Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 105–112
ISBN 978-83-60810-27-9
ISSN 1896-7094

Learning taxonomic relations from a set of text documents

Mari-Sanna Paukkeri*, Alberto Pérez García-Plaza†, Sini Pessala*, and Timo Honkela*
*Aalto University School of Science and Technology, Adaptive Informatics Research Centre
P.O. Box 15400, FI-00076 AALTO, Email: first.last@tkk.fi
†NLP & IR Group, E.T.S.I. Informática, UNED
28040, Madrid, Spain, Email: alpgarcia@lsi.uned.es

***Abstract*—This paper presents a methodology for learning taxonomic relations from a set of documents that each explain one of the concepts. Three different feature extraction approaches with varying degree of language independence are compared in this study. The first feature extraction scheme is a language-independent approach based on statistical keyphrase extraction, and the second one is based on a combination of rule-based stemming and fuzzy logic-based feature weighting and selection. The third approach is the traditional *tf-idf* weighting scheme with commonly used rule-based stemming. The concept hierarchy is obtained by combining *Self-Organizing Map* clustering with agglomerative hierarchical clustering. Experiments are conducted for both English and Finnish. The results show that concept hierarchies can be constructed automatically also by using statistical methods without heavy language-specific preprocessing.**

I. Introduction

An ontology is a directed graph consisting of concepts as nodes and relations as the edges between the nodes. A well-defined ontology has names for the concepts and specify what kind of relation there is between the concepts. The ontology is represented using a formal language, such as DARPA Agent Markup Language (DAML) or Web Ontology Language (OWL). With the use of the formal language, axioms can be specified to determine validity and to define constraints in ontologies. Ontologies are used widely in different Natural Language Processing (NLP) applications: examples are word sense disambiguation [1], annotating images [2], assessing text difficulty [3], and monitoring disease epidemics by analysing textual reports from the Web [4].

Creation of ontologies has obvious benefits, since it appears ideal that all systems within some domain would use similar terminologies and shared ontologies. The continuous change of conceptual systems through innovations and other activities requires that also ontologies need to be updated. The costs related to ontologies stem from two main sources: the development of a shared conceptual system and the use of it [5]. The development of an ontology typically consists of defining the concepts and the relationships between the concepts. The typical stages of an ontology building process are the following [6]: (1) domain analysis resulting into the requirements specification, (2) conceptualization resulting into the conceptual model, (3) implementation that leads into the specification of the conceptual model in the selected representation language, and (4) the ontology population i.e. the generation of instances and their alignment to the model that results into the instantiated ontology. Ontology maintenance includes getting familiar with and modifying the ontology. Reuse involves costs for the discovery and reuse of existing ontologies in order to generate a new ontology [6]. Simperl et al. [7] present a cost estimation model for ontology engineering. They estimate the person months associated to building, maintaining and reusing ontologies calculated as the product of the size of the ontology.

Ontologies have been mainly created manually but since the cost of manual creation is huge, also automated ways have been studied, often using methods developed originally for other fields of science. Ontology learning is the process of identifying terms, concepts, taxonomic and non-taxonomic relations and optionally axioms from natural language text and using them to construct and maintain an ontology.

In this paper, we assume that we have a set of text documents, each describing a concept in natural language. We extract taxonomic relations between the described concepts by using a selection of feature extraction methods and clustering the resulting feature vectors. This approach is automatic and easily portable to other domains and languages since it needs just a set of dictionary-like pages or documents as resources. One of our approaches is language-independent while the others use some language-dependent preprocessing steps.

A. Related work

Learning or extracting taxonomic relations means finding hypernymies between concepts with the goal of constructing a concept hierarchy. On the other hand, non-taxonomic relations are other than hyponymic relations between concepts in an ontology including e.g., synonymy and antonymy.

Taxonomic relations have been extracted automatically for a long time: Amsler created automatically a taxonomy for English nouns and verbs using dictionary definitions [8]. Hearst introduced lexico-syntactic patterns that indicate hyponymy relations [9]. Those have been further used for taxonomy learning e.g. in [10], [11]. Taxonomies have been learned also from Wikipedia by using the Wikipedia categories as concepts in a semantic network, connectivity of the network and on applying lexico-syntactic patterns [12]. Statistical methods for extraction of taxonomic relations have been covered in [13] including hierarchical and non-hierarchical clustering,

similarity measures and different linking schemes. [14] introduced their guided agglomerative hierarchical clustering algorithm that create concept hierarchies from text collections exploiting a hypernym oracle. The oracle exploits hypernyms from WordNet and using the Hearst lexico-syntacic patterns matched in a corpus or Internet. [15] proposed a clustering approach for taxonomy learning that incorporates evidence from multiple classifiers to optimize the entire structure of the taxonomy.

Also a combination of natural language processing tools have been used in extracting relations from text. A massive approach uses lemmatiser, syntactic parser, part-of-speech tagger, pattern-based classification and word sense disambiguation models together with resources such as domain ontology, knowledge base, and lexical databases [16]. Another approach is an unsupervised model for learning arbitrary relations between concepts of an ontology [17]. The approach uses corpus of manually tagged named entities, corresponding to ontology concepts and syntactic patterns.

In a recent work [18] the variety of ontology and concept hierarchy learning have been explained comprehensively. The work also introduces the Tree-traversing Ants (TTA) clustering technique for learning taxonomic relations. TTA is based on dynamic tree structures and it adopts a two-pass approach for term clustering. During the first pass, nodes are recursively broken into sub-nodes using Normalized Google Distance (NGD). The second pass is a refinement phase where terms are relocated according to n-degree of Wikipedia (noW) measure that uses Wikipedia Categories information.

The problem of the current methods is that they use of a wide range of language-specific tools, dictionaries or ontologies and thus exporting to new languages or domains is difficult or in some cases even impossible due to the lack of resources. The work by Wong seems to be more independent of the used language but instead needs access to Google and Wikipedia.

B. Our contribution

Our methodology can use any encyclopedia entries or other topic-related documents as definitions of concepts and creates taxonomic relations based on this data alone. No access to online sources is needed after the collection of the document set. Our methodology uses a very small amount of language-specific information and is thus easily portable to other languages. Other works that use Wikipedia use mostly the category information as concepts for taxonomy learning. Instead, we use the Wikipedia articles as concepts. Wikipedia articles have been used as concepts also in [19] but they do not create taxonomies but only measure relatedness between words or documents. In the existing work, term extraction methods are usually used for extraction of labels for concepts in the ontology, but we use keyphrase extraction for selection of a relatively small amount of terms for document representation.

II. Methods

We propose a methodology to generate taxonomic relations or concept hierarchies automatically from a set of encyclopedia documents. Each document is a description of a concept (the same assumption is made e.g. in [19]). These concepts are on the lowest level of the ontology and we aim to cluster the documents hierarchically to obtain an ontological structure of the concepts. Our methodology for generation of taxonomic relations consists of two basic steps: 1) feature extraction and 2) hierarchical clustering of the feature vectors. The result is a hierarchical structure generated according to the contents of the text documents.

In this study, generation of taxonomic relations is carried out for both English and Finnish languages. Finnish is a highly agglutinative language and as such relatively different from English. By using this language pair in our experiments we want to show that our methodology is further exploitable for several other languages.

A. Feature extraction

Feature extraction aims to reduce input data dimensionality, extracting the relevant information and removing redundancies. Traditional document representations are built over the *Vector Space Model (VSM)* [20], using term weighting functions based on term and document frequencies. The weighting is supposed to reflect the importance of each word to represent a particular document in the context of a document collection or corpus.

We propose three different approaches for document representation. One of them is a combination of heuristic criteria exploiting document structure by means of fuzzy logic. The second approach utilizes a statistical keyphrase extraction method to automatically extract features for document representation. The third method is the traditional *tf-idf* term weighting function that is also used as a baseline for our study. These representations are selected to compare how purely statistical method performs compared to a more heuristic method that gathers extra knowledge from the documents' meta information.

1) Fuzzy combination of criteria: When a human reader tries to understand the contents of a document, his or her attention is focused on some particular elements. Title or emphasized words are usually considered more important than the rest of the document. Moreover, the first and the last parts of a document usually contain overviews, summaries or conclusions with important keywords.

A fuzzy logic-based representation called *Extended Fuzzy Combination of Criteria* (*efcc*) [21] aims to exploit the semantics reflected by the use of a specific subset of HTML tags. The main idea is to define the importance of each word by combining several heuristic criteria. These criteria are related to word frequencies in the whole document, in titles, in emphasized text segments, or in first or last parts of a document. If similar information is available, the *efcc* approach can be used with any kind of document, not only with HTML-encoded documents.

The fuzzy system is built over the concept of linguistic variable, which value can be defined using natural language words and fuzzy sets. Each variable describes the membership degree of an object to a particular class. In the *efcc* approach, they are defined from human expert knowledge based on word frequencies mentioned above. Then, the knowledge base is defined by a set of IF-THEN rules combining these variables, in order to describe system behaviour as much precisely as possible. The aim of these rules is to combine one or more input fuzzy sets (antecedents), associating them with another output fuzzy set (consequent). Once the consequents of each rule have been calculated, and after an aggregation stage, the final set is obtained.

In this way, the knowledge used to build the rules is based on the following simple ideas: (1) A word appearing in the title or emphasized should appear in any other criterion to be considered important; (2) Words appearing in the first or last part of a document could be more important than others, because documents usually contain summaries or relevant ideas to attract reader's interest; (3) A non-emphasized word could mean that no words are emphasized in the web page; (4) A word not appearing in the title may indicate that the page has no title or the title has no meaning, i.e. it does not enclose relevant words; (5) High frequency of a word in a page could be important when the previous criteria are not enough to choose the most relevant words.

Some samples of these rules are (the rest can be found in [21]):

TABLE I
SAMPLE RULES

IF Title == High AND emph == High
THEN relevance = Very relevant
IF Title == High AND emph == Medium AND Position == Preferential
THEN relevance = High relevance
IF Title == High AND emph == Medium AND Position == Standard
THEN relevance = Medium relevance
...
IF Frequency == Medium
THEN relevance = Medium relevance

The inference engine is based on a center of mass algorithm (COM) that weights the output of each fired rule taking into account the truth degree of its antecedent. The output is a linguistic label with an associated number related to the relevance of a specific word in the page. A more detailed explanation of the fuzzy system can be found in ([21], [22] and [23]).

In addition to the *efcc* term weighting function, also inverse document frequency *idf* is used (equation 1).

$$\mathrm{idf_i} = \log \frac{|D|}{|\{d : t_i \in d\}|} \tag{1}$$

Where D is the total number of documents, and the denominator represents the number of documents containing the term i in the collection.

This modification of *efcc* aims to penalize those words appearing in a high number of documents due to the fact that Wikipedia pages are template-based and, therefore, there are terms appearing in each and every page (equation 2).

$$(\mathrm{efcc\text{-}idf})_{\mathrm{i,j}} = \mathrm{efcc_{i,j}} \times \mathrm{idf_i} \tag{2}$$

Where $\mathrm{efcc_{i,j}}$ is the relevance value obtained by means of the fuzzy system for a term i in a document j.

Moreover, discarding terms appearing in less documents than a particular percentage of the whole collection size can alleviate the effect of *idf* over too discriminative terms. If a concrete term appears in less than 4 documents, it is removed. This approach is called *df-efcc-idf*. Other intermediate variants were tested, like using just *efcc* values and the one shown in equation 2, but the results were uniformly worse.

To reduce the dimensionality of document vectors by selecting the most important features, we use the *Most Frequent Terms until n level* (*mft*) reduction method. This method consists of ranking the terms document by document based on the term weighting function values. A separate ranking is created for each document. In the first step, we will take terms appearing in the first position of each document ranking, ordering them first based on how many times a term has been found in different document rankings, and then, if two or more terms appear the same number of times in different rankings, based on the maximum weight found for each term. If we do not have terms enough, then we will take the terms appearing in second position in each ranking, and so forth.The process stops when the desired number of terms has been reached.

2) Statistical keyphrase extraction: The second feature extraction approach *Likey* [24] comes from the tradition of statistical machine learning. It extracts keyphrases of a document based on phrase frequencies. *Likey* does not use any language-dependent preprocessing tools or vocabularies and the only language-specific component needed is a reference corpus in each language. The basic idea in the method is to see whether the relative frequency of term candidates in the document collection is larger than their frequency in the reference corpus.

The *Likey ratio* [24] for each phrase is defined as

$$L(p, d) = \frac{rank_d(p)}{rank_r(p)}, \tag{3}$$

where $rank_d(p)$ is the rank value of phrase p in document d and $rank_r(p)$ is the rank value of phrase p in the reference corpus. Phrases are all the n-grams of the document up to a phrase length n. The rank values are the ordered frequencies of the phrases of the same length; the phrase having the largest frequency gets the rank of 1. In case of the same frequency value the rank value also stays the same. If the phrase p does not exist in the reference corpus the value of the maximum rank for phrases of length n is used: $rank_r(p) = max\_rank_r(n) + 1$. The *Likey ratio* is used to order the phrases existing in each document with those phrases having the smallest ratio being the best candidates for being a keyphrase.

As a post processing step, the phrases of length $n > 1$ face an extra removal process. If the reference rank value $rank_r$ of any of the single words constituting the phrase is smaller than the rank of the whole phrase, that means, the word is more common than the phrase, the phrase is removed. This uses the assumption that the maximum rank value is usually smaller for longer phrases than for unigrams since the frequencies of longer phrases are lower. In addition, lower rated subphrases of the existing keyphrase are also pruned.

The *Likey ratio* cannot be used directly as keyphrase weights since the best keyphrases get the smallest *Likey ratio* values. We thus scale the ratio to values between [0, 1], where values closer to 1 are the best keyphrases. The scaled weights are calculated with

$$w_2(p) = \begin{cases} (\frac{1}{t} - L(p,d)) * t & \text{if } L(p,d) < \frac{1}{t} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where $1/t$ is the maximum *Likey ratio* taken into account.

3) Tf-idf *term weighting:* As our third feature extraction method and also the baseline, we use the traditional *tf-idf* term weighting scheme (equation 5).

$$(\text{tf-idf})_{\text{i,j}} = \text{tf}_{\text{i,j}} \times \text{idf}_{\text{i}} \quad (5)$$

Where $\text{tf}_{\text{i,j}}$ is the number of times a given term i appears in document j.

We weight the unigrams of each document with the other documents as a reference. To reduce vector size, terms with high and low document frequencies are discarded.

This is selected as the baseline method because it is a very well known representation used nowadays in many fields like clustering, classification, information retrieval, etc. *Tf-idf* has also been used as a baseline in several unsupervised learning and clustering works that combine VSM with Self-Organizing Maps, e.g. [25], [26], and [27].

B. Hierarchical clustering

To learn the taxonomic relations between concepts, we assume that each concept has a hypernym, i.e. a parent node. Moreover, we assume that there is a maximum number of distinct hypernyms among the concepts. The concepts are represented by term feature vectors, and *Self-Organizing Map (SOM)* [28] is used to create an ordered space of the concept vectors. The *SOM* is an artificial neural network that orders data using unsupervised learning. Even though the *SOM* is a good method for reducing dimensionality and finding a topological order of data, it does not perform the explicit classification task. Because of this, hierarchical clustering of the prototype vectors of the *SOM* is applied on the top of the *SOM* map to define the borders of clusters. We use agglomerative hierarchical binary clustering to produce up to m clusters for the *SOM* map. The next levels are obtained by training a new *SOM* separately for the feature vectors in each cluster and using agglomerative hierarchical clustering for the new *SOM* maps.

A *K*-level concept hierarchy is built by first assuming that the zero-level data (concepts, i.e. term vectors) constitute the zero-level cluster, i.e., the root node. The zero-level data are then clustered into first-level clusters. The feature vectors in each of the level k clusters are further clustered to get the $k + 1$-level clusters and thus a more fine-grained clustering until $k = K$. The level K concepts are then the original concepts represented by the feature vectors. Somewhat similar approach is the *growing Self-Organizing Map* [29] and further the *growing hierarchical Self-Organizing Map* [30]. The growing of the map in that case is based on variations of Average Quantization Error, expanding single neurons instead of clusters.

Another possibility in this study would have been to use hierarchical clustering algorithm directly to the feature vectors but that would have needed another method that reduces the resulting binary tree into a small number of hierarchy levels.

C. Evaluation methodology

The evaluation of automatically generated ontologies may concentrate on different levels: evaluation of the lexical layer, hierarchy, or context level [31]. Four approaches for evaluating ontologies have been considered in the literature: comparison to a gold standard (which may itself be an ontology), evaluation as a part of an application, comparison to a source data about the domain, and evaluation by humans [31]. We follow the first approach and compare the generated conceptual hierarchies to a manually constructed reference ontology.

To be able to compare the hierarchical structure to an ontology, we label the concepts of the generated hierarchical structure with the terms (concept names) in the reference ontology. First, the topic of each text document is extracted from the incorporated meta data of the documents. The topics are used to label concepts on the lowest-level in the generated concept hierarchy. Also the parents for each topic are extracted from the reference ontology. The clusters on the generated hierarchy act as the hypernyms (parent concepts) and they are labeled according to the concepts forming the cluster. The majority of the parents is selected as the label for the cluster, i.e., the hypernym. In a case of two parent candidates of equal sizes the hypernym is selected randomly.

Each label can exist only once in the hierarchy in evaluation. Therefore, for labeling purposes, clusters having both the same label and the same parent are merged. In a case where the clusters with the same label have different parents, the cluster having less concepts assigned with that label is considered as unclassified. In a case of equal size, the unclassified cluster is selected randomly. In this way, we penalize similar clusters when they are far away, avoiding penalizing the errors due to too fine-grained clustering.

We use the TP_{csc} evaluation metrics [32] for evaluation of the automatically generated ontologies. Also Wilks and Brewster [33] and Brewster et al. [34] have performed their evaluation by a gold standard using these metrics, but not using reference labels. The experiments are focused on finding single

concepts and their relations, and not the whole hierarchy as is our case.

Dellshcaft and Staab [32] introduce TP_{csc}, TR_{csc} and TF_{csc} values for the evaluation of a concept hierarchy of an ontology. The TP_{csc} metric is a global taxonomic precision TP, which uses the common semantic cotopy csc. The common semantic cotopy csc is defined as a set of concept identifiers, which are sub concepts $c <_{\mathcal{C}_1} c_1$ and super concepts $c >_{\mathcal{C}_1} c_1$ of the concept identifier c_1 in concept a hierarchy $\mathcal{C}_1$ and which are also concept identifiers in the ontology $\mathcal{O}_2$

$$csc(c, \mathcal{O}_1, \mathcal{O}_2) = \{c_i | c_i \in \mathcal{C}_1 \cap \mathcal{C}_2 \wedge (c_i <_{\mathcal{C}_1} c \vee c <_{\mathcal{C}_1} c_i)\}. \quad (6)$$

Local taxonomic precision tp_{csc} compares common semantic cotopies of concepts of $c_1 \in \mathcal{C}_1$ and $c_2 \in \mathcal{C}_2$

$$tp_{csc}(c_1, c_2, \mathcal{O}_1, \mathcal{O}_2) = \frac{|csc(c_1, \mathcal{O}_1, \mathcal{O}_2) \cap csc(c_2, \mathcal{O}_1, \mathcal{O}_2)|}{|csc(c_1, \mathcal{O}_1, \mathcal{O}_2)|}. \quad (7)$$

The global taxonomic precision TP_{csc} metric is based on local taxonomic precision of the common concepts of a learned ontology $\mathcal{O}_C$ and the reference ontology. The value of TP_{csc} tells how many of the semantic relations of the learned ontology can be found in the reference ontology

$$TP_{csc}(\mathcal{O}_C, \mathcal{O}_R) = \frac{1}{|\mathcal{C}_C \cap \mathcal{C}_R|} \sum_{c \in \mathcal{C}_C \cap \mathcal{C}_R} tp_{csc}(c, c, \mathcal{O}_C, \mathcal{O}_R). \quad (8)$$

The global taxonomic recall TR_{csc} metric is calcuated using global taxonomic precision TP_{csc}

$$TR_{csc}(\mathcal{O}_C, \mathcal{O}_R) = TP_{csc}(\mathcal{O}_R, \mathcal{O}_C). \quad (9)$$

The taxonomic F-measure TF_{csc} is the harmonic mean of the global taxonomic precision and recall.

$$TF_{csc}(\mathcal{O}_C, \mathcal{O}_R) = \frac{2 \cdot TP_{csc}(\mathcal{O}_C, \mathcal{O}_R) \cdot TR_{csc}(\mathcal{O}_C, \mathcal{O}_R)}{TP_{csc}(\mathcal{O}_C, \mathcal{O}_R) + TR_{csc}(\mathcal{O}_C, \mathcal{O}_R)} \quad (10)$$

III. Data

The data used in this study consists of a document collection from Wikipedia that is used in learning of taxonomic relations, evaluation data that is a manually constructed ontological structure of the text documents, and a reference corpus for the *Likey* keyphrase extraction method.

A. Wikipedia articles

The data for concept representation are collected as HTML pages from the English[1] and Finnish Wikipedia[2]. The Wikipedia data are articles about animals (mammals and birds) in both English and Finnish. The data are collected manually using the meta information from both categories and information boxes. The data were collected originally for Finnish such that the length of each article exceeds 200 words to be able to extract keyphrases of sufficient quality, resulting 119 articles. The English articles are the Wikipedia-provided translations of the Finnish articles, resulting 113 articles due to five too short articles and one article that is linked from two Finnish articles. These data sets are available on our web pages.

[1]http://en.wikipedia.org, accessed on 12th January 2010
[2]http://fi.wikipedia.org, accessed on 13th August 2009

B. Reference ontology

The evaluation data are collected from the Wikipedia articles. Most of the Wikipedia articles about animals contain a separate meta information box explaining the scientific classification of the animal. This meta information is collected to construct the reference ontology manually. In our evaluation, we use a simplified version of the scientific classification: just three levels of hierarchy are taken into account besides the actual articles. The three levels were inherent in the document collection and it is also the first non-trivial number of levels.

The Wikipedia articles are situated as the leaf nodes, located on the third level of the reference ontology. Their topics are about different Families, Subfamilies or Species, e.g. Black Grouse, Galapagos Hawk and Jaguar. In the English reference ontology there are 113 third-level concepts and in the Finnish ontology 119 concepts. Each third-level concept has a super concept on the second level, which consists of nine different animal Orders (according to the scientific classification) in both languages, e.g. the parent of Black Grouse is Order *Galliformes*, Galapagos Hawk is of Order *Accipitriformes* and Jaguar is of Order *Carnivora* (see Figure 1). There are two concepts on the first level in both languages: the Classes *Mammalia* (parent of e.g. *Carnivora*) and *Aves* (parent of e.g. *Accipitriformes* and *Galliformes*). The root concept of both of the reference ontologies is Kingdom *Animalia*. Both ontologies have 5 subclasses (*Orders*) for *Mammalia* and 4 subclasses for *Aves*. In the Finnish ontology, 84 concepts on the third level (79 in English) belong to Class *Mammalia* and 35 (34) to Class *Aves*.

C. Europarl corpus

The statistical keyphrase extraction method *Likey* requires a reference corpus that is a sample of the general language. We use English and Finnish parts of Europarl, European Parliament plenary speeches [35] as the reference corpus. Our preprocessing excludes all XML tags containing some meta data and results in the sizes of 35 758 149 word tokens in English and 22 676 344 word tokens in Finnish.

IV. Experiments

Our concept hierarchy generation process is based on unsupervised learning. As ontologies are structured by default, in this first approach we decided to select manually the maximum number of clusters on each hierarchy level to simplify the process of comparison to the reference ontology. In our experiments, the maximum was two clusters on the first level and ten clusters (5 + 5) on the second level. If the hierarchical

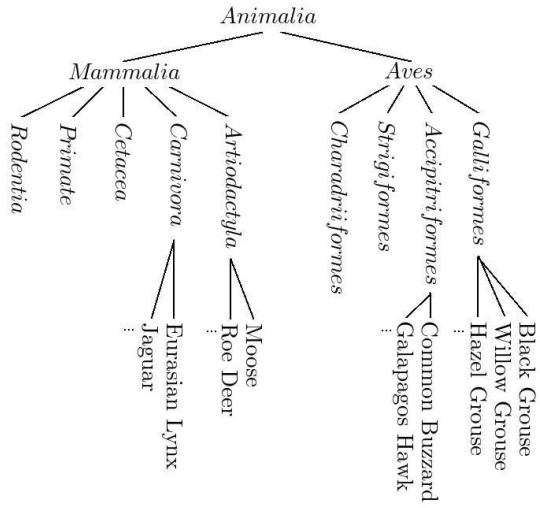

Fig. 1. Part of the reference ontology.

clustering was not able to find the total number of classes, a smaller amount was accepted.

On the first level, we trained a 3x3-cell (5x5 cells in a larger map) *SOM* with normalization of the variance and batch training using SOM Toolbox[3]. On the second level, the map size was slightly larger, 5x5 cells (7x7 cells). These sizes were selected according to our preliminary tests where we found that larger maps (up to 20x20 cells) do not perform very well. Hierarchical clustering was carried out with correlation distance using complete linking scheme. In the preliminary tests, we also used other distance measures (e.g. cosine and spearman) and linking schemes (single and average). Correlation and cosine distance performed very similarly but spearman distance usually slightly worse. Overall, the differences were very small. Single and average linking had the problem that with our implementation in Matlab there were cases where the clustering could not be found.

A. Fuzzy-based representation

In the *efcc* feature extraction approach, the data was preprocessed in the following way. A set of stop words for each language was used to remove common words. The HTML-specific entities were converted to their corresponding characters or discarded, depending on the case. The punctuation marks were also removed. Finally, suffixes were removed using a language dependent stemmer: a standard implementation of Porter's algorithm for English and Snowball Finnish stemmer[4].

The number of features per document vector was chosen to be 100 since that had been enough in our preliminary tests. A second representation for both languages is selected to be approximately the sizes used by the keyphrase extraction method. The second vector size is 913 for English and 1157 for Finnish.

[3]http://www.cis.hut.fi/somtoolbox/

[4]http://snowball.tartarus.org/algorithms/finnish/stemmer.html

B. Keyphrase extraction

The keyphrase extraction method uses only the plain text portion of the Wikipedia HTML pages without any meta data and is thus a language-independent approach. The Wikipedia preprocessing produces plain text by removing HTML and Wikipedia markup-specific tags, figures, tables, lists, links, and references. Preprocessing for both Wikipedia articles and Europarl reference corpora in Finnish and English removes punctuation and changes numbers to a <NUM> tag. Note that no stemming nor other linguistic filtering takes place in this approach.

We selected the first 15 keyphrases from each article and weighted them with a scaled Likey weight (Eq. 4) that might decrease to zero. We used a threshold value $1/t = 0.01$ since in most of the documents the *Likey ratio* for the first 15 keyphrases is less than that. The resulting feature vectors have 934 features in English, and 1211 features in Finnish.

C. Tf-idf *baseline*

For *tf-idf* the same preprocessing as for *efcc* representation was carried out. The number of features per document vector were selected in the same manner than *efcc* representation, described in section IV-A.

D. Results

The results of the experiments for English Wikipedia documents are given in Table II. The global taxonomic precision (TP), recall (TR) and F-measure (TF) results for the three feature extraction approaches are presented. For *df-efcc-idf* and similarly for *tf-idf*, results for vectors with 100 features (df-efcc-idf 100 and tfidf 100, respectively) and 913 features (df-efcc-idf 913 and tfidf 913, respectively) are shown. Also the results for *Likey* are presented. The results are calculated as averages of the two different map sizes.

TABLE II
RESULTS FOR ENGLISH FOR DIFFERENT REPRESENTATIONS. TP, TR, AND TF STAND FOR GLOBAL TAXONOMIC PRECISION, RECALL AND F-MEASURE, RESPECTIVELY.

Representation	TP	TR	TF
df-efcc-idf 100	0.782	0.900	0.836
df-efcc-idf 913	0.725	0.956	0.825
Likey	0.764	0.886	0.820
tf-idf 100	0.707	0.853	0.772
tf-idf 913	0.698	0.869	0.774

The second experiment is for the Finnish language with the same parameters and representatons than for English, except for the vector size of the second representation of *df-efcc-idf* and *tf-idf* is 1157. The results are presented in Table III.

These results show that for both English and Finnish languages a level of about 80% in Taxonomic F-measure can be achieved within the task of generating hierarchical structures. For the Finnish language, the language-independent feature extraction approach *Likey* does not perform as well as the other approaches. Anyway, the stemming preprocessing step is missing from the *Likey* results and that might explain the poorer performance at least partly.

TABLE III
RESULTS FOR FINNISH FOR DIFFERENT REPRESENTATIONS. TP, TR, AND TF STAND FOR GLOBAL TAXONOMIC PRECISION, RECALL AND F-MEASURE, RESPECTIVELY.

Representation	TP	TR	TF
df-efcc-idf 100	0.734	0.872	0.796
df-efcc-idf 1157	0.722	0.852	0.781
Likey	0.685	0.841	0.755
tf-idf 100	0.779	0.847	0.812
tf-idf 1157	0.837	0.865	0.851

For the English language, the heuristic fuzzy logic-based *df-efcc-idf* performs better than the statistical approaches. This is of course a natural result since *df-efcc-idf* exploits more knowledge by using the semantic information of the HTML structure.

The results seem to be not very consistent. This may be due to the fact that a small difference in the *SOM* clustering may have large effect in the resulting ontology. If *SOM* were initialized randomly instead of using the default setting of linear initialization along the two greatest eigenvectors, different results could be obtained on different runs and more precise results reached.

V. CONCLUSIONS AND DISCUSSION

In this paper, we have presented an automated methodology for concept hierarchy generation from a set of text documents. We used three different representations of the documents: 1) a combination of rule-based stemming and fuzzy logic-based feature weighting and selection, 2) automatic keyphrase extraction and 3) the traditional *tf-idf* measure with rule-based stemming. The hierarchy generation has been run by a hierarchical approach of the Self-Organizing Map (*SOM*) together with agglomerative hierarchical clustering. The experiments have been conducted for English and Finnish to show the applicability to different kinds of languages.

We used more than 100 Wikipedia articles about *Animalia* as our data for both English and Finnish. We also created reference ontologies out of the Wikipedia articles for both languages. In the future work, a much larger collection of Wikipedia articles could be used to obtain larger number of levels in the ontology. We also want to exclude the information about the amount of clusters needed for building the hierarchy. Another future improvement consist of extracting concept identifiers from the corpus instead of generating just the taxonomy. Any of our representations could be utilized also here.

REFERENCES

[1] D. Yuret and M. A. Yatbaz, "The noisy channel model for unsupervised word sense disambiguation," *Computational Linguistics*, vol. 36, no. 1, pp. 111–127, 2010.

[2] T. Ruotsalo, L. Aroyo, and G. Schreiber, "Knowledge-based linguistic annotation of digital cultural heritage collections," *IEEE Intelligent Systems*, vol. 24, no. 2, pp. 64–75, 2009.

[3] N. Duran, C. Bellissens, R. Taylor, and D. McNamara, "Quantifying text difficulty with automated indices of cohesion and semantics," in *Proceedings of the 29th Annual Meeting of the Cognitive Science Society*, 2007, pp. 233–238.

[4] R. Steinberger, F. Fuart, E. van der Goot, C. Best, P. von Etter, and R. Yangarber, "Text mining from the Web for medical intelligence," in *Mining Massive Data Sets for Security*, F. Fogelman-Souli, D. Perrotta, J. Piskorski, and R. Steinberger, Eds. The Netherlands: IOS Press, 2008.

[5] T. Honkela, V. Könönen, T. Lindh-Knuutila, and M.-S. Paukkeri, "Simulating processes of concept formation and communication," *Journal of Economic Methodology*, vol. 15, no. 3, pp. 245–259, 2008.

[6] M. Fernández-López and A. Gómez-Pérez, "Overview and analysis of methodologies for building ontologies," *Knowledge Engineering Review*, vol. 17, pp. 129–156, 2002.

[7] E. P. B. Simperl, C. Tempich, and Y. Sure, "A cost estimation model for ontology engineering," in *International Semantic Web Conference 2006*, 2006, pp. 625–639.

[8] R. A. Amsler, "A taxonomy for English nouns and verbs," in *Proceedings of the 19th annual meeting on Association for Computational Linguistics*, Standford, CA, 1981, pp. 133–138.

[9] M. A. Hearst, "Automatic acquisition of hyponyms from large text corpora," in *Proceedings of the 14th International Conference on Computational Linguistics*, 1992, pp. 539–545.

[10] S. A. Caraballo, "Automatic construction of a hypernym-labeled noun hierarchy from text," in *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*. Association for Computational Linguistics, 1999, pp. 120–126.

[11] M. Pennacchiotti and P. Pantel, "A bootstrapping algorithm for automatically harvesting semantic relations," *Proceedings of Inference in Computational Semantics (ICoS-06)*, 2006.

[12] S. Ponzetto and M. Strübe, "Deriving a large scale taxonomy from Wikipedia," in *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, Vancouver, Canada, 2007, pp. 1440–1445.

[13] A. Maedche, V. Pekar, and S. Staab, "Ontology learning part one – on discovering taxonomic relations from the Web," in *Proceedings of the Web Intelligence conference*, 2002, pp. 301–322.

[14] P. Cimiano and S. Staab, "Learning concept hierarchies from text with a guided hierarchical clustering algorithm," in *Proceedings of Workshop on Learning and Extending Lexical Ontologies by using Machine Learning Methods at ICML 2005*, 2005.

[15] R. Snow, D. Jurafsky, and A. Y. Ng, "Semantic taxonomy induction from heterogenous evidence," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the ACL*. Association for Computational Linguistics, 2006, pp. 801–808.

[16] L. Specia and E. Motta, "A hybrid approach for extracting semantic relations from texts," in *Proceedings of 2nd Workshop on Ontology Learning and Population*, Sydney, 2006, pp. 57–64.

[17] M. Ciaramita, A. Gangemi, E. Ratsch, J. Saric, and I. Rojas, "Unsupervised learning of semantic relations between concepts of a molecular biology ontology," in *Proceedings of the 19th international joint conference on Artificial intelligence*, 2005, pp. 659–664.

[18] W. Y. Wong, "Learning lightweight ontologies from text across different domains using the web as background knowledge," Ph.D. dissertation, The University of Western Australia, September 2009.

[19] E. Gabrilovich and S. Markovitch, "Computing semantic relatedness using Wikipedia-based explicit semantic analysis," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, Hyderabad, India, 2007, pp. 1606–1611.

[20] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[21] A. García-Plaza, V. Fresno, and R. Martínez, "Web page clustering using a fuzzy logic based representation and self-organizing maps," in *Web Intelligence*, 2008, pp. 851–854.

[22] V. Fresno, "Representacion autocontenida de documentos html: una propuesta basada en combinaciones heuristicas de criterios," Ph.D. dissertation, DITTE, URJC, 2006.

[23] A. Ribeiro, V. Fresno, M. C. Garcia-Alegre, and D. Guinea, "A fuzzy system for the web page representation," *Studies in Fuzziness and Soft Computing*, vol. 111, pp. 19–37, 2003.

[24] M.-S. Paukkeri, I. T. Nieminen, M. Pöllä, and T. Honkela, "A language-independent approach to keyphrase extraction and evaluation," in *Coling 2008: Companion volume: Posters*. Manchester, UK: Coling 2008 Organizing Committee, August 2008, pp. 83–86.

[25] J. Bakus, M. Hussin, and M. Kamel, "A som-based document clustering using phrases," in *ICONIP*, 2002.

[26] C. Hung and S. Wermter, "Neural network based document clustering using wordnet ontologies," *Int. J. Hybrid Intell. Syst.*, 2004.

[27] Y. Liu, X. Wang, and C. Wu, "Consom: A conceptional self-organizing map model for text clustering," *Neurocomput.*, 2008.
[28] T. Kohonen, *Self-Organizing Maps*. Springer, 2001.
[29] P. Koikkalainen and E. Oja, "Self-organizing hierarchical feature maps," in *Proceedings of International Joint Conference on Neural Networks*, vol. 2, 1990, pp. 279–285.
[30] M. Dittenbach, D. Merkl, and A. Rauber, "The growing hierarchical self-organizing map," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2000)*, vol. 6, 2000, pp. 15–19.
[31] J. Brank, M. Grobelnik, and D. Mladenic, "A survey of ontology evaluation techniques," in *Proceedings of the Conference on Data Mining and Data Warehouses (SiKDD 2005)*, 2005.
[32] K. Dellschaft and S. Staab, "On how to perform a gold standard based evaluation of ontology learning," *Lecture Notes in Computer Science*, vol. 4273, pp. 228–241, 2006.
[33] Y. Wilks and C. Brewster, "Natural language processing as a foundation of the semantic web," *Foundations and Trends in Web Science*, vol. 1, no. 3–4, pp. 199–327, 2006.
[34] C. Brewster, J. Iria, Z. Zhang, F. Ciravegna, L. Guthrie, and Y. Wilks, "Dynamic iterative ontology learning," *Recent Advances in Natural Language Processing (RANLP 07)*, 2007.
[35] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *Proceedings of MT Summit 2005*, 2005.