the taxonomy. These three taxonomies, based on the WordNet, the PMI, and the Web method, are evaluated by comparing each one separately with the golden taxonomy.

C. Results

The results of our experiment are computed using the taxonomy comparison method described previously. For each comparison the lexical recall, precision, recall, F-measure, and F′-measure are calculated.

The lexical recall shows to what degree the clusters from the automatic constructed taxonomy are also represented in the golden taxonomy. The precision and the recall compare the clusters of the taxonomies overlap. The F-measure gives an overview by balancing the precision and the recall values. The F′-measure indicates to what degree the taxonomies are similar to each other.

Table I gives a detailed overview of the results for each method. The overview shows that the precision is relatively high for all three concept similarity methods. For the WordNet and PMI methods, the high precision is achieved because of the used semantic techniques. For the Web method, the huge amount of data available on the Web can be a possible explanation for the high precision. The high precision for all three similarity methods indicates that most of the SCs represented by the constructed taxonomies are also found in the golden taxonomy.

Table I: Detailed overview of the results

Evaluation Measures	WordNet	Web	PMI
Lexical recall	0.42	0.43	0.44
Precision	0.50	0.99	0.69
Recall	0.27	0.19	0.21
F-measure	0.35	0.32	0.32
F′-measure	0.38	0.37	0.37

All three methods have a quite low recall, with the Web method having the lowest value of 0.19. The recall is low because many relations are hidden in the text semantics and are difficult to be extracted by a fully automated algorithm. This indicates that only a few SCs in the golden taxonomy are also found in the constructed taxonomies.

Figure 2 shows a part of the taxonomy that is obtained for our use case, i.e., the financial domain. For example, we can see that the “financial gain” concept consists of a “net income” concept and a “return” concept.

V. Conclusion

In this paper we proposed TaxoLearn, a corpus-based semantic taxonomy learning framework. For the implementation of TaxoLearn, we aggregate and adapt steps from existing approaches. The contribution of this paper to the field of taxonomy learning is threefold. First, a word sense disambiguation method that improves the quality (precision) of the results is proposed. Second, we show the use of semantics-based hierarchical clustering for the purpose of taxonomy learning. Third, we propose a novel dynamic labeling procedure for concept clusters that allows large clusters to be labeled properly.

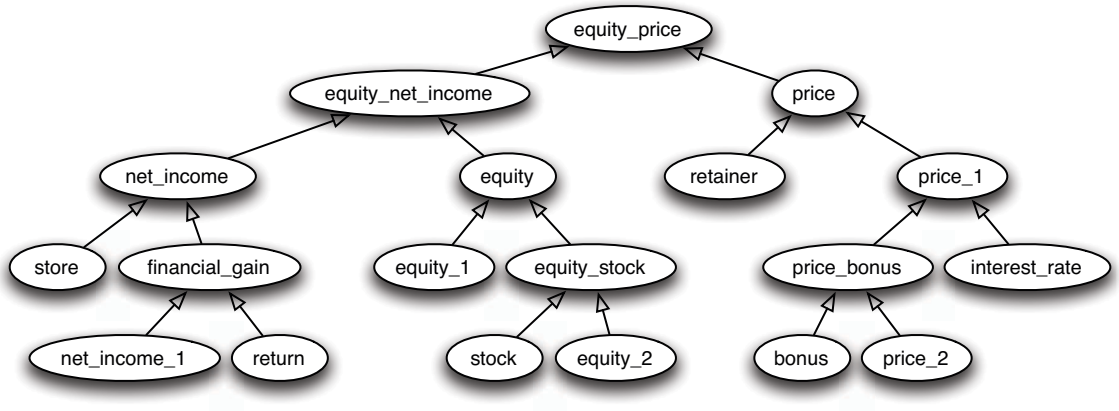

Figure 2: Example taxonomy with cluster labels and the corresponding senses

The results demonstrate a high precision for all three examined methods. This is mainly due to the semantic techniques that are employed in our approach. The recall has been found to be low for all three methods, this is probably because of the many relations that are hidden in the text semantics.

For future work we would like to evaluate TaxoLearn with a text corpus of another domain. Also, based on [16], we would like to improve the concept similarity using the three concept similarity methods as an ensemble technique.

REFERENCES

[1] P. Buitelaar, P. Cimiano, and B. Magnini, *Ontology Learning from Text: Methods, Evaluation And Applications*, ser. Frontiers in Artificial Intelligence and Applications. IOS Press, 2005.

[2] P. Cimiano, A. Hotho, and S. Staab, "Learning Concept Hierarchies from Text Corpora using Formal Concept Anaylsis," *Journal of Artificial Intelligence Research (JAIR)*, vol. 24, pp. 305–339, 2005.

[3] B. Fortuna, N. Lavrac, and P. Velardi, "Advancing Topic Ontology Learning through Term Extraction," in *Conference on Data Mining and Data Warehouses 2007 (SiKDD 2007)*. PASCAL EPrints (United Kingdom), 2007.

[4] P. V. F. Sclano, "TermExtractor: a Web Application to Learn the Common Terminology of Interest Groups and Research Communities," in *9th Conference on Terminology and Artificial Intelligence (TIA 2007)*. Springer, 2007, pp. 287–290.

[5] C. Fellbaum, *WordNet An Electronic Lexical Database*. The MIT Press, 1998.

[6] P. Velardi, A. Cucchiarelli, and M. Petit, "A Taxonomy Learning Method and Its Application to Characterize a Scientific Web Community," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 2, pp. 180–191, 2007.

[7] M. L. R. Navigli, "Graph Connectivity Measures for Unsupervised Word Sense Disambiguation," in *Joint Conference on Artificial Intelligence 2007 (IJCAI 2007)*, 2007, pp. 1683–1688.

[8] R. Navigli and M. Lapata, "An Experimental Study of Graph Connectivity for Unsupervised Word Sense Disambiguation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 678–692, 2010.

[9] K. W. Church and P. Hanks, "Word Association Norms, Mutual Information, and Lexicography," in *27th Annual Meeting on Association for Computational Linguistics (ACL 1989)*. Association for Computational Linguistics, 1989, pp. 76–83.

[10] M. Neshati, A. Alijamaat, H. Abolhassani, A. Rahimi, and M. Hoseini, "Taxonomy Learning Using Compound Similarity Measure," in *IEEE/WIC/ACM International Conference on Web Intelligence (WI 2007)*. IEEE Computer Society, 2007, pp. 487–490.

[11] G. N. Lance and W. T. Williams, "A General Theory of Classificatory Sorting Strategies," *Computer Journal*, vol. 9, pp. 373–380, 1967.

[12] S. A. Caraballo, "Automatic Construction of a Hypernym-Labeled Noun Hierarchy from Text," Ph.D. dissertation, 2001, adviser-Eugene Charniak.

[13] P. Jaccard, "Etude Comparative de la Distribution Florale dans une Portion des Alpes et des Jura," Bulletin del la Socit Vaudoise des Sciences Naturelles 37, pp. 547–579, 1901.

[14] A. Maedche and S. Staab, "Measuring Similarity between Ontologies," in *13th International Conference on Knowledge Engineering and Knowledge Management 2002 (EKAW 2002). Ontologies and the Semantic Web*. Springer-Verlag, 2002, pp. 251–263.

[15] RePub, "The EUR repository," 2012, http://repub.eur.nl.

[16] M. Sanderson and B. Croft, "Deriving Concept Herarchies from Text," in *22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1999, pp. 206–213.