Automated Taxonomy Expansion and Tag Recommendation in a Knowledge Management System

Author:
M.J.A. van Duren

Universiteit Utrecht

InfoSupport
Solid Innovator

Automated Taxonomy Expansion and Tag Recommendation in a Knowledge Management System

Master of Science Thesis, Computing Science

Author:
M.J.A. van Duren
ICA-3470687

Universiteit Utrecht

Utrecht University, Faculty of Science
Department of Information and Computing Sciences

Info Support B.V.

November 2014
INF/SCR-14-3470687

Examiners:
Dr. A.J. Feelders
Prof. Dr. A.P.J.M. Siebes

Abstract

We investigate two problems in order to improve a knowledge management system. The first problem is insert newly created tags, used to classify documents, to their semantically correct position in a taxonomy. To solve this previously unexplored problem we try several techniques including association rules, Bayesian network learning and a custom approach. The accuracy of tag insertion was 23.3% on realistic scenarios and 71.0% on adapted scenarios. This score leads us to conclude that this approach is not practically applicable. Data set analysis gives a good insight in why this score is low and gives motivation for further research. The second problem is tag recommendation. Our custom approach finds words that have most mutual information with the tag and selects these words to train a classifier for each tag in order to make a recommendation. The classifiers used are naive Bayes and support vector machines. The best setting has a micro average F_1 score of 0.197. This score leads us to conclude that this approach is not practically applicable.

Contents

Chapter 1

Introduction

1.1 Motivation

For itself and some clients, Info Support is developing a knowledge management system. This system is characterized by its ability to answer questions like: “Who has knowledge about mobile development?”, “What knowledge does our company have of IT architecture with banks?”, “Which one of my colleagues has similar knowledge as I do?”.

Users can share and search content. Users can also rate content and the system will recommend content which is interesting to the user. In this process the system involves the user’s preferences, profile and self-learning and self-adaptation of the system.

Using the user ratings, the system builds a reputation about content. With these data a relevance score can be determined and adjusted. This concept can be compared to web search engines, they also rate pages and create a ranking based on search criteria, user feedback and user behaviour analysis.

For this knowledge management system, Info Support seeks algorithms for searching, rating and ranking. In the current situation, the user is offered the most relevant knowledge based on his profile, the process being executed, optional search criteria and rated content. Important requirements for this problem’s solution are correctness, limited search time and self-learning of the algorithms.

Info Support expects a theoretical framework with a design for a prototype and experiment set-up. With these, the chosen strategy can be tested. With the results, further improvements or alternatives can be developed and tested. Next to existing and proven algorithms and techniques, new ideas and concepts are welcome.

1.2 Background

Content in the system is tagged by users to describe it. For example, tags for this document could be “research”, “master thesis” and “graduation project”. These tags are also used in the user profiles to indicate their skills and interests. These tags are the most important input for the search algorithms to find similar users, content matching the search criteria and for recommending content the user may be interested in.

These tags are stored in a tree and form a taxonomy. A tag in the taxonomy can have an arbitrary number of children but always exactly one parent. Exceptions to these rules are the root node (which has no parent) and the static tag “Uncategorised” (which also has no parent). The taxonomy represents is-a relations between the tags. At the top levels of the tree are the most general concepts, concepts at lower levels are more specific. Also a tag’s parent is a more general concept than the tag itself. For example: “Motor vehicle” is more general than “Car” and should therefore be the parent of “Car”. We denote this relation as “Motor vehicle” → “Car”. The idea behind this taxonomy and the is-a relation is that we can use this information when searching content. For example, suppose the taxonomy is “Vehicle” → “Motor vehicle” → { “Car”, “Truck” }. When content is assigned with the tag “Car”, it is also implicitly assigned with all tags on the path to the root (in this case “Motor vehicle” and “Vehicle”).

When searching for content tagged with “Motor vehicle” we will also find that document because of the implicitly assigned tags. Also some document assigned with the tag “Truck” will be found, again because of the implicit assignment. When we would search for the tag “Car” a document tagged with “Truck” will be ranked lower in the result set. This is because the tags “Car” and “Truck” are siblings in the taxonomy and therefore do not match. However, they have a common parent (and ancestors), “Motor vehicle” (and “Vehicle”), and therefore do have some similarity.

In ranking the search results exact matches are most relevant, after that content that is more specific (tags lower in the taxonomy) and lastly the content that is more general (tags higher in the taxonomy).

In the current situation, most of the system is self-learning and requires little administration by people. Based on the user ratings, content reputations are automatically adjusted and also the user profiles change because of that. This affects the search results, ranking and recommendations.

Maintaining the taxonomy requires administration by people and is sometimes a hard and time-consuming job. Therefore Info Support would like to improve or even fully automate this process.

Next to this problem of automated taxonomy expansion, Info Support would like to improve the user experience by incorporating tag recommendation. When a user creates a new document, he must assign at least one tag which describes the document. By providing tag recommendations, the user no longer needs to search the taxonomy to find tags that describe the document.

1.3 Goals

It is important to insert a new tag “X” into the taxonomy at the correct place to benefit from the is-a or is-part-of relations for this tag when searching documents. When a new tag “X” is introduced, it is inserted into the taxonomy as a child of the static tag “Uncategorised” (which has no parent). When this tag “X” is used to search documents, the concept of using ancestors and descendants cannot be used since the tag “X” has no descendants and only one parent, “Uncategorised”. This will obviously limit the search algorithm’s potential of finding matching or relevant documents.

Therefore it is important that “X” is inserted into the taxonomy at the correct place. In the current situation this is done manually by people. Because this is a time-consuming process, Info Support would like to automate this process. The best position of this tag "X" in the taxonomy will have to be determined, based on the usage of the tag "X" (assignment to documents). All possible positions of the new tag "X" in the taxonomy can be ranked from best to worst. Based on this ranking, the tag can either be inserted at the best position automatically (based on some conditions) or the top k positions in this ranking can be presented to the user. The user may then select the best position and insert the tag there.

This would pose additional challenges but may also overcome limitations the taxonomy has. For example, suppose we would like to capture the relations between "Person", "Sports fan" and "Student". The is-a relations "Person" → { "Sports fan", "Student" } hold but "Sports fan" and "Student" are not mutually exclusive but neither one more specific than the other.

The main focus of this research will be to solve the problem of inserting a new tag "X" into the taxonomy at the correct position. Additional focus is to investigate how tag recommendation for new documents can yield recommending tags the user would have tagged himself. Therefore we pose the research questions:

- Which techniques or algorithms can be used to insert a new tag into an existing taxonomy based on the assignment of this tag to documents?

  The most important question to answer is which techniques or algorithms can solve this problem and which of these is best. A number of approaches are possible, so first of all we need to analyse the system and the data to be able to determine which techniques or algorithms can be used. Next, the goal is to maximize the number of correct insertions of new tags into the taxonomy.

- Which techniques or algorithms can be used to handle multi-label classification for tag recommendation and how can the input features for the models be selected?

  The second question to answer is how we can handle multi-label classification since a document can be tagged with any number of tags. This is a more challenging problem than single-label classification. Another challenge is to select the features that will be the input to train the classification models. In literature, some common approaches are investigated and the question is whether these will work well on the data and whether other approaches may yield better results.

The next questions will have to be answered for each technique or algorithm:

- Is the taxonomy semantically correct after insertion of a new tag?

  An important property of the taxonomy is that, in the current situation, it is maintained by people. Therefore we assume it to be correct. It is important that the relations in the taxonomy are intuitive to people and correct (see Definition 7 and Assumption 1). For example that "Car" and "Train" are more specific than "Transportation" and not the other way around. To measure and verify this aspect, the taxonomy has to be checked manually after an insertion. Next to this approach, the current taxonomy (with the assumption that it is correct) can be used by removing a tag from the taxonomy and then run the technique or algorithm on this tag. If it inserts the tag at the same position it was in then we can consider it correct, otherwise incorrect.

- What is the accuracy of the technique or algorithm?

  To be able to select the best technique or algorithm we need a way to assess its accuracy. Important to consider is that we would like the taxonomy to be semantically correct and to maximize the number of correct insertions of new tags into the taxonomy.

- What is the running time and scalability of the technique or algorithm?

  To determine accuracy and performance, several aspects can be measured. Most important are correct insertions. Next, running time and scalability are important. Based on the running time in O-notation more can be said about scalability.

- Under what conditions does the technique or algorithm work well?

  In these applications, the actual data plays a crucial role. Finding out in what conditions the technique or algorithm works well will provide more insight in how it should be used to ensure optimal accuracy and performance. This aspect can be experimented with by analysing certain aspects of the data set. These include the number of assigned tags per document and the number of co-occurring tag assignments.

1.4 Approach and requirements

There are several requirements the techniques and algorithms must meet to be practically applicable.

1.4.1 Domain independence

The system is deployed at Info Support and several clients, which all run an independent instance. Therefore, the system's data set strongly depends on the company and users the system instance is used by. Info Support is an IT company and the system's documents will mostly consist of technical IT-related articles and text documents. Clients in other branches will use the system for other (types of) documents. Therefore we require our techniques and algorithms to be domain independent. This implies that external knowledge (like the Wikipedia categorization [23] or Microsoft's ProBase [22]) cannot be used. Concepts in these taxonomies are very general and try to capture common or even world-wide knowledge. For some companies the system's documents may be very specific to the company, for example when describing company policies, processes and procedures. Obviously, for this very specific domain these external knowledge sources are useless.

1.4.2 Data set restrictions

Partly because of the domain independence, we prefer to only use the taxonomy and the tag sets assigned to documents but not the content. That is, next to the taxonomy we only require some way to uniquely identify a document and the set of tags explicitly assigned to it. Implicit assignment of tags (see Definition 5) can be determined using the explicitly assigned tags (see Definition 4) and the taxonomy.

This requirement is also stated because any type of document can be handled, from texts in any language to presentation slides to videos. This makes the technique or algorithm independent of the document type and thus relies on the user's ability to describe the documents by tagging.

However, using full text search may improve results. The text of a document will add information which may help to solve the problem. Note that this may not work on any data set since documents not always contain text. Documents always have a title which may also be used in the text search.

1.4.3 Running time

If the technique or algorithm is to be used online, a requirement is to provide the user with a result within 2 seconds. New tags are introduced only once in a while so if the algorithm is run off-line this constraint can be relaxed quite a bit. Imagine scheduling to run the algorithm or preprocessing at a time when the system is not used. In that case, the running time may be extended up to several hours.

Analogously for tag recommendation, the online part of the technique or algorithm has to provide the user with results within 2 seconds. Training these classifiers (or off-line parts or preprocessing) may take up to several hours.

1.5 Definitions

Definition 1 (Tag). *A* tag *is a word or several words defining an (abstract) concept or (actual) thing, describing the content of a document.*

Definition 2 (Document). *A* document *consists of some (type of) content and a title. This document may contain text but also images, videos, presentation slides, schemes, links, etcetera. The document contains the knowledge being shared. Each document is described by at least one tag.*

Definition 3 (Taxonomy). *A* taxonomy *is a tree where nodes are tags and arcs represent the relations. We denote "parent" → "child" which represent a "child" is-a "parent" relation or a "child" is-part-of "parent" relation. Each node has exactly one parent, except for the root node and the static node with tag "Uncategorised", which both have no parent. Each node can have any number of children.*

Figure 1.1 shows an example taxonomy.

Documents have tags assigned to them but the distinction between explicit and implicit assignment is important.

Definition 4 (Explicitly assigned tags). Explicitly assigned tags *are tags assigned to documents. These tags describe and classify the document.*

Definition 5 (Implicitly assigned tags). Implicitly assigned tags *are all tags that are ancestors of explicitly assigned tags in the taxonomy. These tags are more general concepts*

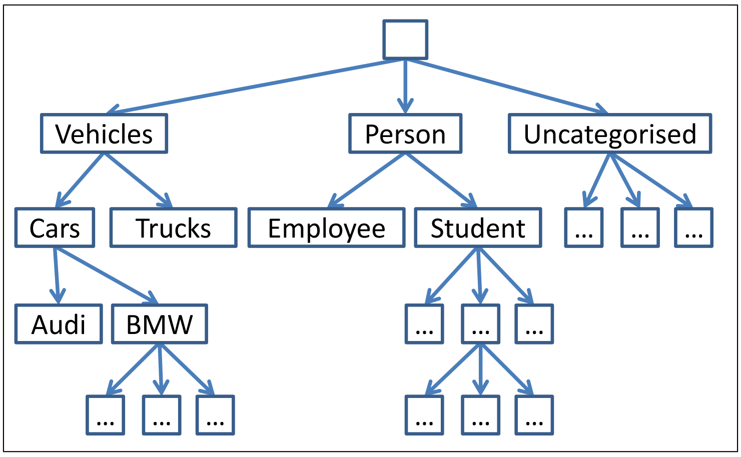

Figure 1.1: Example of a taxonomy.

than the explicitly assigned tags and therefore also describe and classify the content. These implicitly assigned tags are not actually assigned to the document, but have to be derived using the taxonomy and the explicitly assigned tags.

A document is explicitly assigned at least one tag. This implies that the document may have any number of implicitly assigned tags. This implicit assignment is not stored with the document but can de derived from the explicitly assigned tags and the taxonomy. In the search algorithms, these implicitly assigned tags may be used for matching.

Implicitly assigned tags are not actually assigned to the document because the taxonomy may change. For example, when moving a tag which is a child of "Uncategorised" to some other position in the taxonomy. From this point on, the set of implicitly assigned tags has changed and this information can be used in the search algorithms.

Note that explicitly assigned tags may be parent/child or ancestor/descendant of one another. Users are free to explicitly assign any tag they like. This implies that the set of implicitly assigned tags may intersect the set of explicitly assigned tags.

Definition 6 (Uncategorised tags)**.** Uncategorised tags *are tags whose parent in the taxonomy is the static tag "Uncategorised". These tags are allowed to be assigned to documents.*

Definition 7 (Semantically correct)**.** *A tag's position in the taxonomy is* semantically correct *if the relation "new tag" is-a "parent" or "new tag" is-part-of "parent" holds.*

An example of a semantically correct situation would be that "Car" is a child of "Vehicle" since it is true that a car is a vehicle. An example of a semantically incorrect situation would be that "Student" is a child of "Language" since it is false that a student is a language.

Note that semantic correctness cannot be checked automatically but is based on human judgement.

Also note that for any tag there may be multiple semantically correct situations. For example, both "Vehicle" → "Car" → "Sports car" and "Vehicle" → { "Car", "Sports car" } are semantically correct, even thought we prefer the first.

Definition 8 (Accuracy)**.** *The* accuracy *of a technique or algorithm is its ability to insert new tags into the taxonomy at a semantically correct position.*

1.6 Assumptions

Next to these definitions we also make an assumption.

Assumption 1. *The* existing taxonomy *in the system is assumed to be semantically correct. This means that for all pairs of parent/child tags in the taxonomy, the expressed is-a relation or is-part-of relation actually holds.*

The existing taxonomy is constructed and maintained by people with domain knowledge about the tags involved. This implies that all is-a and is-part-of relationships hold in the domain. Therefore we assume that the existing taxonomy is correct (Definition 7) to be able to automatically measure the algorithm's accuracy (Definition 8) by using the taxonomy and this assumption.

Note that other situations may also be semantically correct but this assumption makes it possible to assess accuracy automatically. However, this will result in a crude accuracy assessment.

1.7 Requirements practical applicability

The goal of this project is to develop techniques and algorithms to solve the two problems in a practical application. To be able to decide whether the developed approach is practically applicable we define two accuracy thresholds.

Requirement 1 (Accuracy threshold automated taxonomy expansion)**.** *For the problem of automated taxonomy expansion we define an accuracy threshold of* 80%*. An approach with higher accuracy is considered practically applicable.*

Requirement 2 (Accuracy threshold tag recommendation)**.** *For the problem of tag recommendation we define an accuracy threshold of* 60% *correct recommendations. An approach with higher accuracy is considered practically applicable.*

1.8 Overview

This master thesis is structured as follows. In chapter 2 we compare our work with previous work done in the area. In chapter 3 all techniques and algorithms for automated taxonomy expansion are explained. In chapter 4 all techniques and algorithms for tag recommendation are discussed. Next, in chapter 5, the data set is described and analysed. In chapter 6 further details of the experimental set-up are described. In chapter 7 the experiment results are shown and discussed. In chapter 8 we conclude this thesis and provide suggestions for future work.

Chapter 2

Previous work

No literature was found about research that has been done about the exact problem of taxonomy expansion we are investigating. There is previous work to be found about some related problems.

Related research includes taxonomy construction. However, most research is about constructing an entire taxonomy instead of extending an existing taxonomy. Sujatha and Bandaru [6] build a taxonomy of the tags using several techniques. Another approach is to analyse the document texts and check for term co-occurrence like Kim and Lee [8] describe. This may be an interesting technique for discovering relations, but using text will not always work on the data set we use in this project. Important in all taxonomy construction techniques is the similarity measure. Overviews are provided by Resnik [16] and Tan et al. [37] [38]. For binary similarity and distance measures, Choi et al. [24] provide a survey.

Another related topic we consider in this project is tag recommendation. Van Leeuwen and Puspitaningrum [29] use associations in collaborative tagging systems. In this project, there is no collaborative tagging but we do use association rules. Tuarob et al. [34] investigated the tag recommendation problem for a controlled tag library, similar to this project. Difference is that it is applied to environmental science metadata, which is different to the data set we use. In that data set, each document consists of text of which the title, abstract and description are preprocessed and then used as document context. In our data set this information is not available. Song et al. [35] investigated a fast, real-time technique for tag recommendation. However, it uses the text in documents, which will not always work for the data set we use in this project because we allow documents of any type of content, for example images, videos, presentation slides, etcetera. Song et al. [36] use a tag hierarchy to derive new tag recommendations.

Techniques we try in this project are association rules and association rule mining. Interesting literature includes Agrawal et al. [15] which is about mining association rules from large databases. Srikant and Agrawal [20] present generalised association rule mining. For multiple-level association rules involving a hierarchy, Han and Fu [21] developed several algorithms. The book by Sullivan [18] includes chapters about association rules and Bayesian statistics. Chapters in Larose [17] cover association rules and clustering techniques. More general books about data mining include Rajaraman et al. [1] and Hastie et al. [11], which include topics like frequent item sets, A-Priori, clustering, recommendation, association rules and graphical models. All of which gave inspiration for finding techniques to solve the two problems in this project.

Another field of interest in research is using the text of a document itself for concept extraction and taxonomy construction. Ontology-based feature extraction from text resources is discussed by Vicient et al. [7]. It is interesting to automatically construct an ontology from text, but in this project we investigate the problem of extending an existing taxonomy. Liu et al. [3] construct a taxonomy from keywords using a Bayesian approach. Compared to this project we already have a taxonomy, so expansion is more relevant than construction. Still, we might use ideas from these papers to improve our techniques and obtain higher accuracy for documents containing text.

A Bayesian approach to hierarchical classification and using a taxonomy is investigated by Cesa-Bianchi et al. [28]. Interesting is that it learns a hierarchical classifier, which could be useful for tag recommendation and taxonomy expansion in this project.

In some research, external knowledge sources are used for taxonomy construction. For example, Medelyan et al. [4] first extract concepts from text and link them to several external knowledge sources. Picca and Popescu [10] also use external knowledge sources, but adopt super sense tagging as a preprocessing step. Super sense tagging is a natural language processing task that consists of annotating each entity in a text like nouns, verbs, etcetera with a general semantic taxonomy defined by the WordNet lexicographer classes. For this project, no external knowledge sources can be used since the domain of the data may be so specific that there is no source that contains this knowledge.

Another big topic in late research involves social tagging, semantic web and collective tagging. Balby Marinho et al. [33] present recommender systems for social tagging systems. Heymann and Garcia-Molina [14] propose an algorithm to construct a taxonomy for a collaborative tagging system. What makes these collaborative tagging systems different to the system in this project, is that all users can tag any document. In the system for this project, only the author can assign tags to a document. This makes the data sets fundamentally different and therefore requires different approaches. Aliakbary et al. [9] investigate an approach to classify web pages involving social tagging. Although the research is based on web page classification, the proposed framework provides inspiration for this project.

Other interesting research is by Yao et al. [12] which is about evolutionary taxonomy construction. Although this is beyond the scope of this project, it may be an interesting approach to apply to the system we investigate in this project.

Chapter 3

Techniques for automated taxonomy expansion

The major part of this research involved investigating automated taxonomy expansion. In this chapter, all techniques, algorithms and custom approaches we experimented with are discussed.

3.1 Association rules

To introduce the formal definition of association rules we looked at the definitions given by Agrawal et al. [15] and Srikant and Agrawal [20]. However, we state the definitions specific to this project. Amongst other differences, this means we have documents instead of transactions and tags instead of items.

Let $\mathcal{T}$ be the set of tags in the taxonomy and let $\mathcal{D}$ be the set of documents. We define the relation $A(t, d)$ where $t \in \mathcal{T}$ and $d \in \mathcal{D}$, which denotes that the tag t is assigned to document d. Let the relation I be equal to A, except that it denotes assignment of only the set of implicitly assigned tags and let E denote only assignment of the set of explicitly assigned tags.

Now $\mathcal{D}_t = \{d \in \mathcal{D} | A(t, d)\}$ denotes documents to which the tag t has been assigned. Let $\mathcal{D}_X = \cap_{t \in X} \mathcal{D}_t$ denote all documents to which all tags in the set X have been assigned.

An *association rule* is defined as the implication of $X \Rightarrow t$ where X is a subset of tags in $\mathcal{T}$ $(X \subset \mathcal{T})$ and t a single tag in $\mathcal{T}$ and not in X $(t \in \mathcal{T} \wedge t \notin X)$. We call X the antecedent and t the consequent of the association rule.

Support is formally defined as:

$$\text{support}(X \Rightarrow t) = \frac{|\mathcal{D}_{X \cup \{t\}}|}{|\mathcal{D}|} \tag{3.1}$$

The support of the association rule $X \Rightarrow t$ is defined as the ratio of documents in $\mathcal{D}$ which have all tags in X and t assigned to them.

Confidence is formally defined as (also see the equations in Table 3.2):

$$\text{confidence}(X \Rightarrow t) = \frac{|\mathcal{D}_{X \cup \{t\}}|}{|\mathcal{D}_X|} \tag{3.2}$$

The confidence of the association rule $X \Rightarrow t$ is defined as the ratio of documents in $\mathcal{D}$ which have all tags in X and t assigned to them over the number of documents which have all tags in X assigned to them.

Support is an important measure because a rule which has low support may occur by chance. If so, the rule is likely to be uninteresting. Confidence is also an important measure because it measures the reliability of the inference made by the rule. The higher the confidence for rule $X \Rightarrow t$, the more likely it is for tag t to be assigned to documents the tags in X are assigned to.

3.2 Association rule mining

In general, association rules are mined from the data set. A naive approach would be to generate and evaluate all possible association rules. However, finding all subsets of $\mathcal{T}$ already takes $O(2^{|\mathcal{T}|})$ time since $|\mathcal{P}(\mathcal{T})| = 2^{|\mathcal{T}|}$. Therefore this approach is not feasible for a realistic number of tags.

To overcome this issue we can prune rules from the set of all possible association rules. Rules which are not interesting anyway will not be computed. The Apriori algorithm [39], for example, sets a support threshold and prunes all rules which have a support value lower than the threshold.

In this project, we are interested in finding the correct parent in the taxonomy for a new tag (as described in section 1.3). Therefore, association rules of the form $t \Rightarrow x$ are interesting where x is a new tag currently not in the taxonomy and t some tag currently in the taxonomy. Since only these rules are of interest, the number of possible association rules is exactly $|\mathcal{T}|$ (the number of tags) since potentially each tag can be the correct parent. The association rules to test are formally defined as:

$$\{(t \Rightarrow x) \mid t \in \mathcal{T}\}$$

All association rules generated have to be evaluated with an objective measure. Based on these values, a ranking can be made from most likely parent of x to least likely parent of x by sorting on these values. The tag “parent” in the new relation “parent” $\rightarrow x$ is determined by selecting the tag which is ranked highest.

A rule $t \Rightarrow x$ with high quality is an indication that t is a correct parent of x. Because of the implicitly assigned tags and the taxonomy, any time a tag x is assigned to a document, it can also be assigned with the parent of x because that parent is a more general concept than x. Since documents tagged with x can also be tagged with its parent tag, we expect that a high quality rule $t \Rightarrow x$ indicates that t is a correct parent of x.

3.3 Objective measures

For ranking the association rules, several symmetric and asymmetric objective measures are calculated. These are taken from Tan et al. [19]. Equation 3.1 defines the support. Table 3.1 displays a 2 way contingency table, this notation is used in the following equations. Table 3.2 lists the basic objective measures used in association rule mining. In Appendix A, Table A.1 lists the symmetric objective measures calculated and Table A.2 lists the asymmetric objective measures calculated.

Because the data set is very sparse we suspect that objective measures which include the number of true negatives will not perform well since the majority of cases will be a true negative. Because the is-a and is-part-of relations are asymmetric, the asymmetric objective measures may perform better than the symmetric objective measures.

		Actual		
		$A(t,d)$	$\neg A(t,d)$	
Predicted	$A(t,d)$	tp ($true\ positive$)	fp ($false\ positive$)	ppv ($positive\ predictive\ value$) $= tp + fp$
	$\neg A(t,d)$	fn ($false\ negative$)	tn ($true\ negative$)	npv ($negative\ predictive\ value$) $= fn + tn$
		tpr ($true\ positive\ rate$) $= tp + fn$	tnr ($true\ negative\ rate$) $= fp + tn$	$\lvert\mathcal{D}\rvert = tp + fp + fn + tn$

Table 3.1: A 2-way contingency table for predicting tag assignment.

Measure	Range	Definition
Support	(0...1)	$\mathrm{s}(A \Rightarrow B) = \frac{\lvert\mathcal{D}_{A \cup B}\rvert}{\lvert\mathcal{D}\rvert}$
Confidence	(0...1)	$\mathrm{c}(A \Rightarrow B) = \frac{\lvert\mathcal{D}_{A \cup B}\rvert}{\lvert\mathcal{D}_A\rvert}$
Lift	$(0...\lvert\mathcal{D}\rvert)$	$\mathrm{l}(A \Rightarrow B) = \frac{c(A \Rightarrow B)}{\lvert\mathcal{D}_B\rvert \div \lvert\mathcal{D}\rvert}$

Table 3.2: Basic objective measures used in association rule mining.

3.4 Bayesian networks

Next to using association rules to find the correct parent for a new tag x, we consider using Bayesian networks [25]. This choice is justified because using the implicitly assigned tags means that when a document is tagged with tag t, we expect the document to be also (implicitly) tagged with all ancestors of t. This implies that we expect the probability of the document being tagged with the ancestors of t, given that it is tagged with t, is 1. Formally, for any document d:

$$\forall\ x \in ancestors(t) : \Pr(A(x,d) \mid E(t,d)) = 1$$

This property is not used in association rules. Since a Bayesian network is a graph and not a ranking of $|\mathcal{T}|$ items like in association rules, we suspect a Bayesian network can express more information than the association rules can. Since the Bayesian network is more complex than association rules we might be able to draw better conclusions. In the association rules we only look at the interaction of the inserted tag to all other tags. In the Bayesian network all pairwise interactions are considered.

So we construct the Bayesian network from data and use the structure to derive the most likely parent for a new tag x.

3.5 Bayesian network learning

To construct the Bayesian network from data we use two methods. The first method is the Chow-Liu algorithm [40] [27] [26] and the second method the ARACNE algorithm [41]. Both are implemented in R [42] in the package *bnlearn* [43] [44].

Both techniques calculate the mutual information between each pair of nodes. For discrete variables T_1 and T_2 the concrete formula is:

$$I(T_1, T_2) = \sum_{t_1 \in \{0,1\}} \sum_{t_2 \in \{0,1\}} P(t_1, t_2) \log \frac{P(t_1, t_2)}{P(t_1)P(t_2)} \tag{3.3}$$

To estimate $I(T_1, T_2)$ from data we compute:

$$I(T_1, T_2) = \sum_{t_1 \in \{0,1\}} \sum_{t_2 \in \{0,1\}} \hat{P}(t_1, t_2) \log \frac{\hat{P}(t_1, t_2)}{\hat{P}(t_1)\hat{P}(t_2)} \tag{3.4}$$

Where T_1 and T_2 are tags. For actual values for T_1 and T_2 the probabilities are computed by counting the number of documents which match the tag assignment for t_1 and t_2.

3.5.1 Chow-Liu

The Chow-Liu algorithm first constructs a Bayesian network from data. It does so by computing the mutual information between each pair of nodes. Then it computes and returns the maximum spanning tree of the network. We find the most likely parent by finding the new tag x in this tree and see which other tags it is connected to via an edge. Since this tree is a maximum spanning tree of the constructed network, one of the connected tags should be the most likely parent. Any tag in this tree is connected to at least 1 and at most $|\mathcal{T}| - 1$ tags. On average it is connected to just a few tags.

Figure 3.1 shows an example of a Chow-Liu tree computed on the example taxonomy. In this example, the tag "Corvette" is to be inserted. All edges have a weight, calculated by the mutual information between the pair of nodes. The edges in the maximum spanning tree are coloured blue. In this example, we would conclude that "Corvette" is-a or is-part-of "Cars".

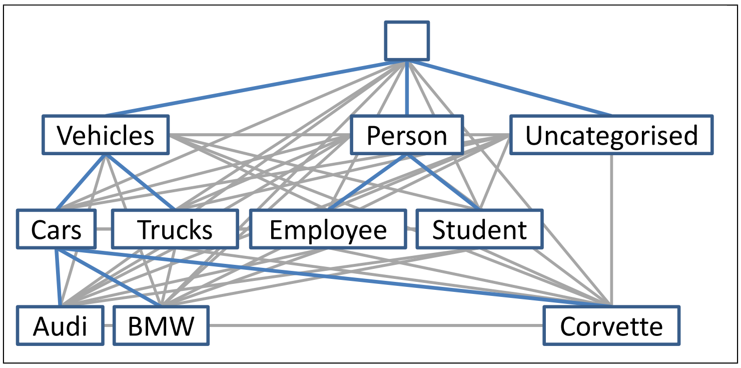

Figure 3.1: Example of a network and Chow-Lui tree (edge weights omitted).

3.5.2 ARACNE

The ARACNE algorithm (Califano et al. [41]) also first calculates the Bayesian network from data. Like Chow-Liu, it does so by computing the mutual information between each pair of nodes. Then it checks all triplets of nodes which are fully connected in this network (there is an edge between each pair of these three nodes) and of each triplet it deletes the edge with the smallest weight. Thus, all triplets have to be checked and the edge with smallest weight is removed if it exists in the triplet. As opposed to Chow-Liu, which selects edges to include, ARACNE selects edges to remove and returns the remaining edges. In general, this results in a graph which has more edges than a Chow-Liu tree but never less since Chow-Liu returns the maximum spanning tree of the same initial network.

3.6 Custom approaches

The Chow-Liu and ARACNE algorithms gave inspiration for a custom approach. In these algorithms, mutual information is used to calculate the statistical dependency between two variables. These are then used as weights in the Bayesian network from which the final tree or graph is computed. With all the objective measure values calculated for the association rules there are many more choices for setting the edge weights. Therefore, our custom approach is to select an objective measure and construct a fully connected graph in which the edge weights are the chosen objective measure values calculated for the association rules. On this graph the final tree or graph can be computed just like the Chow-Liu or ARACNE algorithm does. This way, the usage of different objective measures and graph construction methods (Chow-Liu and ARACNE) can be compared to one another.

3.6.1 Kruskal

For computing the tree from the Bayesian network like Chow-Liu does, we need to compute the maximum spanning tree from a graph. After constructing the graph we run Kruskal's algorithm [45] to compute the maximum spanning tree.

3.6.2 ARACNE's graph construction

For computing the graph like in the ARACNE algorithm, we can simply iterate over all triplets of nodes which are fully connected. Of each triplet we remove the edge with the smallest weight but only if it is strictly smaller than the other two edge weights. After checking all fully connected triplets of nodes, the final graph has been computed.

3.6.3 Custom framework using full text search

This custom approach uses a full text search to improve results. A text is not available in all documents, but whenever present it will at least add information which will most likely help to solve the problem. Note that this approach may not work well on every data set. The document may be an article with lots of text, but it may just as well be a URL or video, which means a full text search can only use the document's title.

First we pre-process all documents and tags. To make the text search return more matches, we strip all text of markup encoding, punctuation and line breaks. Finally, we convert all letters to lower case. We use the regular expressions in Equations 3.5, 3.6 and 3.7 and replace all matches with a blank space.

$$\text{''} < [\hat{}\ >]* > \text{''} \tag{3.5}$$

$$\text{''}\&[\hat{};]*;\text{''} \tag{3.6}$$

$$\text{''}[.,!?()/\backslash @\backslash - :;'\text{''}\backslash[\backslash]\&\backslash\backslash]\text{''} \tag{3.7}$$

Next, for each document we take all the words and see if these words are a tag. Some tags consist of several words, so we also check for the concatenation of several subsequent words. Important is to analyse the tags found this way and compare them to the set of explicitly assigned tags. At least this will provide more insight in the data set.

Another technique we expect will result in more text matches is to use a word stemmer. These stemmers are language dependent and the Info Support data set contains texts mostly in Dutch but also in English. We use the original Porter stemmer [30], Snowball English, Snowball Dutch and both Snowball English and Dutch [46]. For more information about the Dutch stemmer see Kraaij and Pohlmann [31] and [32].

Now that we have the documents with explicitly assigned tags and tags found in the full text search, perform the following two steps. First, make all tags found in the full text search explicitly assigned. Second, for each tag now explicitly assigned to the document, find all implicitly assigned tags and make them explicitly assigned. With the tags assigned to documents this way we then calculate the association rule objective measures.

With these objective measure values we use the same approach as described earlier to compare accuracy results. That is, using these values as input for constructing a Bayesian network and using Kruskal's algorithm or ARACNE's graph construction.

We suspect this method yields higher accuracy since more information is used. That is, we also use information contained in the document's title and text in addition to the assigned tags. Analysis of the data set will provide more insight in this hypothesis and provides opportunities to further improve this approach.

3.6.4 Assumption validation

In preliminary experiments, the accuracy was low. Therefore we test the assumption that the existing taxonomy is semantically correct (Assumption 1). To do this we apply the two set-ups (first, using both explicitly and implicitly assigned tags and second, both explicitly and implicitly assigned tags plus the full text search) on a trimmed version of the taxonomy. The taxonomy contains branches which are known to not follow the definition of the is-a or is-part-of relationship. For example, there is a top level tag "Terms" which contains a lot of leaf nodes. Thus, a human expert trimmed the taxonomy of all these branches, resulting in a taxonomy of 213 tags of which the expert is convinced that it is semantically correct and used correctly in the system. Of the 213 tags, 160 are explicitly tagged to at least one document in the Info Support data set.

3.6.5 Analysis

To gain more insight in the data set we need to analyse it. But to test hypotheses about it, which may be used to create better models, we need to perform some experiments. The first hypothesis to test is that siblings in the taxonomy are mutually exclusive. The second test is to find out whether users do or do not assign a tag's parent to a document in addition to that tag.

Chapter 4

Techniques for tag recommendation

When the user creates a document, he must assign at least one tag to it. To ease this task, tag recommendation can be used. While the user is creating the document, the system can suggest tags to assign to the document. All the user has to do is to assign these tags instead of searching tags in the possibly large and complex taxonomy.

So the task of tag recommendation is actually predicting which tags should be assigned to a document.

Because each document has at least one but usually more tags explicitly assigned to it, we are dealing with recommendation for multi-label classification. Since we want to be able to make recommendations for each individual tag we need to find a way to handle this.

4.1 Multi-label classification

The problem of multi-label classification is harder to solve than single-label classification. There are several approaches to convert the multi-label classification problem to a single-label classification problem [58]. Converting the data set with the label powerset approach is not suited for this data set. Since the Info Support data set has more tags $(2,478)$ than documents (947), the powerset of assigned tags explodes. Transforming the data set using the instance copy approach is also less suitable because then we have to handle multi-class classification. Instead, we transform using binary relevance.

With binary relevance, we virtually copy the data set for each tag in the taxonomy. For each copy (for one tag) we change the class of the document, only recording whether or not that tag is assigned to that document. This way each data set copy is a single-label, two-class classification problem.

Since a classifier is trained for each tag we suspect that the accuracy of these models will be higher than the other approaches. For each of these classifiers the data set has exactly the same size before and after this transformation.

The disadvantage of binary relevance is that associations between tags are not used. These may improve accuracy because they add information. In chaining predictions, for example, this information is used. When chaining predictions, we first predict the first tag. Then for the second tag, we include the prediction for the first tag as input. For the third tag we include the prediction for the first and second tag as input. And so on until we reach the last tag, for which the prediction of all other tags is included as input.

4.2 Baseline

Since the Info Support data set is very sparse, we set a baseline to compare other techniques and algorithms to. To do so we evaluated the data set in four set-ups. The four set-ups are obtained by choosing two settings. The first setting is to either use the trimmed taxonomy or use the full taxonomy. The second setting is to either use the explicitly assigned tags only or use the explicitly and implicitly assigned tags. Each document has at least one tag explicitly assigned to it, but since the trimmed taxonomy has only 8.6% of the tags compared to the full taxonomy, there are a lot of documents with no tags at all in these set-ups.

To determine the baseline for the set-ups, each classifier predicts a single value. This value is determined by predicting the most occurring value for that tag. For most of the set-ups using the Info Support data set this means that for most tags, we predict that the document is not classified with that tag. Since the data set is very sparse, a large fraction of the predictions will be true negatives and some false negatives.

4.3 Feature selection

For selecting the words to use as input for the models we cannot select all words. After the preprocessing there are over $12,000$ distinct words, which would cause the training to take too long to be applicable in a practical situation. There are also words that provide no additional information about whether the tag should be assigned to the document or not. Therefore we need to limit the number of words used as input. This can be done by counting the number of documents the word occurs in and trim words which occur in less than (for example) 10% of the documents. Another option is to calculate the Term Frequency - Inverse Document Frequency (TF-IDF) [2] and select the top k most important words.

In this research, we try a method that will hopefully yield better results. For each tag the mutual information between the words and that tag is calculated and the top k words selected as input features. The idea is that the words selected have most information in common with the tag and thus are best to predict the assignment of the tag.

$$I(T,W) = \sum_{t\in\{0,1\}} \sum_{w\in\{0,1\}} \hat{P}(t,w) \log \frac{\hat{P}(t,w)}{\hat{P}(t)\hat{P}(w)} \tag{4.1}$$

Calculating the mutual information is done like in Equations 3.3 and 3.4. Except that T_1 and T_2 are no longer tags but we replace them with T and W where T are tags and W are words, see Equation 4.1. Now $\hat{P}(t,w) = \frac{N_{tw}}{|\mathcal{D}|}$, $\hat{P}(t) = \frac{N_t}{|\mathcal{D}|}$ and $\hat{P}(w) = \frac{N_w}{|\mathcal{D}|}$. For actual values for T and W, we compute N_{tw}, N_t and N_w by counting the number of documents which have the tag assigned to it and the word occurring in that document.

The most important question we want to answer with these experiments is how many words are needed to obtain a reasonable accuracy. Therefore we test the number of words set at 100, 200, 400, 800 and 1600.

4.4 Naive Bayes

For training the classifiers, we first use a naive Bayes classifier as implemented in the R [42] package *e1071* [59]. For each tag we select the input features and train the model on the training set. This classifier's accuracy is then assessed using the test set.

The input features for these models are binary variables, denoting whether a word occurs in a document or not. We run the experiments both with and without Laplace smoothing [2].

4.5 Support vector machines

For training the classifiers we also use a support vector machine classifier as implemented in the R [42] package *e1071* [59]. For each tag we select the input features and train the model on the training set. The classifier's accuracy is then assessed using the test set.

The input features for these models are binary variables, denoting whether a word occurs in a document or not. This way accuracy can be compared to the naive Bayes classifiers accuracy. The experiments are run with several kernel functions and cost values. The first experiment is with the standard linear kernel. The second experiment is with the radial kernel function and cost set at 1. The third experiment is the radial kernel but with cost set at 100.

4.6 Cross validation

Because the Info Support data set is so sparse, splitting the data set into a training set and test set at random may result in unfortunate splits. This may severely impact the accuracy results. Therefore, cross validation is performed to obtain more reliable results. Thus the data set is split into 4 equal partitions. For each experiment, we evaluate 4 times, taking one partition as test set and the other three together as training set each time. The results are averaged over all folds to obtain the final result. To make sure the experiments are repeatable we set a constant feed to the random number generator, which is used to partition the dataset at random.

4.7 Accuracy assessment

It is important to select the proper measure for assessing accuracy. It is especially important in these experiments because the Info Support data set is very sparse. There are a lot of true negatives, so taking these into account will yield a high accuracy for the baseline. Therefore measures involving the true positives, false positives and false negatives will be more informative.

4.7.1 Accuracy and classification accuracy

Classification accuracy is defined as the ratio of documents over all documents for which the set of predicted tags is equal to the set of actually assigned tags. This measure is very strict since the prediction must be correct for every tag.

Accuracy is defined as the ratio of correct predictions over all predictions (see Formula 4.2). This means the number of true positives and true negatives over the total number of predictions. Considering that there will be a lot of true negatives we expect the accuracy to be high. See Table 3.1 for a two-way contingency table.

$$Accuracy = \frac{tp + tn}{tp + fp + fn + tn} \tag{4.2}$$

4.7.2 Precision, recall, F_1

Inspired by the field of information retrieval, we calculate precision, recall and F_1.

$$Precision = \frac{tp}{tp + fp} \tag{4.3}$$

$$Recall = \frac{tp}{tp + fn} \tag{4.4}$$

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{4.5}$$

As explained above, we do not take the number of true negatives into account because this number is several orders of magnitudes larger than the rest. Goal is to maximize the number of true positives and minimize the number of false positives and false negatives. This is exactly what precision and recall express. F_1 combines these two (it is the harmonic mean of precision and recall).

4.7.3 Micro average and macro average

Since there is one classifier for each tag, we need a way to aggregate the accuracy measures. Classification accuracy is a measure over all tags and documents, so no additional aggregation is required. Accuracy is aggregated over all tags, summing all true positives, all true negatives, all false positives and all false negatives before calculating accuracy using Equation 4.2.

For precision, recall and F_1, micro averaging and macro averaging is applied. For micro averaging, all terms in Formulas 4.3 and 4.4 are summed over all tags. See Formulas 4.6 and 4.7, here each sigma operator sums over all tags.

$$Micro\ average\ precision = \frac{\sum tp}{\sum tp + \sum fp} \tag{4.6}$$

$$Micro\ average\ recall = \frac{\sum tp}{\sum tp + \sum fn} \tag{4.7}$$

For macro averaging, the values of Formulas 4.3 and 4.4 are averaged over all tags. See Formulas 4.8 and 4.9, here each sigma operator sums over all tags.

$$Macro\ average\ precision = \frac{\sum precision}{|\mathcal{T}|} \tag{4.8}$$

$$Macro\ average\ recall = \frac{\sum recall}{|\mathcal{T}|} \tag{4.9}$$

Chapter 5

Data set

In data mining applications the data plays an important role. Certain properties or characteristics of the data may make or break the application. We already set the requirement that our approach should be domain independent. So it should not only work on one particular data set, but also on data sets covering other domains. In the following section we discuss and analyse the data set used in this project.

Data sets of the knowledge management system instances running at clients of Info Support contain dozens to about one hundred documents and tags. This makes these data sets too small to obtain reliable results from the experiments.

5.1 Info Support

The most important data set is the one of the knowledge management system instance running at Info Support. The knowledge management system started about two years ago and has been growing ever since. In February 2014, the data set contained 947 documents and 2478 tags of which 189 tags are currently uncategorised. In total, there are 5160 explicitly assigned tags, which means an average of 5.45 explicitly assigned tags per document.

Figure 5.1 shows the number of tags per level (distance to the root) in the taxonomy. This is much like what one would expect from a taxonomy. Few tags at the first level and gradually increasing as the level increases. At some point, the number decreases again because the taxonomy reaches more and more leaf nodes.

Figure 5.2 shows the number of explicitly assigned tags to a document, per level of the tag in the taxonomy. As expected, this chart follows the trend in the taxonomy, depicted in Figure 5.1. It is actually good to see that users tend to use tags at deeper levels in the taxonomy, this implies there are more implicitly assigned tags. More implicitly assigned tags means the document can be found with more search terms and this also expresses more information.

Figure 5.3 shows the number of documents, with a certain number of explicitly assigned tags. This chart clearly shows that documents with over 11 explicitly assigned tags are rare. The user is required to explicitly assign at least one tag to the document, but is free to assign more. Again, the trend seen in the chart is beneficial to the search process. The more tags a document is explicitly tagged with, the easier it will be found and the more information is expressed by the assigned tags.

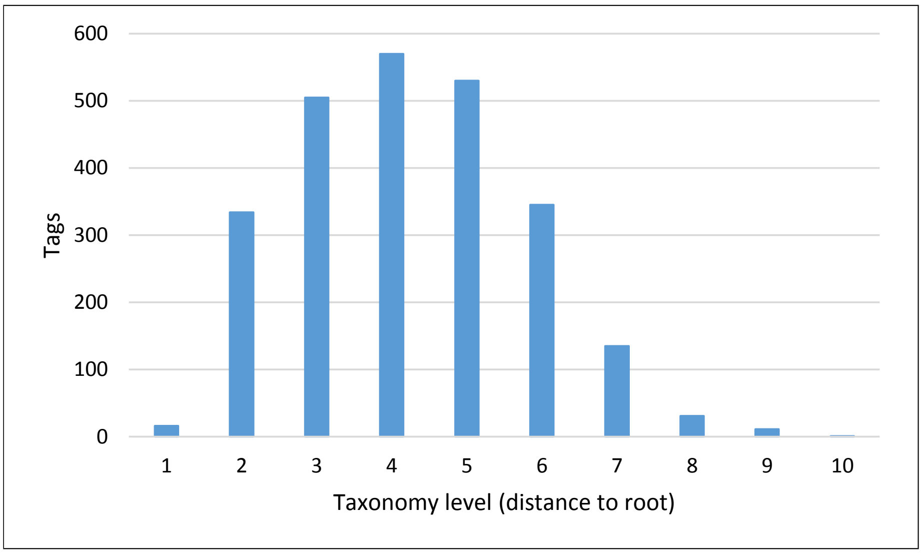

Figure 5.1: Number of tags per level in the taxonomy.

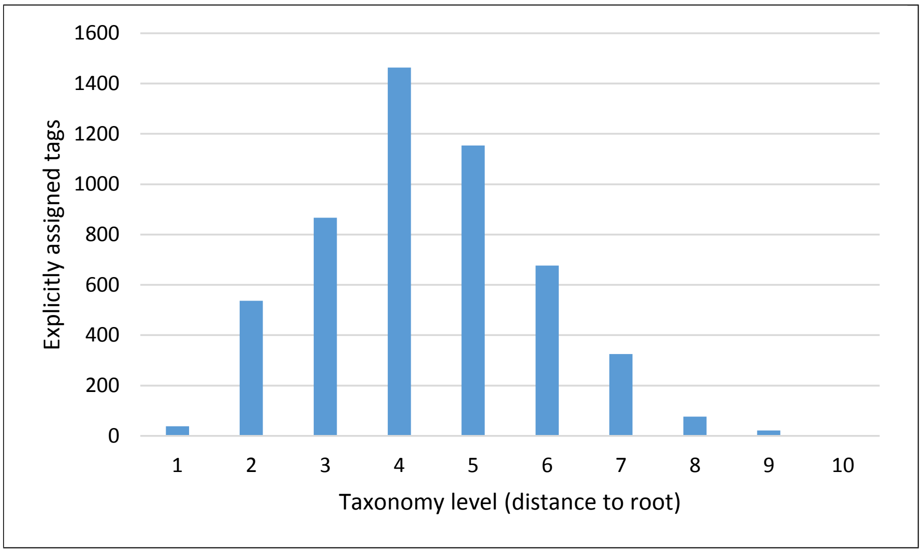

Figure 5.2: Number of explicitly assigned tags per level in the taxonomy.

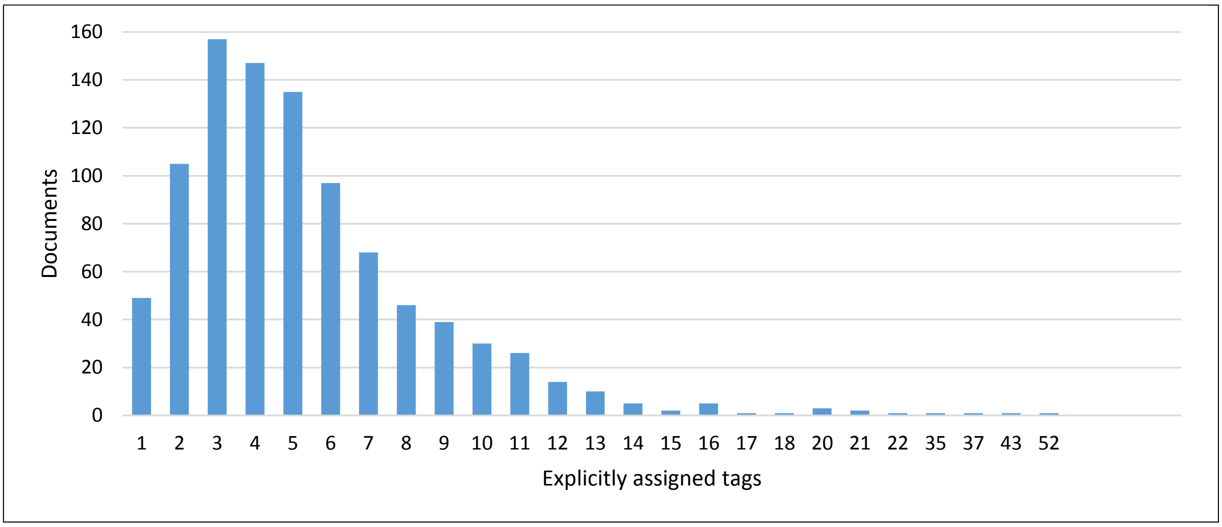

Figure 5.3: Number of documents grouped by the number of explicitly assigned tags to a document.

Chapter 6

Experiments

An experimental set-up is needed to assess the accuracy (Definition 8) and assess the algorithm's ability to keep the taxonomy semantically correct (Definition 7) after inserting a new tag. All experiments are run using the Info Support data set.

To make sure the semantically correct requirement holds, we take the existing taxonomy and use that for evaluation. Each tag in the taxonomy, except children of "Uncategorised", is taken out of the taxonomy. Then the algorithm is run and we check whether the tag is inserted at its original position. If so, the insertion was semantically correct, otherwise incorrect. We can draw this conclusion because we made the assumption that the existing taxonomy is semantically correct (Assumption 1). Note that this may not be the only correct position, thus this is a crude accuracy measure, but using this assumption enables automated accuracy assessment.

To assess accuracy we check whether the actual parent (using the existing taxonomy) is in the neighbourhood (connected by an arc in the tree or graph) of the tag to be inserted. We count all these cases and divide this by the total number of insertions.

The neighbourhood of the new tag are all tags with relatively high objective measure values. So in a practical application, it seems reasonable to recommend these tags as most likely parents to the user and let him make the final decision of where to actually insert the new tag.

6.1 Analysis

In preliminary experiments, we analysed the data set to test our theories. The assessed accuracy when making a ranking using the association rules alone was low. Therefore we tested the theory that siblings in the taxonomy are mutually exclusive. Siblings are more specific than some common tag, therefore they should be mutually exclusive. This way the models might be extended to make better rankings.

To analyse the theory that siblings are mutually exclusive, we count the number of times that a tag and one of its siblings are explicitly assigned to a document.

In Table 6.1 we can see that the theory that siblings are mutually exclusive seems to hold in the Info Support data set. Drawback is that extending the model with this theory will provide no or only little improvement. These counts are very small (if not 0), so overall they do not change the rankings.

Number of co-occurrences	7	6	5	4	3	2	1
Number of tags	3	0	5	17	38	116	562

Table 6.1: Co-occurrences of siblings in the taxonomy.

A possible explanation as to why the accuracy is low is that documents tagged with a new tag are not also tagged with a more general concept of that new tag. For example, suppose this document is tagged with "Master thesis", most likely a user will not also tag this document with "Thesis" since it is obvious to the user that a Master thesis is a thesis.

So we analysed the data set to count the number of times it occurs that a document tagged with a tag, is also tagged with its parent in the taxonomy.

The highest number of co-occurrences is 26 and only 5 tags co-occur at least 10 times with their parent. For at least 5 co-occurrences this is 35 tags and 735 tags co-occur with their parent at least once. The tags with most co-occurrences in the Info Support data set are tags about just a few topics and concepts very popular at Info Support. Lots of documents are about these topics and are tagged a lot with these tags and their parents.

We suspect the approaches described in the next section will perform better since our association rules approach only considers pairwise tag interaction and the others construct a graph which may contain more information.

6.2 Algorithms

We implemented the techniques and algorithms described in Chapters 3 and 4.

For automated taxonomy expansion these are:

- Chow-Liu (R package *bnlearn*)
- ARACNE (R package *bnlearn*)
- Association rules with Kruskal (custom C#)
- Association rules with ARACNE's graph construction (custom C#)

For tag recommendation these are:

- Naive Bayes (R package *e1071*)
- Support vector machines (R package *e1071*)

The association rule objective measures were calculated for all pairwise combinations of tags. This resulted in $2478 \cdot 2477 = 6,138,006$ association rules. These values were used as input for the last two algorithms. Constructing the initial graph takes $O(n+m)$ time or equivalently $O(m^2)$ time[1]. Running Kruskal's algorithm takes $O(n\ log\ n)$ time

[1] The initial graph is fully connected.

or equivalently $O(n\ log\ m)$ time and ARACNE's graph construction $O(m^3)$ time worst case[2], which is infeasible on a graph with 6 million edges. Therefore we limited the initial graph to only include those edges which have the objective measure value defined and a value not 0. This reduces the number of edges to a little over $30,000$.

Including edges with weight 0 would not make sense because it means the tags have no information in common at all. Second, in the ARACNE graph construction, most of these edges would be kept in the graph, resulting in lots of very large strongly connected components (the data set is sparse) but they would have no meaning. Also, the exact order of computation would play a role to which exact edges are included in the final graph and which are not. So for some tags this would mean that being connected to their actual parent (in the existing taxonomy) would be a mere chance instead of the tags actually having some mutual information.

We limited the experiments to using the trimmed taxonomy due to time constraints. This means that training 2478 classifiers is reduced to training 160 classifiers.

[2] For each triplet where an edge is deleted, this reduces the number of triplets by m^2, thus (depending on the edge weights) the expected running time is less than $O(m^3)$.

Chapter 7

Results

This chapter shows the results per approach, as described in previous chapters. Note that in the results we also distinguish between using explicitly assigned tags only and including the implicitly assigned tags and using the full taxonomy or the trimmed taxonomy.

7.1 Automated taxonomy expansion

7.1.1 Chow-Liu

As we can see in Table 7.1, the accuracy of using the explicitly assigned tags only is low and increases when including the implicitly assigned tags. However, it is still low. This is what one would expect since the implicitly assigned tags add a lot of information.

7.1.2 ARACNE

Compared to Chow-Liu, the accuracy is better for ARACNE. These approaches use the same objective measure, the only difference is in the way the final tree or graph is constructed from the network. This also explains the difference in accuracy. Chow-Liu returns a tree, so on average each tag is connected to just a few other nodes. The probability that one of these is the semantically correct parent is small. In ARACNE, only some edges are deleted from the graph and the resulting graph is never smaller than the tree returned by Chow-Liu. Especially since a lot of edges have the same weight because the data set is sparse. On average, a tag in the ARACNE graph is connected to more tags than in the Chow-Liu tree. Therefore the probability of the semantically correct parent being in the neighbourhood is larger.

Despite the increase in accuracy, this approach may not be more suitable. Since the neighbourhood is larger than in Chow-Liu (over 20 tags is not uncommon), recommending these tags to the user as most likely parents will make the task of choosing the correct parent harder for the user.

7.1.3 Association rules with Kruskal

All objective measures we tested perform better than the standard Chow-Liu algorithm. However, the highest accuracy is only 0.051 which is still low.

Approach	**Taxonomy mode**	**Objective measure**	**Tags checked**	**Parents in neighbourhood**	**Accuracy**
Chow-Liu	Explicit tags only	Mutual information	2431	82	0.034
	Implicit tags included			566	0.233
ARACNE	Explicit tags only	Mutual information	2431	1527	0.628
	Implicit tags included			1727	0.710
Association rules + Kruskal	Explicit tags only	Confidence	1674	59	0.035
		Support		86	0.051
		Lift		63	0.038
		Jaccard		85	0.051
		IS		79	0.047
		Certainty factor		62	0.037
		Added value		71	0.042
Association rules + ARACNE graph	Explicit tags only	Confidence	1674	76	0.045
		Support		306	0.183
		Lift		208	0.124
		Jaccard		186	0.111
		IS		189	0.113
		Certainty factor		73	0.044
		Added value		73	0.044

Table 7.1: Accuracy results for taxonomy expansion.

7.1.4 Association rules with ARACNE's graph construction

Accuracy of all objective measures is lower than the standard ARACNE algorithm. But, this is due to the fact that only edges with weight larger than 0 are included. Thus the final graph has way less edges, implying the accuracy drops. As expected, the accuracy is higher than the Kruskal approaches, with a maximum accuracy of 0.183 when using support as objective measure.

7.1.5 Custom framework using full text search

For the custom full text search, we first analysed the occurrence of tag's parents in the full text search. 75.23% of the explicitly assigned tags are also found in the full text search. 27.79% of the tag's parent is also found in the full text search and 59.21% of the tag's ancestor is also found in the full text search. These numbers give reason to suspect that using the tags found in the full text search may improve accuracy results.

Analysing the other way around provides insight in the possibilities for recommending tags to the user while creating a document. 30.21% of tags found in the full text search are also explicitly assigned. Only 6.22% of the tag's parent is explicitly assigned and 11.83% of the tags found have their ancestor explicitly assigned. 23.78% of the tag's descendant is also explicitly assigned and only 15.11% of the tag's siblings are explicitly assigned.

Applying any stemmer decreased accuracy. The number of tags matched dropped by about 87%. One would expect the number of matches to be at least the same or higher, since the stem of equal words is also equal. This certainly gives reason for further research to find out why this is the case.

Table 7.2 and Figure 7.1 show the accuracy results.

7.1.6 Assumption validation

For testing the assumption that the existing taxonomy is semantically correct we apply the association rules and Kruskal's algorithm on the trimmed taxonomy and all documents. The best objective measures (correlation and Piatetsky-Shapiro) resulted in an accuracy of 0.145. This is an improvement over the same approach applied on the full taxonomy.

When we also incorporate the full text search approach and use the trimmed taxonomy, accuracy increased to 0.566 on the best objective measure (Piatetsky-Shapiro). Given the difficulty of this problem, this is not a bad result. However, it is still lower than the threshold defined in Requirement 1, thus not practically applicable.

Even though accuracy increased on this trimmed taxonomy, it is tricky to draw conclusions about the assumption. Some branches are known to violate the definition but this trimmed taxonomy is small compared to the full taxonomy (160 tags versus 2478 tags).

Table 7.2 and Figure 7.1 also show the accuracy of the custom approaches.

Approach	**Taxonomy mode**	**Objective measure**	**Tags checked**	**Parents in neighbourhood**	**Accuracy**
Association rules + full text search + Kruskal	Implicit tags included	Confidence	1934	103	0.053
		Support		238	0.123
		Lift		112	0.058
		Jaccard		402	0.208
		IS		420	0.217
		Certainty factor		98	0.051
		Added value		142	0.073
		Interest		122	0.063
		Odds ratio		104	0.054
		Kappa		394	0.204
		Piatetsky-Shapiro		561	0.290
		Goodman-Kruskal		417	0.216
		J-Measure		493	0.255
Trimmed taxonomy + Kruskal	Implicit tags included	Confidence	110	13	0.118
		Support		10	0.091
		Lift		15	0.136
		Jaccard		15	0.136
		IS		15	0.136
		Certainty factor		15	0.136
		Added value		14	0.127
		Interest		15	0.136
		Correlation		16	0.145
		Odds ratio		15	0.136
		Kappa		15	0.136
		Piatetsky-Shapiro		16	0.145
		Goodman-Kruskal	108	11	0.102
		J-Measure	108	15	0.139
Trimmed taxonomy + full text search + Kruskal	Implicit tags included	Confidence	145	35	0.241
		Support		46	0.317
		Lift		18	0.124
		Jaccard		52	0.359
		IS		62	0.428
		Certainty factor		48	0.331
		Added value		52	0.359
		Interest		18	0.124
		Correlation		62	0.428
		Odds ratio		40	0.276
		Kappa		50	0.345
		Piatetsky-Shapiro		82	0.566
		Goodman-Kruskal		77	0.531
		J-Measure		67	0.462

Table 7.2: Accuracy results for taxonomy expansion.

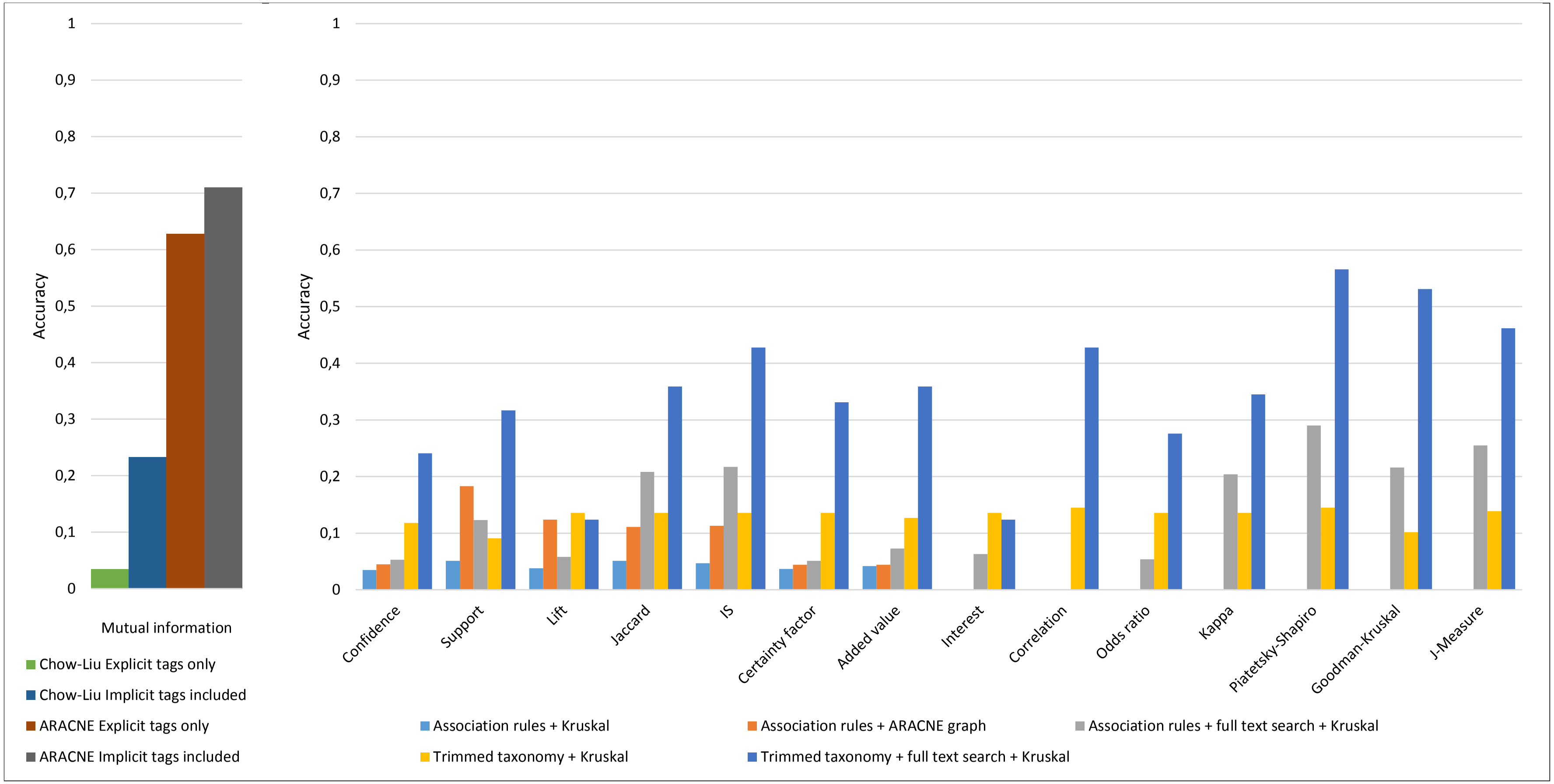

Figure 7.1: Accuracy results for taxonomy expansion.

7.2 Tag recommendation

From the baseline experiments we see that indeed the data set is very sparse. By predicting that no tag is assigned to any document, the accuracy is 0.994 to 0.998. On the trimmed taxonomy, the classification accuracy is 0.719. Therefore these measures are not suited for assessing accuracy.

Instead, micro average F_1 is better suited since it takes the numbers into account which are most interesting for tag recommendation: true positives, false positives and false negatives.

Independent of the number of words used as input features, the naive Bayes classifier yields low accuracy values. The support vector machines achieve higher values. Worth investigating though is that with the radial kernel and cost value 1, all scores are 0. As expected, values increase as more words are used as input features. At some point, we expect the increase in value to decrease, as adding more and more words will add less and less information to learn from compared to the information already used. The best micro average F_1 value is 0.197 for the support vector machine with linear kernel.

Figures 7.2, 7.3 and 7.4 show the trend in the micro average precision, recall and F_1 for various approaches.

Table 7.3 shows the accuracy for tag recommendation.

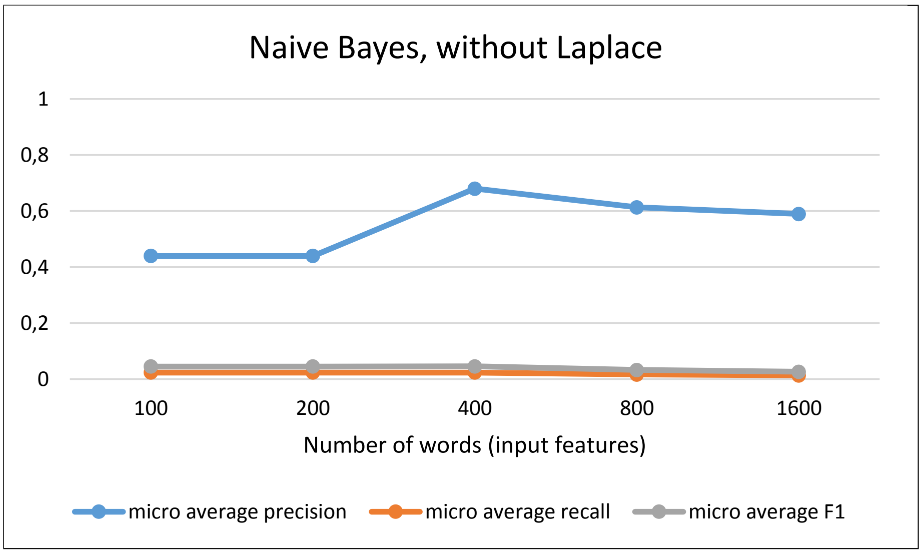

Figure 7.2: Tag recommendation using naive Bayes classifiers.

Words	Taxonomy	Taxonomy mode	Model	Classification Accuracy	Accuracy	Micro average			Macro average		
						Precision	Recall	F_1	Precision	Recall	F_1
100	Trimmed	Explicit tags only	Baseline	0.719	0.994	0	0	0	0	0	0
	Trimmed	Implicit tags included		0.719	0.997	0	0	0	0	0	0
	Full	Explicit tags only		0	0.994	0.049	0.766	0.093	0.000	0.000	0.000
	Full	Implicit tags included		0	0.998	0	0	0	0	0	0
100	Trimmed	Implicit tags included	Naive Bayes	0	0.892	0.440	0.024	0.045	0.087	0.004	0.007
			Naive Bayes, with Laplace	0	0.609	0.172	0.002	0.005	0.124	0.001	0.002
			Support vector machines, linear	0.712	0.994	0.004	0.3	0.007	0.000	0.002	0.000
			Support vector machines, radial	0.719	0.994	0	0	0	0	0	0
			Support vector machines, radial, cost=100	0.676	0.994	0.021	0.275	0.038	0.001	0.002	0.001
200	Trimmed	Implicit tags included	Naive Bayes	0	0.892	0.440	0.024	0.045	0.087	0.004	0.007
			Naive Bayes, with Laplace	0	0.598	0.299	0.004	0.008	0.136	0.002	0.004
			Support vector machines, linear	0.638	0.994	0.050	0.287	0.084	0.002	0.003	0.002
			Support vector machines, radial	0.719	0.994	0	0	0	0	0	0
			Support vector machines, radial, cost=100	0.644	0.994	0.049	0.291	0.083	0.002	0.004	0.002
400	Trimmed	Implicit tags included	Naive Bayes	0	0.837	0.680	0.024	0.046	0.133	0.005	0.009
			Naive Bayes, with Laplace	0	0.587	0.523	0.007	0.014	0.150	0.003	0.006
			Support vector machines, linear	0.560	0.993	0.094	0.254	0.136	0.004	0.004	0.004
			Support vector machines, radial	0.719	0.994	0	0	0	0	0	0
			Support vector machines, radial, cost=100	0.613	0.994	0.075	0.312	0.120	0.003	0.006	0.004
800	Trimmed	Implicit tags included	Naive Bayes	0	0.791	0.614	0.017	0.033	0.159	0.004	0.008
			Naive Bayes, with Laplace	0	0.598	0.338	0.005	0.010	0.136	0.002	0.004
			Support vector machines, linear	0.539	0.993	0.122	0.277	0.169	0.008	0.010	0.009
			Support vector machines, radial	0.719	0.994	0	0	0	0	0	0
			Support vector machines, radial, cost=100	0.647	0.994	0.089	0.415	0.147	0.005	0.010	0.007
1600	Trimmed	Implicit tags included	Naive Bayes	0	0.745	0.590	0.014	0.026	0.196	0.004	0.009
			Naive Bayes, with Laplace	0	0.614	0.145	0.002	0.004	0.117	0.001	0.002
			Support vector machines, linear	0.579	0.993	0.141	0.339	0.197	0.012	0.019	0.015
			Support vector machines, radial	0.719	0.994	0	0	0	0	0	0
			Support vector machines, radial, cost=100	0.668	0.994	0.092	0.542	0.156	0.005	0.014	0.007

Table 7.3: Accuracy results for tag recommendation.

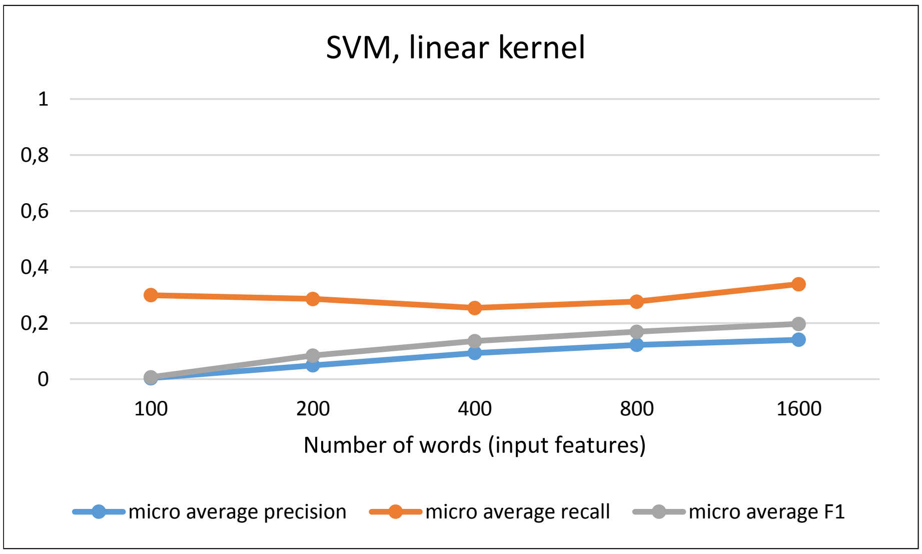

Figure 7.3: Tag recommendation using support vector machines.

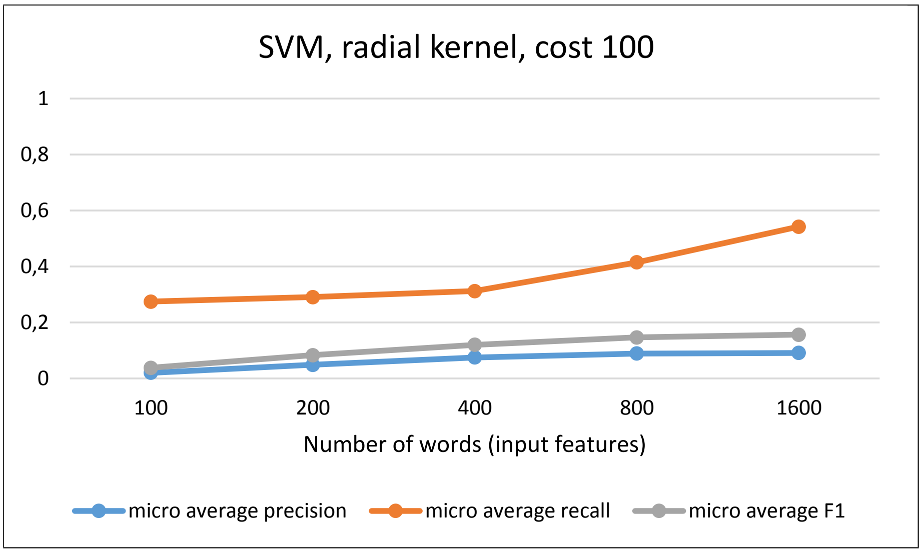

Figure 7.4: Tag recommendation using support vector machines.

Chapter 8

Conclusion

To conclude this thesis, we draw conclusions for the two problems investigated, provide a summary and discuss some future work.

8.1 Conclusions automated taxonomy expansion

Accuracy never exceeded 23.3% for any of the approaches except ARACNE and on the trimmed taxonomy. We do not consider ARACNE applicable in the practical application because a neighbourhood size of over 20 is not uncommon. Presenting this number of tags as most likely parents to the user is too much in a practical application. Since the accuracy scores are lower than the threshold we defined in Requirement 1, we do not consider these approaches suitable for a practical application.

A possible explanation as to why the approaches we tested do not work is because only in few cases a tag's parent is also explicitly assigned. Simply because users think the is-a or is-part-of relation we are trying to discover is trivial. So the principal question is how we can ever discover this relation if it is never recorded. Therefore we doubt if this exact problem is solvable with an acceptable accuracy on the Info Support data set.

To answer the research questions, the proposed approaches are not suitable for solving the problem of inserting new tags into the taxonomy based on the assignment of the new tag to documents for the Info Support data set. We suspect it will work well if a new tag's semantically correct parent is also explicitly assigned to documents the new tag is explicitly assigned to, but this requires further research.

Using the trimmed taxonomy instead of the full taxonomy improved the results. This trimmed taxonomy makes the data set less sparse which causes the algorithms to perform better. Trimming the taxonomy by assigning the parent tag when trimming a tag might improve results even more. The full text search also improved the results a lot, meaning the text adds a lot of helpful information. Not all documents have a large text, but at least a title which in itself might contain enough information to improve the accuracy.

8.2 Conclusions tag recommendation

For tag recommendation the accuracy results did not exceed the threshold defined in Requirement 2 and therefore are not practically applicable. The best micro average F_1 score is 0.197. We suspect this is also due to the sparsity of the Info Support data set. We suspect accuracy results will improve if the taxonomy is smaller. This will also result in tags being assigned to more documents. For the full taxonomy, lots of leaf nodes are assigned to just one or a few documents.

In theory, the more words are used to train the classifiers, the higher the accuracy scores get. This trend clearly shows in the support vector machine with radial kernel and cost value 100. This method also yielded the third best micro average F_1 value. Following the trend, using more words than tested in these experiments may yield higher accuracy values. Doing so might yield scores which are considered good enough to be practically applicable.

8.3 Summary

We investigated the problem of inserting newly created tags to their semantically correct parent in the taxonomy. The goal was to find an approach that can solve this problem automatically. We generalised the problem to be domain independent and use only the tag assignments to documents and the existing taxonomy. For trying to solve this previously unexplored problem we investigated several approaches. These include association rules, association rule mining and Bayesian network learning. Preliminary experiments showed that the accuracy of association rules was low, providing motivation to look further into other approaches like Bayesian networks and developing some custom approaches. The accuracies of all approaches investigated are below the threshold, not exceeding 23.3% in realistic and acceptable scenarios and not exceeding 71.0% when using ARACNE and the trimmed taxonomy. Data set analysis gives a good insight in why this may be the case and gives motivation for further research.

We also investigated the problem of tag recommendation. Focus was to select the best input features to train the classifiers on by calculating the mutual information between the words and tags. We limited the experiments to varying the number of words used to train the classifiers. The idea is that selecting words that co-occur most with a tag will be the best predictors for that tag. Results show that the best setting has a micro average F_1 value of 0.197. It is lower than the threshold and therefore not suitable for a practical application and gives motivation for further research.

8.4 Future work

The first issue this research lacks to address is to test the approaches on multiple data sets. The data sets of the knowledge management system instances running at clients of Info Support were not suitable because they are too small. The approach we would have labelled as suitable would be applied to these data sets in this system.

Other data sets could be used, like the Reuters corpus [47] and Wikipedia documents from the Large Scale Hierarchical Text Classification Challenge [61]. These have different characteristics compared to the Info Support data set when comparing sparsity, taxonomy size and document to tag ratios.

Time did not allow us to run the tag recommendation experiments on the full data set for this project. These experiments were only run on the trimmed taxonomy, which proved to yield better results in the experiments for automated taxonomy expansion.

Second, the suspicion that these approaches perform poorly because the tag's parents are not explicitly assigned to documents as well can be investigated further. In a practical application we could ask the user to add another tag he thinks is closest to the new tag when explicitly assigning the new tag to a document. Drawback is that the user has to perform more actions than strictly required for him to just add the document. On the other hand, some users enjoy the feeling of improving the system and will be eager to perform this optional task. Hopefully it will provide reliable information to find out which tag is the semantically correct parent of the new tag. Interesting question to answer would be to see how many of these extra tags we need to determine the semantically correct position of the new tag in the taxonomy. A technique worth investigating here is a statistical approach using the interquartile range to find the high outliers in the set of selected parent tags.

Third is to find out why accuracy drops when using any stemmer. This is a strange result since one would expect the number of matches to at least stay the same or rise instead of drop.

Fourth is to experiment with more models and settings on the tag recommendation problem. In this project we limited to experimenting with a naive Bayes classifier and support vector machines. Both have several parameters that can be tuned. The same holds for the data set pre-processing. Because of the way naive Bayes classifiers work, we chose to just indicate whether a word occurs in a document or not. Other measures, like term frequency or TF-IDF, might be able to select features that yield higher accuracy for predicting a tag. There are also many more models which may be suitable, for example k Nearest Neighbours and neural networks.

Fifth is to reconsider the accuracy measures used. Because the data set is sparse, the accuracy was 99% for the tag recommendation when doing no predictions at all. Therefore we chose to use accuracy measures that do not involve the number of true negatives. But besides this is the principle that we train the models on other criteria than the accuracy measures we assess on. The question is whether this will ever yield good results if the model is trained with different measures or criteria. Worth investigating is whether other measures or criteria in learning the models or other accuracy assessment measures yield better results.

Alternatively, the accuracy measure could be chosen to be a measure based on ranking. Suppose we rank the possible parents and look at the position in this ranking where the actual parent or first match occurs. We considered using a measure like this in the experiments but this introduced extra issues and considerations. For example, how to handle the ranking for tags with equal sort values and what value to rank on. Some models may provide excellent measures for ranking.

Another possibility is to look at the position in the taxonomy where the tag is inserted by the algorithm (based on the best prediction). Knowing where the tag should be positioned, we could come up with some distance measure between these two position in the taxonomy. The goal would then be to minimize this distance.

The same could be applied to the tag recommendation problem. We prefer to predict tags that are as specific as possible but recommending more general tags are not incorrect predictions.

Another idea for a research project is to extensively test the hypothesis that the is-a or is-part-of relation between tags can be discovered (using the tested methods) when users do assign the parent of a new tag explicitly. It would be interesting to know whether the is-a or is-part-of relation will be discovered and if so, what ratio of documents explicitly assigned with a new tag also need to have the semantically correct parent explicitly assigned to it to be able to discover the relation.

Inspired by Vogrinčič and Bosnić [58] we chose a method to transform multi-label classification to single-label classification. There are more ways to deal with multi-label classification like label powerset, instance copy and chaining recommendation. It would be worth investigating these methods to find out which is suitable in what situation and why.

As suggested by Info Support, it may be worth investigating how the taxonomy can be expressed as a graph instead of a tree. This way we can allow more and other relations than just the is-a and is-part-of relation. It will not so much help the problems we investigated in this project, but it might improve the search algorithms. With more relations the search algorithm has more information to match documents, possibly yielding better matches and rankings. Another benefit of this approach is that the graph is more flexible and it can evolve with the data set, thus improving the system's ability to learn and adapt.

Last issue to investigate is whether the approaches tested in this project would work better if the data set were to have different characteristics. We suspect that because there are more tags than documents and thereby a sparse data set causes the approaches to have low accuracy results. Because of these characteristics in the current Info Support data set the suspicion rises that the tags describe the content too specific. A proposal to investigate is to shrink the taxonomy to a point where there are more documents than tags while maintaining the information expressed.

Acknowledgements

Writing a master thesis is not an easy task and would not have been possible without the help of others. Therefore I would like to thank a number of people.

First, dr. A.J. Feelders, assistant professor at Utrecht University, primary supervisor and first examiner. Thanks for all the support, help, ideas and guidance.

Second, prof. dr. A.P.J.M. Siebes, professor at Utrecht University, second examiner.

Third, the company Info Support B.V., for supporting this project. A special thanks to its employees involved: Joop Snijder (principal and daily supervisor), Marco Pil (technical supervisor), Michelle van der Zwan - van der Jagt and Pascalle Hijl (process supervisors) and Henk Brands (manager).

Thank you all for all the information, guidance, ideas, feedback, brainstorming, expertise and support throughout this project.

Last but certainly not least, my family and friends for their guidance, advice and support throughout my studies.

Appendix A

Association rules objective measures

Measure	Range	Definition
Interest factor	(0...N)	$I(A,B) = \frac{s(A\Rightarrow B)}{\lvert\mathcal{D}_A\rvert\cdot\lvert\mathcal{D}_B\rvert} = \frac{\lvert\mathcal{D}\rvert\cdot tp}{ppv\cdot tpr}$ ($= 1$) if antecedent and consequent are independent. (> 1) if A and B are positively correlated. (< 1) if A and B are negatively correlated.
Correlation	(-1...1)	$\phi = \frac{tp\cdot tn - fp\cdot fn}{\sqrt{ppv\cdot tpr\cdot npv\cdot tnr}} = \frac{\lvert\mathcal{D}\rvert\cdot tp - ppv\cdot tpr}{\sqrt{ppv\cdot tpr\cdot npv\cdot tnr}}$ ($= 0$) if statistically independent. (-1) if perfect negative correlation. ($+1$) if perfect positive correlation.
IS Measure	(0...1)	$IS(A,B) = \sqrt{I(A,B)\cdot s(A\Rightarrow B)} = \frac{s(A\Rightarrow B)}{\sqrt{\lvert\mathcal{D}_A\rvert\cdot\lvert\mathcal{D}_B\rvert}} = \frac{\vec{A}\bullet\vec{B}}{\lvert\vec{A}\rvert\cdot\lvert\vec{B}\rvert} = cosine(\vec{A},\vec{B})$
Odds ratio	(0...∞)	$\alpha = \frac{tp\cdot tn}{fp\cdot fn}$
Kappa	(-1...1)	$\kappa = \frac{\lvert\mathcal{D}\rvert\cdot tp + \lvert\mathcal{D}\rvert\cdot tn - ppv\cdot tpr - npv\cdot tnr}{\lvert\mathcal{D}\rvert^2 - ppv\cdot tpr - npv\cdot tnr}$
Piatetsky-Shapiro	(-0.25...0.25)	$PS = \frac{tp}{\lvert\mathcal{D}\rvert} - \frac{ppv\cdot tpr}{\lvert\mathcal{D}\rvert^2}$
Collective strength	(0...∞)	$S = \frac{tp+tn}{ppv\cdot tpr + npv\cdot tnr}\cdot\frac{\lvert\mathcal{D}\rvert - ppv\cdot tpr - npv\cdot tnr}{\lvert\mathcal{D}\rvert - tp - tn}$
Jaccard	(0...1)	J $= \frac{tp}{ppv+tpr-tp}$
All-confidence	(0...1)	$h = min[\frac{tp}{ppv}, \frac{tp}{tpr}]$

Table A.1: Symmetric objective measures.

Measure	Range	Definition
Goodman-Kruskal	(0...1)	$\lambda = \frac{\Sigma_j max_k f_{jk} - max_k f_{+k}}{\lvert\mathcal{D}\rvert - max_k f_{+k}}$
Mutual Information	(0...1)	$M = \frac{\Sigma_i \Sigma_j \frac{f_{ij}}{\lvert\mathcal{D}\rvert} \log \frac{\lvert\mathcal{D}\rvert \cdot f_{ij}}{f_{i+} f_{+j}}}{-\Sigma_i \frac{f_{i+}}{\lvert\mathcal{D}\rvert} \log \frac{f_{i+}}{\lvert\mathcal{D}\rvert}}$
	where: $f_{11} = tp,\ f_{10} = fp,\ f_{1+} = ppv,\ f_{01} = fn,\ f_{00} = tn,$ $f_{0+} = npv,\ f_{+1} = tpr,\ f_{+0} = tnr$	
J-Measure	(0...1)	$J = \frac{tp}{\lvert\mathcal{D}\rvert} \log \frac{\lvert\mathcal{D}\rvert \cdot tp}{ppv \cdot tpr} + \frac{fp}{\lvert\mathcal{D}\rvert} \log \frac{\lvert\mathcal{D}\rvert \cdot fp}{ppv \cdot tnr}$
Gini index	(0...1)	$G = \frac{ppv}{\lvert\mathcal{D}\rvert} \cdot ((\frac{tp}{ppv})^2 + (\frac{fp}{ppv})^2) - (\frac{tpr}{\lvert\mathcal{D}\rvert})^2 + \frac{npv}{\lvert\mathcal{D}\rvert} \cdot ((\frac{fn}{npv})^2 + (\frac{tn}{npv})^2) - (\frac{tnr}{\lvert\mathcal{D}\rvert})^2$
Laplace	(0...1)	$L = \frac{tp+1}{ppv+2}$
Conviction	(0...∞)	$V = \frac{ppv \cdot tnr}{\lvert\mathcal{D}\rvert \cdot fp}$
Certainty Factor	(−1...1)	$F = \frac{\frac{tp}{ppv} - \frac{tpr}{\lvert\mathcal{D}\rvert}}{1 - \frac{tpr}{\lvert\mathcal{D}\rvert}}$
Added Value	(−1...1)	$AV = \frac{tp}{ppv} - \frac{tpr}{\lvert\mathcal{D}\rvert}$

Table A.2: Asymmetric objective measures.

Bibliography

[1] Anand Rajaraman, Jure Leskovec, Jeffrey D. Ullman, *Mining of Massive Datasets.* Cambridge University Press, Cambridge, England, 2014.
`http://infolab.stanford.edu/~ullman/mmds.html`

[2] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, *An Introduction to Information Retrieval.* Cambridge University Press, Cambridge, England, 2009.
`http://www.informationretrieval.org/`

[3] Xueqing Liu, Yangqiu Song, Shixia Liu, Haixun Wang, *Automatic Taxonomy Construction from Keywords.* In: KDD'12, August 12-16, 2012, Beijing, China.
`http://research.microsoft.com/en-us/um/people/shliu/brt.pdf`

[4] Olena Medelyan, Steve Manion, Jeen Broekstra, Anna Divoli, Anna-Lan Huang, Ian H. Witten, *Constructing a Focused Taxonomy from a Document Collection.* Pingar Research, Auckland, New Zealand, University of Waikato, Hamilton, New Zealand.

[5] Rob Shearer, Ian Horrocks, Boris Motik, *Exploiting Partial Information in Taxonomy Construction.* Oxford University Computing Labratory, Oxfork, UK.

[6] R. Sujatha, Rama Krishna Rao Bandaru, *Taxonomy Construction Techniques - Issues and Challenges.* School of Information Technology and Engineering, VIT University. 2011. In Journal of Computer Science and Engineering (IJCSE), Vol. 2 No. 5 Oct-Nov 2011.

[7] Carlos Vicient, David Sánchez, Antonio Moreno, *An automatic approach for ontology-based feature extraction from heterogeneous textual resources.* Intelligent Technologies for Advanced Knowledge Acquisition (ITAKA), Departament d'Enginyeria Informática i Matemátiques, Universitat Rovira i Virgili. Tarragona, Catalonia, Spain, 2013. In: Engineering Applications of Artificial Intelligence 26 (2013) 1092-1106.

[8] Han-joon Kim, Sang-goo Lee, *Discovering Taxonomic Relationships from Textual Documents.* School of Computer Science en Engineering, Seoul National University, Seoul Korea.

[9] Sadegh Aliakbary, Hassan Abolhassani, Hossein Rahmani, Behrooz Nobakht, *Web Page Classification Using Social Tags.* Sharif University of Technology. In: CSE'09 Proceedings of the 2009 International Conference on Computational Science and Engineering.

[10] Davide Picca, Adrian Popescu, *Using wikipedia and supersense tagging for semi-automatic complex taxonomy construction.* University of Lausanne, Dorigny, Switserland, CEA LIST, Fontenay aux Roses, France, 2007.

[11] Trevor Hastie, Robert Tibshirani, Jerome Friedman, *The Elements of Statistical Learning (2nd ed.).* (10th printing, Jan 2013) Springer, New York, 2009.
`http://statweb.stanford.edu/~tibs/ElemStatLearn/`

[12] Junjie Yao, Bin Cui, Gao Cong, Yuxin Huang, *Evolutionary taxonomy construction from dynamic tag space.* Springer, 2011. In: World Wide Web (2012) 15:581-602, DOI 10.1007/s11280-011-0150-4.

[13] Xiang Li, Huaimin Wang, Gang Yin, Tao Wang, Cheng Yang, Yue Yu, Dengqing Tang, *Inducing Taxonomy from Tags: An Agglomerative Hierarchical Clustering Framework.* Springer, 2012. In: Advanced Data Mining and Applications LNCS, Volume 7713, 2012, pp 64-77.

[14] Paul Heymann, Hector Garcia-Molina, *Collaborative Creation of Communal Hierarchical Taxonomies in Social Tagging Systems.* Stanford University, Stanford, 2006.
`http://ilpubs.stanford.edu:8090/775/1/2006-10.pdf`

[15] Rakesh Agrawal, Tomasz Imielinski, Arun Swami, *Mining Association Rules between Sets of Items in large Databases.* In: Proceedings of the 1993 ACM SIGMOD Conference Washinton DC, USA. May 1993.
`http://www.almaden.ibm.com/cs/quest/papers/sigmod93.pdf`

[16] Philip Resnik, *Using Information Content to Evaluate Semantic Similarity in a Taxonomy.* 1995.

[17] Daniel T. Larose, *Discovering Knowledge in Data: An Introduction to Data Mining.* 2005.

[18] Rob Sullivan, *Introduction to Data Mining for the Life Sciences.* Springer, 2012.

[19] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, *Introduction to Data Mining.* Addison-Wesley, Pearson Education, 2006.
`http://www-users.cs.umn.edu/~kumar/dmbook/index.php`
`http://www.pearsonhighered.com/educator/academic/product/0,1144,`
`0321321367,00.html`

[20] Ramakrishnan Srikant, Rakesh Agrawal, *Mining Generalized Association Rules.* In: Proceedings of the 21st VLDB Conference, pages 407-419, Zurich, Switzerland, 1995.
`http://www.vldb.org/conf/1995/P407.PDF`

[21] Jiawei Han, Yongjian Fu, *Mining Multiple-Level Association Rules in Large Databases.* In: IEEE Trans. on Knowledge and Data Engineering, 11(5):798-804, 1999.

[22] Microsoft, *Probase.* Microsoft Research.
http://research.microsoft.com/en-us/projects/probase/

[23] Wikipedia, *Wikipedia Portal Categories.*
http://en.wikipedia.org/wiki/Portal:Contents/Categories

[24] Seung-Seok Choi, Sung-Hyuk Cha, Charles C. Tappert, *A Survey of Binary Similarity and Distance Measures.* Systemics, cybernetics and informatics, volume 8, number 1, 2010.
http://www.iiisci.org/journal/CV$/sci/pdfs/GS315JG.pdf

[25] Nir Friedman, Dan Geiger, Moises Goldszmidt, *Bayesian Network Classifiers.* In: Machine Learning, 29, 131-163. Kluwer Academic Publishers, 1997.
http://link.springer.com/article/10.1023%2FA%3A1007465528199

[26] Marina Meilă, *An accelerated Chow and Liu algorithm: fitting tree distributions to high-dimensional sparse data.* Massachusetts institute of Technology, Artificial Intelligence Laboratory, January 1999.
ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-1652.pdf

[27] Timo Koski, *Lectures on Statistical Learning Theory for Chow-Liu Trees.* The 32nd Finnish Summer School on probability Theory, 2010, Institutionen för matematik, Kungliga tekniska högskolan (KTH), Stockholm.
http://www.math.kth.se/~tjtkoski/chowliulect.pdf

[28] Nicoló Cesa-Bianchi, Alex Conconi, Clausio Gentile, *A Bayesian Framework for hierarchical Classification.*
http://eprints.pascal-network.org/archive/00000021/01/subm06.pdf

[29] Matthijs van Leeuwen, Diyah Puspitaningrum, *Improving Tag Recommendation using Few Associations.* Dept. of Information & Computing Sciences, Utrecht University, The Netherlands.

[30] M.F. Porter, *An algorithm for suffix stripping.* Computer Laboratory, Corn Exchange Street, Cambridge.

[31] Wessel Kraaij, Renée Pohlmann, *Porter's stemming algorithm for Dutch.*

[32] Wessel Kraaij, Renée Pohlmann, *Evaluations of a Dutch stemming algorithm.*
http://www.cs.ru.nl/~kraaijw/pubs/Keyword/papers/kraaij95evaluation.pdf

[33] Leandro Balby Marinho, Andreas Hotho, Robert Jäschke, Alexandros Nanopoulos, Steffen Randle, Lars Schmidt-Thieme, Gerd Stummer, Panagiotis Symeonidis, *Recommender Systems for Social Tagging Systems.* SpringerBriefs in Electrical and Computer Engineering, Springer, 2012.
http://www.springer.com/computer/database+management+%26+information+retrieval/book/978-1-4614-1893-1

[34] Suppawong Tuarob, Line C. Pouchard, C. Lee Giles, *Automatic Tag Recommendation for Metadata Annotation using Probabilistic Topic Modeling.* In: JCDL'13 proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries, pages 239-248, 2013.
`http://www.csm.ornl.gov/~7lp/publis/fp050-tuarob.pdf`

[35] Yang Song, Ziming Zhuang, Huajing Li, Qiankun Zhao, Jia Li, Wang-Chien Lee, C. Lee Giles, *Real-time Automatic Tag Recommendation.* SIGIR'08, July 20-24, 2008, Singapore.
`http://grads.ist.psu.edu/zzhuang/docs/sigir08_tagging.pdf`

[36] Yang Song, Baojun Qiu, Umer Farooq, *Hierarchical Tag Visualization and Application for Tag Recommendations.* CIKM'11, October 24-28, 2011, Glasgow, Scotland, UK.
`http://research.microsoft.com/pubs/151942/cikmfp0486-song.pdf`

[37] Pang-Ning Tan, Vipin Kumar, Jaideep Srivastava, *Selecting the Right Interestingness Measure for Association Patterns.* SIGKDD'02, Admonton, Alberta, Canada, 2002.
`http://www.dbis.informatik.hu-berlin.de/dbisold/lehre/WS0405/KDD/paper/TKS02.pdf`

[38] Pang-Ning Tan, Vipin Kumar, Jaideep Srivastava, *Selecting the Right Objective Measure for Association Analysis.*
`http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.331.4740&rep=rep1&type=pdf`
`http://www.cse.msu.edu/~ptan/papers/IS.pdf`

[39] Rakesh Agrawal, Ramakrishnan Srikant, *Fast Algorithms for Mining Association Rules.* In: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, pages 487-499, Santiago, Chile, September 1994.
`http://rakesh.agrawal-family.com/papers/vldb94apriori.pdf`

[40] C.K. Lee, C.N. Liu, *Approximating Discrete probability distributions with depences trees.* In: IEEE Transactions on Information Theory, IT-14 (3): 462-467, 1968.

[41] Adam A Margolin, Ilya Nemenman, Katia Basso, Chris Wiggins, Gustavo Stolovitzky, Riccardo Favera, Andrea Califano, *ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context.* BMC Bioformatics 2006, 7(Suppl 1):S7.
`http://www.biomedcentral.com/1471-2105/7/S1/S7`

[42] R Core Team, *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2014
`http://www.r-project.org/`

[43] Radhakrishnan Nagarajan, Marco Scutari, Sophie Lebre, *Bayesian Networks in R with Applications in Systems Biology.* Sprinker, New York, 2013. ISBN 978-1461464457.

http://www.springer.com/statistics/computational+statistics/book/978-1-4614-6445-7

[44] Marco Scutari, *Learning Bayesian Networks with the bnlearn R Package.* In: Journal of Statistical Software, 35(3), 1-22, 2010.
http://www.jstatsoft.org/v35/i03/

[45] Joseph B. Kruskal, Jr., *On the shortest spanning subtree of a graph and the traveling salesman problem.* In: Proceedings of the American mathematical Society 7: 48-50, 1956.
http://www.ams.org/journals/proc/1956-007-01/S0002-9939-1956-0078686-7/home.html

[46] *Snowball.*
http://snowball.tartarus.org/

[47] David D. Lewis, Yiming Yang, Tony G. Rose, Fan Li, *RCV1: A New benchmark Collection for Text Categorization Research.* In: Journal of machine Learning Research 5 (2004) 361-397.
http://www.jmlr.org/papers/volume5/lewis04a/lewis04a.pdf

[48] Inderjit S. Dhillon, *Co-clustering documents and words using Bipartite Spectral Graph Partitioning.* In: KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pages 269-274, New York, NY, USA, 2001. ACM Press.
http://www.cs.utexas.edu/users/inderjit/public_papers/kdd_bipartite.pdf

[49] Artemis Parvizi, Chris Huyck, Roman Belavkin, *Short Paper: Non-Taxonomic Concept Addition to Ontologies.*
http://aow2012.yolasite.com/resources/aow20120_submission_11.pdf

[50] Klaas Dellschaft, Steffen Staab, *On How to perform a Gold Standard Based Evaluation of Ontology Learning.* In: The Semantic Web - ISWC 2006, Lecture Notes in Computer Science, Volume 4273, pp 228-241, 2006.
http://link.springer.com/chapter/10.1007%2F11926078_17

[51] Alexander Maedche, Steffen Staab, *Ontology Learning for the Semantic Web.* In: IEEE Intelligent Systems, 16:72-79, 2001.
http://userpages.uni-koblenz.de/~staab/Research/Publications/ieee_semweb.pdf

[52] Lina Zhou, *Ontology learning: state of the art and open issues.* In: Information Technology and Management, 8:241252, 2007.
https://ai.wu.ac.at/~kaiser/birgit/Nonaka-Papers-Alfred/Zhou_OntologyLearning.pdf

[53] Mark Sanderson, Bruce Croft, *Deriving concept hierarchies from text.* In: Proceedings of the 22nd ACM Conference of the Special Interest Group in Information

Retrieval, pages 206-213, 1999.
http://www.seg.rmit.edu.au/mark/publications/my_papers/SIGIR99.pdf

[54] Jon Atle Gulla, Terje Brasehvik, *A Hybrid Approach to Ontology Relationship Learning.* In: Lecture Notes in Computer Science, Volume 5039, pp 79-90, 2008.
http://link.springer.com/chapter/10.1007%2F978-3-540-69858-6_9

[55] Mark Last, Abraham Kandel, *Automated Detection of Outliers in Real-World Data.* 03-2002.

[56] Charu C. Aggarwal, *Outlier Analysis.* Springer, January 2013.
http://www.charuaggarwal.net/outlierbook.pdf

[57] Frank van Meeuwen, *Multi-label text classification of news articles for ASDMedia.* Utrecht University, Department of Information and Computing Sciences, August 2013.

[58] Sergeja Vogrinčič, Zoran Bosnić, *Ontology-based multi-label classification of economic articles.*
http://www.doiserbia.nb.rs/img/doi/1820-0214/2011/1820-02141000034V.pdf

[59] David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel, Friedrich Leisch, *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien.* 2014.
http://CRAN.R-project.org/package=e1071

[60] Jean Hausser, Korbinian Strimmer, *entropy: Estimation of Entropy, Mutual Information and Related Quantities.* 2013.
http://CRAN.R-project.org/package=entropy

[61] Organisers: Massih-Reza Amini et al., *Fourth Challenge on Large Scale Hierarchical Text classification.* 2014.
http://lshtc.iit.demokritos.gr
http://www.kaggle.com/c/lshtc

[62] S. Kullback, R.A. Leibler, *On Information and Sufficiency.* Annals of Mathematical Statistics 22 (1): 7986, 1951.
http://www.csee.wvu.edu/~xinl/library/papers/math/statistics/Kullback_Leibler_1951.pdf