Automated Knowledge Discovery in Advanced Knowledge Management

Marko Grobelnik, Dunja Mladenić
J.Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia

Purpose

Present approaches and some research results of various research areas contributing to knowledge discovery from different sources, different data forms, on different scale, and for different purpose.

Design / methodology / approach

Contribute to knowledge management by applying knowledge discovery approaches to enable computer search for the relevant knowledge whereas the humans give just broad directions.

Findings

Knowledge discovery techniques provide to be very appropriate for many problems related to knowledge management. Surprisingly, it is often the case that already relatively simple approaches provide valuable results.

Research limitations / implications

Still there are many open problems and scalability issues that arise when dealing with real-world data and especially in the areas involving text and network analysis.

Practical implications

Each problem should be handled with care taking into account different aspects and selecting/extending the most appropriate available methods or developing some new approaches.

Originality / value

This paper provides an interesting collection of selected knowledge discovery methods applied in different context but all contributing in some way to knowledge management. Several of the reported approaches were developed in collaboration with the authors of the paper with especial emphases on their usability for practical problems involving knowledge management.

Automated Knowledge Discovery in Advanced Knowledge Management

Marko Grobelnik, Dunja Mladenic
J.Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia

Abstract

Knowledge Management is a discipline with many faces – among very provocative ones is the research area dealing with automatic discovery of the hidden truth within the data describing the world around us. The basic idea of knowledge discovery is to let the computer search for the knowledge whereas the humans give just broad directions about where and how to search. Surprisingly, it is often the case that already relatively simple techniques are able to uncover useful hidden truth beneath the surface of the known facts and relationships. In this paper we present approaches of various research subfields working in the area of knowledge discovery from different sources (such as databases, documents, networks, etc), in different forms (e.g. probabilistic, various kinds of logic, visualizations), on different scale (small data-sets or terra bytes), and for different purpose (e.g. prediction, segmentation, explanation). Knowledge Discovery will be presented in the light of one of the key paradigms within Knowledge Management with the emphasis on the cases where humans need to go steps further from what could be captured manually.

1. Introduction

Knowledge Discovery could be defined as a set of techniques coming mainly from the area of Artificial Intelligence but also borrowing important building blocks from other fields such as Statistics and Databases. The main goal of the whole area is to find useful pieces of knowledge within the data with none or little human involvement.

In this paper we describe relationship between Knowledge Management (KM) and Knowledge Discovery (KD). In this context, KM plays a role of very broad field dealing with 'knowledge' from all kind of aspects, while KD has a function of a sub-area within KM dealing with automatic and semi-automatic approaches for processing data for the purpose of extracting hidden and useful pieces of knowledge which are used further in the KM procedures.

An important stage in the development of KM and KD technologies was initiated by the maturity of the Internet technologies (approx. 10 years after its public appearance) and by a strong push from the commercial side. There is a big need to develop appropriate solutions, standards, tools, etc. to resolve many situations appearing on the market where classical way of dealing with data, integration, processes etc. are not appropriate anymore. The 'old' paradigm for building and integration of applications is getting simply too costly and cumbersome to be able to offer solution on another range of magnitude scale and complexity which will appear in the near future.

What is the reason for all these developments? One possible answer is that the focus of modern information systems is moving from "data-processing" towards "concept-processing", meaning that the basic unit of processing is less and less an atomic piece of data and is becoming more a semantic concept which caries an interpretation and exists in a context with other concepts.

The research and technological area that appeared as a consequence of such developments is "Semantic-Web" (SW). SW can be seen as mainly dealing with integration of many, already existing ideas and technologies with the specific focus of upgrading the existing nature of web-based information systems to a more "semantic" oriented nature. In this context SW could be viewed as a frontier of KM with some emphasis on web-based applications.

What is the role of Knowledge Discovery within Semantic-Web? There are several dimensions along which KD can bring important contributions to SW:

- SW applications involve deep structured knowledge composed into ontologies. Since KD techniques are mainly about discovering structure in the data, this can serve as one of the key mechanisms for structuring knowledge. We refer to such approaches as "ontology learning" which is usually performed in automatic or semi-automatic mode. The goal is to extract the structure from unstructured data-sources into an ontological structure being further used in KM process.
- Automatic KD approaches are not always the most appropriate, since often it is too hard or too costly to integrate the available background knowledge about the domain into fully automatic KD techniques. For such cases there are KD approaches such as "Active Learning" and "Semi-supervised Learning" which make use of small pieces of human knowledge for better guidance towards the desired model (e.g., ontology). The effect is that we are able to reduce the amount of human effort by an order of magnitude while preserving the quality of results.
- SW applications are typically associated with more or less structured data such as text-documents and corresponding metadata. An important property of such data is that it is relatively easy manageable by humans (e.g. people are good in reading and understanding texts). In the future we may expect applications in the areas where the data is not so "human friendly" (e.g. multimedia, signals, graphs/networks). In such situations there will be significant emphasis on automatic or semi-automatic methods offered by KD technologies which are not limited to a specific data representation.
- Language technologies (including lexical, syntactical and semantic levels of natural language processing) are benefiting from KD area in a great deal. Modeling a natural language includes number of problems where models created by automatic learning procedures from rare and costly examples enable to capture soft nature of language.
- Data and corresponding semantic structures change in time. As the consequence, we need to be able to adapt ontologies that are modeling the data accordingly –

we call this kind of structures “dynamic ontologies”. For most of such scenarios an extensive human involvement in building models from the data is not economical anymore, since it gets too costly, too inaccurate and too slow. Sub-field of KD called “stream mining” deals with these kinds of problems – the idea is to be able to deal with the stream of incoming data fast enough to be up-to-date with the corresponding models (ontologies). Again, this kind of scenarios might not be critical for today’s application but may become very important in the future.

- Scalability is one of the central issues in KD, especially in the sub-areas such as Data-Mining where one needs to be able to deal with real-life datasets of the terra-byte sizes. SW is ultimately concerned with real-life data on the web which have exponential growth – currently we talk about 10 billions of indexed web pages by major search engines. Because of this, approaches where human interventions are necessary will get inapplicable. KD with its focus to scalability will certainly be able to offer some answer to these questions.

Some of the above points describing possible intersections between KD and KM/SW are further discussed in the following sections. But first, let’s have some overview of KD area from the historical and scientific-focus point of view.

2. What is Knowledge Discovery?

Knowledge discovery could be defined as a research area with several subfields. There have been several definitions – here we cite just two of them which we find the most suitable for describing the main idea of the research area:

- Studying the design and analysis of algorithms for making predictions about the future based on past experiences (from http://www.learningtheory.org/)
- A process which aims at the extraction of interesting (non-trivial, implicit, previously unknown and potentially useful) information from data in large databases. (Fayad et al., 1996)

Knowledge Discovery as any other scientific field could be best identified (a) by typical scientific communities, (b) by typical problems, and (c) usual methods for solving the problems. In the next sections we’ll first describe main KD communities, and further, the problems and methods will be presented on the sub-area of KD called Text-Mining which is probably the most relevant from the KM perspective.

2.1 Scientific Communities within Knowledge Discovery

From a scientific point of view, KD could be structured in the following communities:

- *Computational Learning Theory* with a focus on mainly theoretical questions about learnability, computability, design and analysis of learning algorithms (http://www.learningtheory.org/). Main events are COLT (Computational Learning Theory) and ALT (Algorithmic Learning Theory) conferences.

- *Machine Learning*, where the main questions are how to perform automated learning on different kinds of data and especially with different representation languages for representing learned concepts. The field is in part applied and in part theoretical. Its main events are ICML (International Conf. on Machine Learning), ECML (European Conf. on Machine Learning), UAI (Uncertainty in AI), NIPS (Neural Information Processing Systems).
- *Data-Mining*, being rather applied area with the main questions on how to use learning techniques on large-scale real-life data. The main questions are about computational efficiency and quality of the results. Main events are KDD (ACM Knowledge Discovery in Databases), PKDD, ICDM (IEEE International Conf. on Data Mining), SDM (SIAM Data Mining), PKDD (Practice on Knowledge Discovery in Databases).

Each of three above fields of Knowledge Discovery has several sub-fields covering different aspects of data analysis depending of current research trends and market needs. E.g. recently there has been increased interest for learning in structured domains such as text (text-mining), web (web-mining), graphs/networks (link-analysis), learning models in relational/first-order form (relational data-mining), analyzing data streams (stream mining), etc.

2.2 Typical KD problems and solutions on Text Mining example

Data can be found in many different forms. Some of the formats are more appropriate for automatic data analysis and easier to handle than others. The usual data analysis methods assume that the data is stored in one or more tables, organized in a number of fields (called variables) with a predefined range of possible values. The question is, what can be done if the data is stored in a textual form, consisting of no records and no variables – just text. Are there any methods capable of handling the text data in order to obtain some insight from the data? Text mining is a field addressing such problems.

Text mining is an interdisciplinary area that involves at least the following key research fields:

- *Machine Learning and Data Mining* (Mitchell, 1997; Fayyad et al., 1996; Witten and Frank, 1999; Hand et al., 2001) which provides techniques for data analysis with varying knowledge representations and large amounts of data,
- *Statistics* and statistical learning (Hastie et al., 2001) which in the context of text mining contributes data analysis (Duda et al., 2000) in general,
- *Information Retrieval* (Rijsberg, 1979; Mani and Maybury, 1999) providing techniques for text manipulation and retrieval mechanisms, and
- *Natural Language Processing* (Manning and Schutze, 2001) providing the techniques for analyzing natural language. Some aspects of text mining involve the development of models for reasoning about new text documents based on words, phrases, linguistic and grammatical properties of the text, as well as extracting information and knowledge from large amounts of text documents.

One of the most popular applications of text mining is document categorization. Document categorization aims to classify documents into pre-defined taxonomies/categories based on their content. Other important problems addressed in text mining include document clustering, visualization, search based on the content, automatic document summarization, automatic construction of document hierarchies, document authorship detection and identification of plagiarism of documents, user profiling, information extraction, question answering in natural language. The following Sections briefly describe some of the approaches used on the problems we find the most relevant for SW.

2.2.1 Document categorization

Text document categorization can be applied when a set of predefined categories, such as "arts, education, science", are provided as well as a set of documents labeled with those categories. The task is to classify new (previously unseen) documents by assigning each document one or more content categories. This is usually performed by representing documents as word-vectors (usually referred to as the 'bag-of-words' representation) and using documents that have already been assigned the categories, to generate a model for assigning content categories to new documents (Jackson and Moulinier, 2002). In the word-vector representation of a document, a vector of word frequencies is formed taking all the words occurring in all the documents (usually several thousands of words). The representation of a particular document contains many zeros, as most of the words from the collection do not occur in particular document. The categories can be organized into an ontology, for example, the MeSH ontology for medical subject headings or the Yahoo! hierarchy of Web documents that can be seen as a topic ontology. Other applications of document categorization into hierarchies/taxonomies are of US patents, Web documents (McCallum et al., 1998; Mladenić, 1998; Mladenić and Grobelnik, 2003), and Reuters news articles (Kholer and Sahami, 1997).

2.2.2 Document clustering and similarity

Document clustering (Steinbach et al., 2000) is based on general data clustering algorithm adopted for text data by representing each document as a word-vector, which for each word contains some weight proportional to the number of occurrences of the word. The similarity of two documents is commonly measured by the cosine-similarity between the word-vector representations of the documents. For two documents *X* and *Y*, their similarity is calculated as:

$$\cos(X,Y) = \frac{\sum_i X_i Y_i}{\sqrt{\sum_j X_j^2 \sum_l Y_l^2}}$$

The same similarity is commonly used in document categorization for finding a set of the most similar documents (e.g., in *k*-Nearest Neighbor algorithm (Mitchell, 1997)) to a given document (target). Once again documents are represented as word-vectors and the cosine-similarity between the documents and the target is used to find the *k* most similar documents to the target.

2.2.3 Document visualization

Visualization of data in general and also of the textual contents of a document set is a method to obtain early measures of data quality, content, and distribution (Fayyad et al., 2001). For instance, by applying document visualization it is possible get an overview of the content of documents at a certain Web-site or in some other document collection. One approach to text document visualization is based on the clustering of the documents (Grobelnik and Mladenić, 2002) by first representing the documents as word-vectors and perform K-Means clustering (Steinbach et al., 2000) on the set of word-vectors. The obtained clusters are then represented as nodes in a graph, where each node in the graph is described by the set of most characteristic words in the cluster. Similar nodes – as measure by their cosine-similarity – are connected by a link. When such a graph is drawn, it provides a visual representation of the document set. Another example related to Semantic Web is visualizing news stories via visualizing relationships between named entities that appear in the news stories (Grobelnik and Mladenic, 2004) as shown in illustrative example in Figure 2.

2.2.4 User profiling

One of the main applications of user profiling (or user modeling) in text and Web mining is for filtering information, either content-based filtering or collaborative filtering. It is used to decide what information is potentially interesting for the user, for instance in the context of personalized search engines, browsing the Web, or shopping on the Web.
In the content-based approach to information filtering, the system searches for the items similar to those the user liked based on the content comparison. For instance, observing the user browsing the Web and providing help by highlighting potentially interesting hyperlinks on the requested Web pages (Mladenic, 2002). Content-based document filtering is based on document categorization (see Section 2.2.1). One of the main problems with this approach is that it is difficult to capture different, non-textual aspects of the document content (e.g., music, movies, and images). In addition to the representation problems, content-based systems tend to specialize the search for items similar to the ones already seen by the user.

The content-based approach can be successfully applied to a single user, which is in contrast with collaborative approaches that assume that there is a set of users using the same system. Here, advice is provided to the user based on the reaction of other similar users (Maes, 1994). Given a target user, the system searches for other users with similar interest, and then to the target user recommends the items, that these like minded users liked. In the collaborative approach o user profiling, instead of computing similarity between items the system computes similarity between users, and is often based on their item preferences. The assumption is that the users provide some kind of ratings for the items. In the collaborative approach there is no analysis of the item content, that is to say that items of any content can be handled with equal success. Each item is assigned a unique identifier and a rating given by the user. The similarity between users is based on the comparison of the ratings that they assigned to the same items. One of the main problems with the collaborative approach is that the small number of users relative to the number of items brings a danger of a sparse coverage of ratings. Also, for any new item in the database, information must be collected from different users to be able to recommend it, and similar users are not matched unless they have rated a sufficient number of the same items. Also if some user has unusual tastes compared to the rest of the system users, the system will not be able to find suitable similar users to him/her and the system performance will be poor.

3. Advanced Knowledge Discovery tasks for KM

3.1 Ontology Learning

Ontology is a fundamental data object for organizing knowledge in a structured way in many areas – from philosophy to Knowledge Management and Semantic Web. We usually refer to an ontology as being a graph/network structure consisting from:

1. a set of concepts (vertices in a graph),
2. a set of relationships connecting concepts (directed edges in a graph),
3. a set of instances assigned to a particular concepts (data records assigned to vertices in a graph.

More precise technical definition of an ontology depends on the actual context in which the ontology is used. Instances, concepts and relations can be of different nature in different ontologies. Examples of some more widely used types of ontologies are:

- Terminological ontologies where concepts are word senses and instances are words (such as WordNet ontology at http://www.cogsci.princeton.edu/~wn/),
- Topic ontologies where concepts are topics and instances are documents (such as DMoz at http://www.dmoz.org/ or Yahoo Directory at http://dir.yahoo.com/),
- Data-model ontology where concepts are tables in a data base and instances are data records (such as in a database schemas).

In general ontologies can be even more complex objects as the ones listed above – concepts and relationships can be described in e.g., first order logic, instances can be arbitrary complex data objects etc. In this section we try to stick with a rather operational definition of an ontology, which will enable us to define ontology learning problems.

From the perspective of using KD methods for automatic or semi-automatic ontology construction, the important part is identification of how to compare ontological instances to each other – the goal is to form concepts from sets of related instances and to put relations between them. For instance, in Information Retrieval and Text Mining, we have several measures for estimating similarity between documents as well as similarity between objects used within the documents (e.g., named entities, words, etc.) – these similarity measures can be used together with clustering algorithms as an approach for forming approximations of ontologies from document collections.

Operational definition of an ontology from the KD perspective needs to be rather technical and concise to the degree that enables estimating complexity of various semi-automatic ontology construction approaches. In Machine Learning terminology we refer to the ontology construction task as *ontology learning*. Consequently, we define an ontology just as another class of models (slightly more complex compared to usual Machine Learning models) which needs to be expressed in some kind of hypothesis language. This definition of ontology learning includes the following sub problems which are relevant in different contexts:

1. learning just the concepts,
2. learning just the relationships between the existing concepts,
3. learning both the concepts and relations at the same time,
4. populating an existing ontology/structure,
5. dealing with dynamic data streams.

The language (e.g. linear functions on one side of the spectrum or first order logic on the other extreme) that we use to express the concepts and their relationships determines complexity and power of the learning process. In the broader context, KD approach to automatic or semi-automatic ontology construction deals with some kind of data objects which need to have some kind of properties – may be text documents, images, data records or some combination of them.

One example of using KD techniques for extraction of semantic graphs from text of news articles is described in (Leskovec et al., 2004). The idea is to use ontological representation of a document (its semantic graphs) for learning how to automatically generate document summaries. The approach is based on performing deep parsing, co-reference resolution, anaphora resolution and extraction of subject-predicate-object triples. An example summary obtained using this approach is given in Figure 1. Another example of visualizing news stories via visualizing relationships between named entities that appear in the news stories (Grobelnik and Mladenic, 2004) can be seen in Figure 2.

3.1.1 Related work on building ontologies

Different approaches have been used for building ontologies, most of them using mainly manual methods. An approach to building ontologies was set up in the CYC project (Lenat and Guha, 1990), where the main step involved manual extraction of common sense knowledge from different sources. There have been some definitions of methodology for building ontologies, again assuming manual approach. For instance, the methodology proposed in (Uschold and King, 1995) involves the following stages: identifying purpose of the ontology (why to build it, how will it be used, range of the users), building ontology, evaluation and documentation. Building of the ontology is further divided in three steps. The first is ontology capture, where key concepts and relationships are identified, a precise textual definition of them is written, terms to be used to refer to the concepts and relations are identified, the involved actors agree on the definitions and terms. The second is step involves coding of the ontology to represent the defined conceptualization in some formal languages (committing to some meta-ontology, choosing a representation language and coding). The third step involves possible integration with existing ontologies. An overview of methodologies for building ontologies is provided in (Fernandez, 1999), where several methodologies, including the above described one, are presented and analyzed against the IEEE Standard for Developing Software Life Cycle Processes viewing ontologies as parts of some software product.

Recently, a number of workshops at Artificial Intelligence and Machine Learning conferences (ECAI, IJCAI, ECML/PKDD) have been organized on learning ontologies. Most of the work presented there addresses one of the following: a problem of extending an existing ontology WordNet using Web documents (Agirre et al., 2000), using clustering for semi-automatic construction of ontologies from parsed text corpora (Bisson et al., 2000), (Reinberger et al., 2004), learning taxonomic, eg., isa, (Cimiano et al., 2004) and non-taxonomic, eg., "hasPart" relations (Maedche and Staab, 2001), extracting semantic relations from text based on collocations (Heyer et al., 2001), extracting semantic graphs from text for learning summaries (Leskovec et al., 2004).

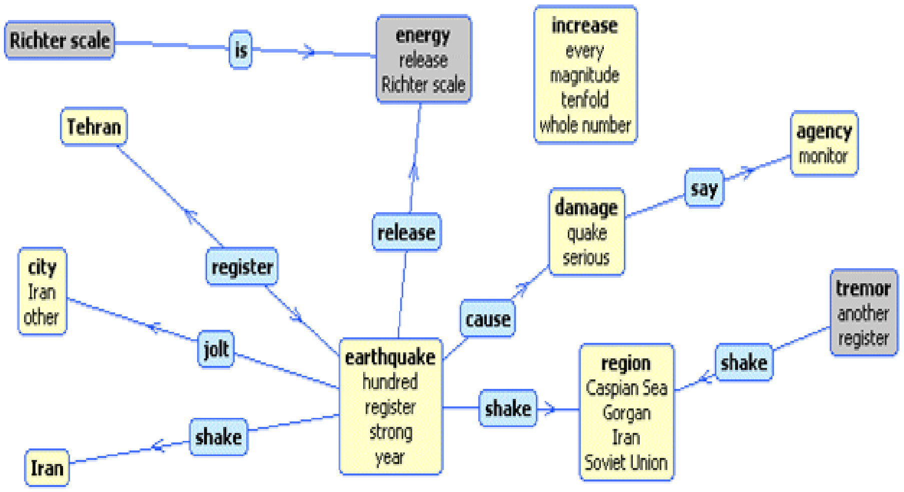


Figure 1. Visual presentation of an example summary of news story about earthquake.

Related to building ontologies is a problem of text annotation by finding labels (concepts) and assigning them to text fragments (instances). For instance, a text document is split into sentences, each sentence is represented as a word-vector, sentences are clustered and each cluster is label by the most characteristic words from its sentences. Some researchers have used WordNet to improve the results by mapping the found sentence clusters upon the concepts of a general ontology (Hotho et al., 2003). The found concepts are then used as semantic labels (XML tags) for annotating documents. A part of the ontology learning problem is labeling of the concepts, which is in the context of the described text annotation a problem of automatic labeling of text clusters (Popescull and Ungar, 2000).

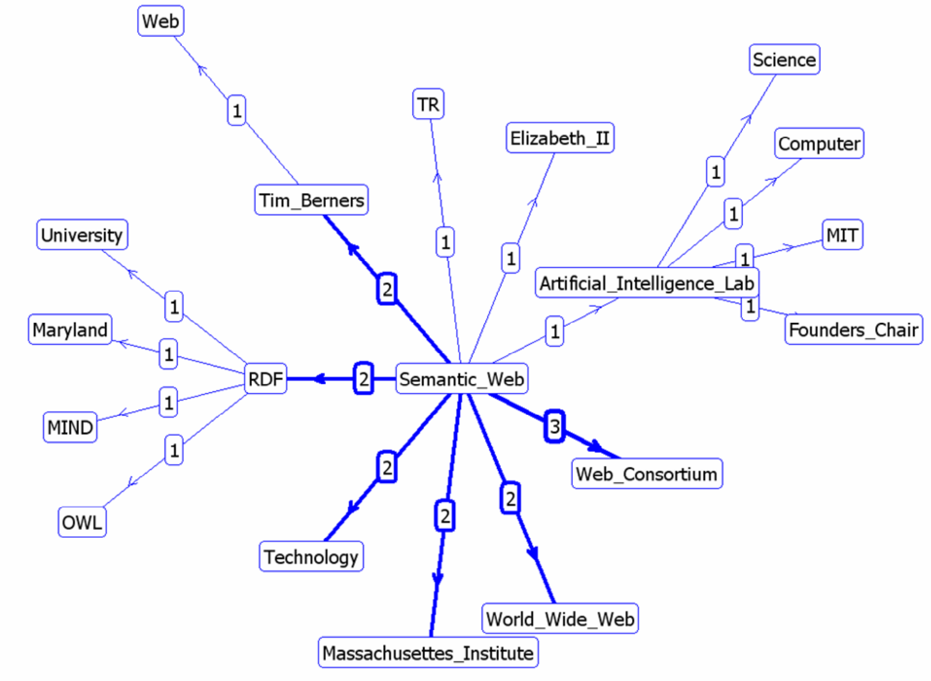


Figure 2. Visual presentation of relationship between named entities (vertices in the graph) appearing in news stories. Each edge shows intensity of co-mentioning of the two named entities. The graph was extracted from the 11.000 ACM Technology news stories from 2000-2004.

3.2 Dealing with unlabelled data

In many KD problem domains large amounts of data are available but the cost of correctly labeling it prohibits its use for model training. From KM & SW perspective especially relevant are large quantities of raw information available on the internet that present an interesting challenge of how to successfully exploit information hidden within it without first having to invest too much human resources into manual labeling. For instance, in the process of building ontologies, this can be very useful for populating ontology, where a large collection of relevant instances is available to be used in ontology population. These methods enable labeling the whole collection of relevant instances in a semi-automatic way involving manually labeling of only a small number of the instances.

In this section we will briefly present two methods: “Active learning” and “Semi-supervised learning” following the presentation in (Novak, 2004). Both methods try to build a model from partial information about the data provided in interactive or non-interactive way. They share and combine ideas from both more typical ways of modeling the data in KD: (a) Supervised learning (e.g. classification task, described in 2.2.1), and (b) Unsupervised learning (e.g. clustering task, described in 2.2.2). The main task of both methods is to attach labels to unlabeled data (such as documents, records etc.) by maximizing quality of label assignment and by minimizing the effort (human or computational).

Typical example scenario for using such methods would be assigning content categories to uncategorized documents from a large collection (e.g., from the Web or from news source). Usual situation is that it is too costly to label each document manually – but usually we also have available some limited amount of human resources. The task of "Active learning" and "Semi-supervised learning" is to use limited human's input in the most efficient way to assign high quality labels (e.g., in the form of content categories) to documents.

3.2.1 Active learning

Active learning has a tight link to the problem of "experiment design" addressed in statistical literature. It is a generic term describing a special, interactive kind of a learning process. In the contrast to the usual (passive) learning where the student is presented with a static set of examples that are then used to construct a model, active learning paradigm means the student can "ask" the "oracle" (e.g. domain expert, user,…) for a label of an example (see Figure 3).

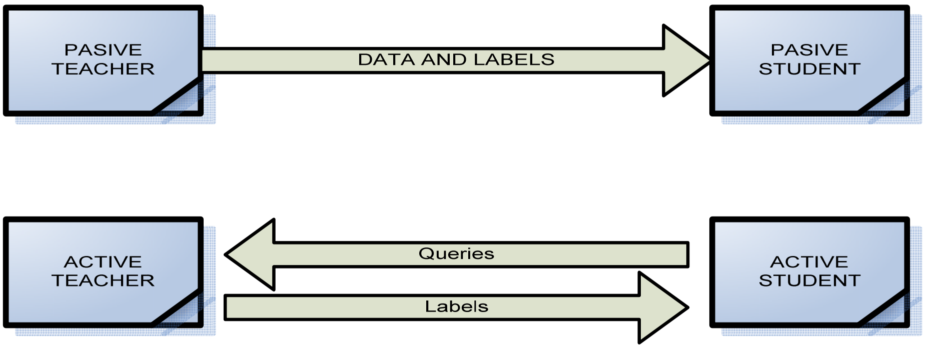


Figure 3. Illustration of passive versus active learning.

The intuition behind the paradigm of active learning is that having labels for a few highly informative examples provides much more information than having labels for many randomly chosen examples. Central question becomes: "what is an informative example?" or "what kind of query should we ask the oracle?". A simple and effective procedure is 'query filtering' from (Lewis and Gale, 1994): the student is provided a large amount of non-labelled examples that are viewed as potential queries. Since the problem of finding the optimum subset of the most interesting questions is hard, a greedy approximation is used in practice, so that a sample selection is interleaved with asking questions. Knowing some of the answers before having selected the next questions in the line in practice partially compensates for not selecting the optimal combination of questions. The basic active learning algorithm is the following:

```
Start with a small labelled set and a large set or a stream of unlabelled examples
repeat until some condition is met:
        from the unlabelled set select the currently most interesting example (or a batch of them)
        query the expert for the label
        add the now-labelled example to the labelled set
```

Based on the labels for a subset of the most informative examples from the dataset given by the “oracle”, we can approximate the labels for the rest of the unlabeled documents in the dataset. Figure 4 shows the typical learning curves when asking smart questions versus random questions.

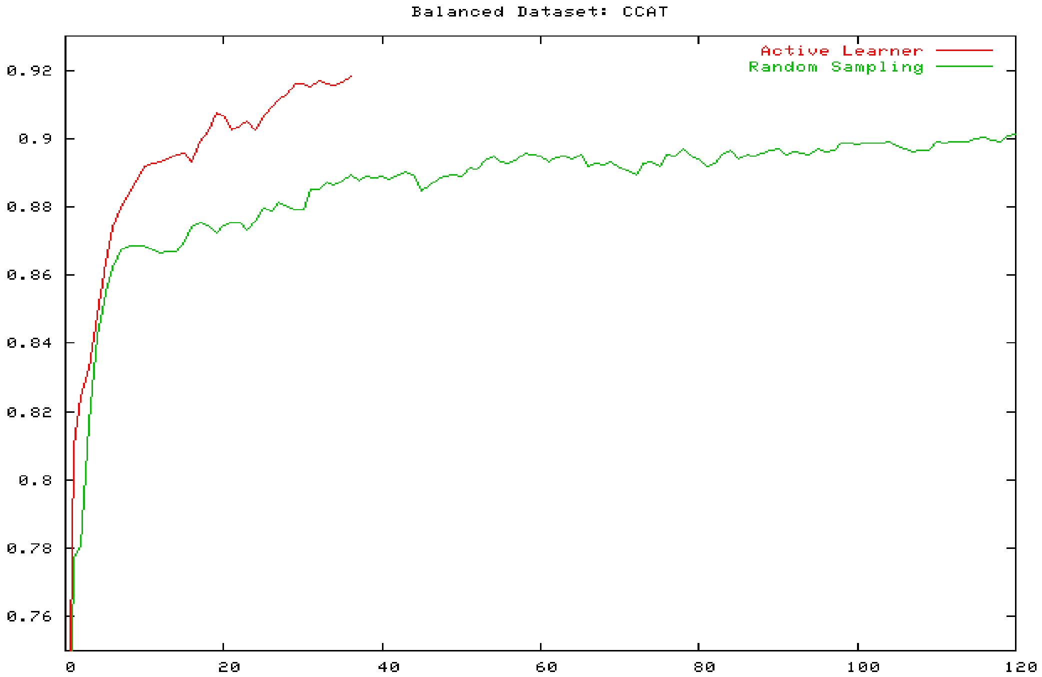


Figure 4. Comparison of typical learning curves for active learning (asking smart question) versus random sampling (asking random questions). The experiment was performed for learning on Reuters-News articles dataset for learning category CCAT. Active learning procedure needs fewer trials (X axis) for better success (Y axis) measured in F1 (measure of quality used in KD and Information Retrieval).

3.2.2 Semi-supervised learning

In comparison to active learning, semi-supervised learning (Seeger, 2000) is not an interactive procedure. In this setting the system is provided with a fixed set of labeled examples and unlabelled examples – information from the labeled part in addition to the statistical properties of the dataset is then used to gain an insight on the data. Final result is the assignment of the labels to the unlabeled part of the dataset.

Popular and effective method for semi-supervised learning is transduction – the idea is just to provide labels for unlabeled part of the dataset without an explicit model to be constructed during the procedure. One common approach is to first construct a graph using all of the examples as vertices and connect those vertices that are similar – close to each other according to some chosen distance measure – and assign that distance as the weight of the edge. That way an implicit bias based on the neighbourhood relationship is introduced into the model which provides the required background knowledge for the algorithm to be able to make use of unlabeled data. The intuition is that labels from the labeled examples are propagated to the unlabeled ones. The simplest algorithm for assignment of binary labels is based on "graph mincut" algorithm (Blum and Chawla, 2001) by minimizing the number of similar vertices with different labels. Here is the outline of the algorithm:

```
construct a graph using all of the examples
add two more vertices (one for each label) (+), (-)
connect labelled vertices with the corresponding (+) or (-) vertex with edges of infinite weight
connect the rest of the vertices with edges weighted by the similarity function
find the minimum cut between (+) and (-) thus minimizing the number of similar vertices that will be given different labels
assign labels to the unlabelled vertices depending on which side of the cut they are
```

3.3 Language Technologies

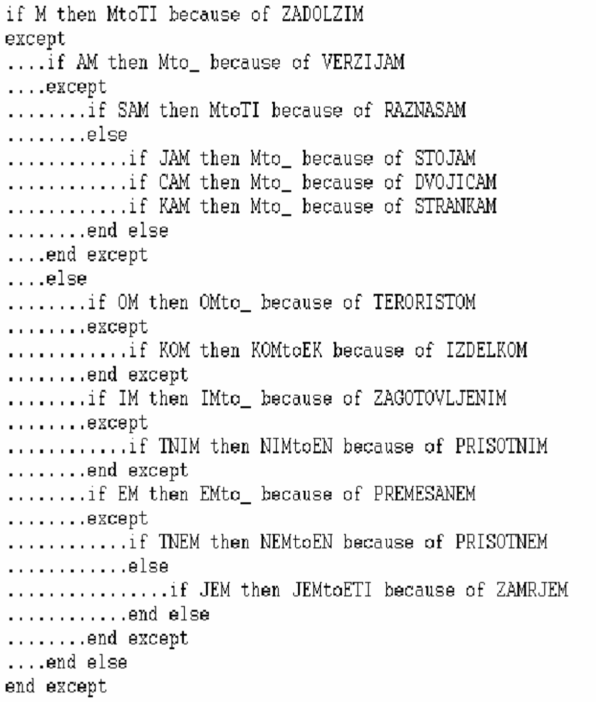
Language technologies (LT) (Manning and Schutze, 2001) comprise all aspects of dealing with natural language data. Although LT in general cover many fields, for the purpose of this paper we will consider three main levels of language processing each requiring different approaches and also providing different information about the language data:

1. Lexical level, where the main unit of processing is lexeme. This level of processing usually deals with language morphology – typical problems are stemming and lemmatization (normalization of the words), word sense disambiguation (detecting different sense of the word or common sense for different words) etc.
2. Syntactical level, where the goal is to find the syntactical structure within the text. The unit of processing is a statement – output is in a case of simpler methods a set of annotations (when performing shallow parsing) or a parse tree denoting deeper syntactical structure of the parsed sentence.
3. Semantic level, where the goal is to approach understanding of the document. The unit of processing is one or more documents – depends on the application. The output of such methods are usually some shallow aspects of understanding – such as semantic networks extracted from text or in the best case (usually not widely applicable) some form of logic formulas trying to capture objects and their relationships from the text. This level of language processing is certainly the most difficult one and is still a long way to go before we will really be able to understand the text in general.

KD techniques are well integrated in many aspects of LT combining human background knowledge about the language with automatic approaches for modeling "soft" nature of ill structured data formulated in natural language. In the next subsections we will describe usage of KD technology in three tasks dealing with different aspects of LT: learning word lemmatization, identification of language independent document representation, and document summarization. More on the usage of LT in knowledge management can be found in (Cunningham and Bontcheva, 2005), this journal issue.

3.3.1 Learning word lemmatization

One of the usual first steps when dealing with the text in Information Retrieval or Text Mining is to transform the words into their normal form. This operation is important since we would like to merge words with the same meaning into one word (otherwise analytic methods may have too hard job). Two types of operations are usually performed: (1) stemming where we transform the word into its stem (which often doesn't represent a meaningful word), or (2) lemmatization where the word is transformed into its regular normal form (e.g., for nouns into first person singular). Lemmatization is more difficult task but also more preferred when processing language. Stemmers and lemmatizers are almost always represented as a set of transformation rules to be efficiently applied on the text.



```
if M then MtoTI because of ZADOLZIM
except
....if AM then Mto_ because of VERZIJAM
....except
........if SAM then MtoTI because of RAZNASAM
........else
............if JAM then Mto_ because of STOJAM
............if CAM then Mto_ because of DVOJICAM
............if KAM then Mto_ because of STRANKAM
........end else
....end except
....else
........if OM then OMto_ because of TERORISTOM
........except
............if KOM then KOMtoEK because of IZDELKOM
........end except
........if IM then IMto_ because of ZAGOTOVLJENIM
........except
............if TNIM then NIMtoEN because of PRISOTNIM
........end except
........if EM then EMto_ because of PREMESANEM
........except
............if TNEM then NEMtoEN because of PRISOTNEM
............else
................if JEM then JEMtoETI because of ZAMRJEM
............end else
........end except
....end else
end except
```

Figure 5. Ripple-down-rule for last-word-letter 'M in Slovenian language. The first line reads as: "if the word ends with the letter 'M' then replace the word ending 'M' by the ending 'TI' (MtoTI) which happened because of the word 'ZADOLZIM'.

It is important to know there are many languages in the world which do not have stemmers and lemmatizers yet and therefore it is important to be able to create such models automatically from the language data. Usually we perform learning from the set of pairs (inflected-word, lemma) which serve as examples for machine learning algorithms.

Here we are presenting an approach taken for creating lemmatizer for Slovenian language. Interesting property of Slovenian language is that it has approx. 20 inflected words (different surface forms) per one normalized word – this number is much lower for e.g. English (approx. 5 to 1). The approach which was taken in (Plisson et al, 2004) was to apply Ripple-down-rules learning algorithm on the set of approx. 500,000 (inflected-word, lemma) pairs. One rule was created per one letter based on the last letter of the word. The example rule for last-word-letter 'M' is presented on the Figure 5. The rule is represented as a set of cascaded if-then-else statements. Each if-then part has 'because' clause providing an example reason for that particular decision and 'except' clause listing the exceptions to the rule. The whole set of rules for Slovenian achieved accuracy of 92% on average (using 5-fold cross-validation) which was significantly better compared to other automatic approaches and even better then manual set of rules.

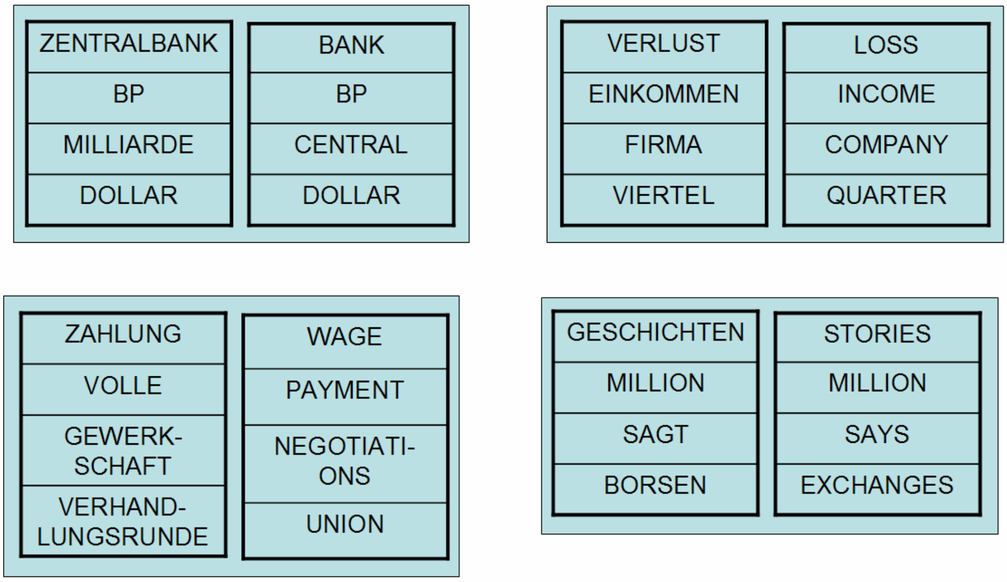
3.3.2 Language independent document representation

Multilinguality causes difficulties in many situations of text processing. For various tasks the problem could be more or less critical – but in general, most of the applications usually do not deal with this problem yet. Some of the most important problems are multilingual document retrieval, classification, clustering, etc., where one of the basic building blocks is calculating similarity between the documents written in different languages. Standard similarity measures, such as cosine distance (see Section 2.2.2) would proclaim two documents with the same content but written in two different languages as totally different. Therefore, the question is if we are able to map each document independent of the language into a subspace (in linear algebra sense), where the documents with the similar contents would lie close to each-other independent of the fact that in the original form both documents used completely different words for their content description.

A solution giving good results to the above problem is Canonical Correlation Analysis (CCA), a technique for finding common semantic features between different views of data. Canonical Correlation Analysis (CCA) is a method of correlating two multidimensional variables. It makes use of two different views of the same semantic object (e.g. the same text document written in two different languages) to extract representation of the semantic. Input to CCA is a paired dataset $S = \{(u_i,v_i);\ u_i \in U,\ v_i \in V\}$, where U and V are two different views on the data – each pair contains two views of the same document. The goal of CCA is to find the common semantic space W and the mappings from each U and V into W space. All documents from U and V can be mapped into W to obtain a view independent representation.

To illustrate on an example from (Fortuna, 2004): Let space V be vectors-space model for English and U vector-space model for German text documents. Paired dataset is than a set with pairs made of English documents, together with their German translation. The output of CCA on this dataset is a semantic space where each dimension shares similar English and German meaning. By mapping English or German documents into this space, language unexpanded representations are obtained. In this way, standard machine learning algorithms can be used on multi-lingual datasets. Figure 6 is showing part of paired vector-spaces represented by some of the characteristic eigenvectors – it can be observed that the methods automatically finds good mapping between words in both languages.

The future research goal is to produce mapping for many such languages which would enable to construct truly “language-independent-space” for text documents.



ZENTRALBANK	BANK
BP	BP
MILLIARDE	CENTRAL
DOLLAR	DOLLAR

VERLUST	LOSS
EINKOMMEN	INCOME
FIRMA	COMPANY
VIERTEL	QUARTER

ZAHLUNG	WAGE
VOLLE	PAYMENT
GEWERK-SCHAFT	NEGOTIATI-ONS
VERHAND-LUNGSRUNDE	UNION

GESCHICHTEN	STORIES
MILLION	MILLION
SAGT	SAYS
BORSEN	EXCHANGES

Figure 6. Four pairs of aligned eigenvectors for German and English language learned from the Reuters news. These kind of eigenvectors could be further used for mapping new German or English documents into language neutral subspace where the statistical properties of similar documents written in different languages are similar independent of the language.

3.3.3 Document Summarization using Semantic Graphs

Document summarization is one of the key tasks in Text Mining and Information Retrieval. It is interesting that most of the existing approaches use very shallow approach which work well only to a certain degree – that might be also the reason that summarization is not used as widely as it could be. Philosophically speaking, the ‘true summarization’ could be done only with in-depth semantic analysis of the document and on the top of this, creating a summary would mean generating a text selected by the importance of content pieces.

In this section we will present an experiment from (Leskovec et al., 2004) which exploit semantic structure of the text represented by a semantic graph created from the original text. By semantic graphs we refer to graphs constructed from Subject-Predicate-Object triples extracted from the sentences within the document. Next, on the Subjects and Objects, co-reference resolution (anaphora resolution and name entity consolidation) was applied. Each consolidated triple represented an edge within a semantic graph. For the experiment set of pairs (document, summary) was taken. For each document and its summary a semantic graph was generated and transformed into set of learning examples. Within the learning problem positive examples were the triples which were selected by humans from the original document into its summary. With machine learning we were able to create a model being able to weight the triples from a semantic graph which are more important to be selected for the summary. From the selected triples a new, smaller semantic graph was created – out of this summary-graph the true textual summary could be generated. The whole process is presented within the Figure 7. The evaluation showed that we were able to extract into summary approx. 40% of all the triples also extracted by humans – this results are comparable to other best summarization procedures which work with more shallow approach.

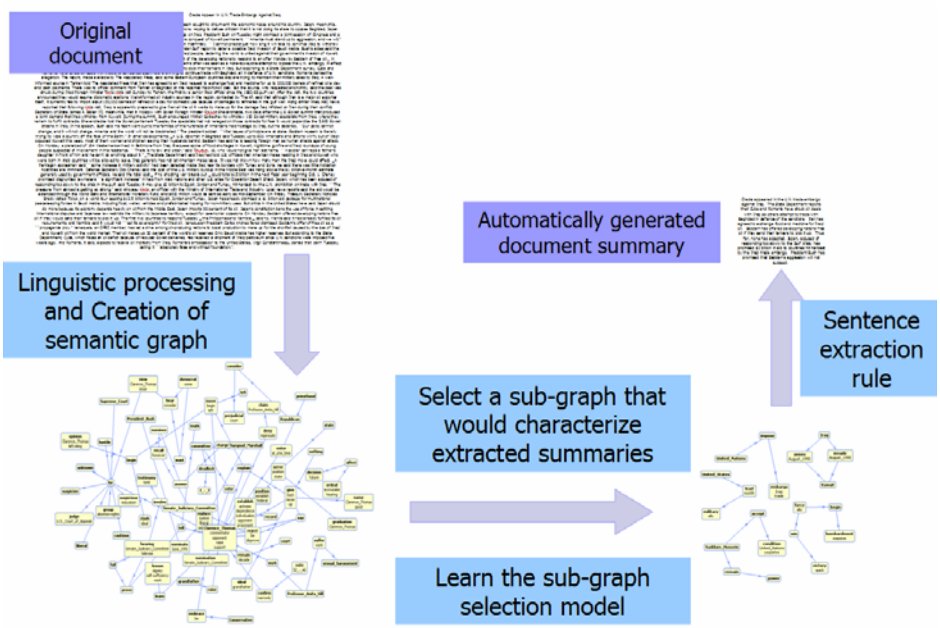

Original document
Automatically generated document summary
Linguistic processing and Creation of semantic graph
Sentence extraction rule
Select a sub-graph that would characterize extracted summaries
Learn the sub-graph selection model

Figure 7. Process of creating a summary, first by transforming an original document into semantic graph, next by applying a model for selecting the most relevant parts of the graph for the summary forming smaller semantic graph and finally, generating text from smaller semantic graph into.

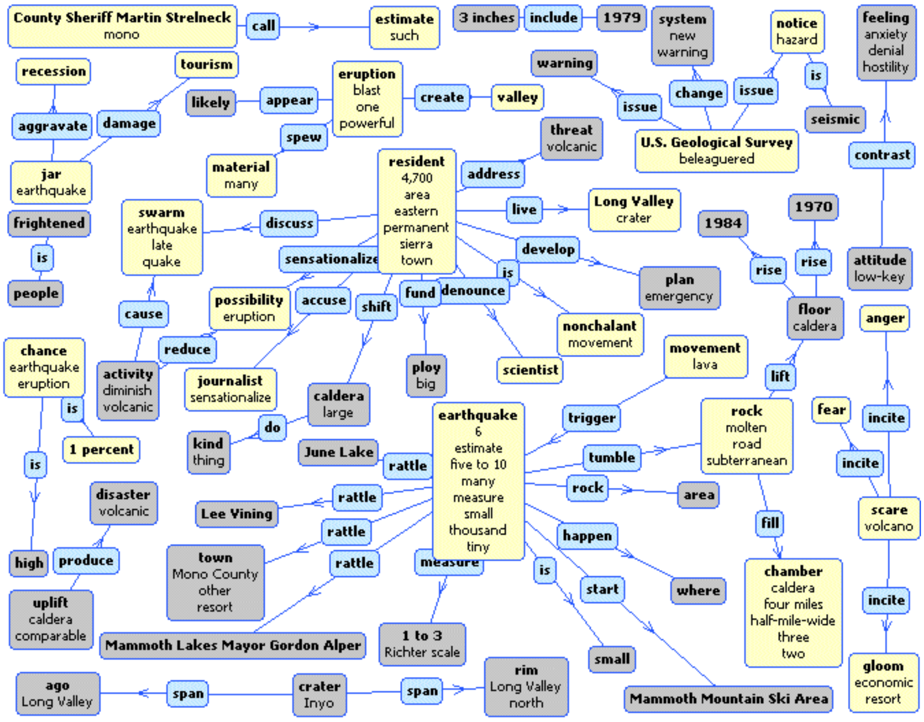


Figure 8. Full-Semantic graph of a news article on an earthquake.

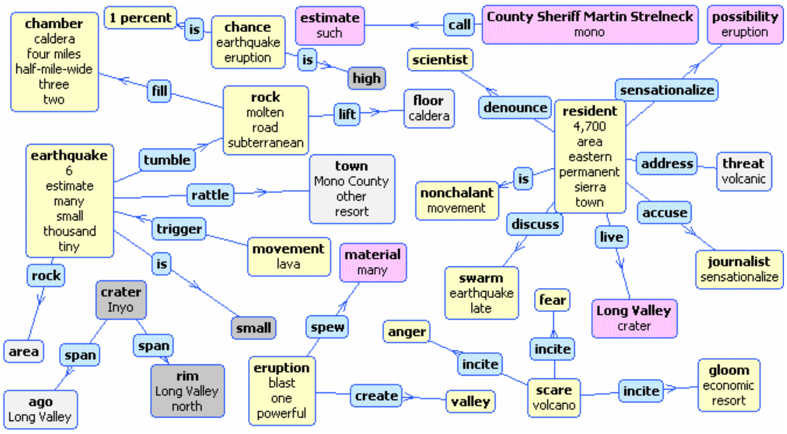


Figure 9. Summary-Semantic graph of a news article presented in Figure 8.

Conclusion

In this paper we presented the research area of Knowledge Discovery and its overlapping with various aspects of Knowledge Management with some focus on Semantic Web. Knowledge Discovery is an area consisting from several research communities sharing similar type of methods approaching analytical problems in various domains. As the main intersection between Knowledge Discovery and Knowledge Management would be learning in highly structured domain and providing models and advice for manipulating the structure. Usually this kind of results is related to ill structured data represented in natural language. Therefore, we presented:

- Text-Mining as a subfield within Data-Mining,
- ontology learning as a key technology for interacting with Semantic Web,
- dealing with unlabeled data as a key technology for efficient utilization of human resources, and
- the role of Knowledge Discovery in Language Technology.

Acknowledgements

This work was supported by the Slovenian Research Agency and the IST Programme of the European Community under SEKT Semantically Enabled Knowledge Technologies (IST-1-506826-IP) and PASCAL Network of Excellence (IST-2002-506778). This publication only reflects the authors' views.

References

Agirre, E., Ansa, O., Hovy, E., Martínez, D. (2000). Enriching very large ontologies using the WWW. In Proceedings of the First Workshop on Ontology Learning OL-2000. The 14th European Conference on Artificial Intelligence ECAI-2000.

Bisson, G, Nédellec, C., Cañamero, D. (2000). Designing clustering methods for ontology building: The Mo'K workbench. In Proceedings of the First Workshop on Ontology Learning OL-2000. The 14th European Conference on Artificial Intelligence ECAI-2000.

Blum, A., Chawla, S. (2001). Learning from Labelled and Unlabelled Data Using Graph Mincuts, Proceedings of the 18th International Conf. on Machine Learning, pg. 19-26.

Chakrabarti. S., (2002). Mining the Web: Analysis of Hypertext and Semi Structured Data, Morgan Kaufmann.

Cimiano, P., Pivk, A., Schmidt-Thieme, L., Staab, S. (2004) Learning Taxonomic Relations from Heterogeneous Evidence. In Proceedings of ECAI 2004 Workshop on Ontology Learning and Population.

Craven, M., Slattery, S., (2001). Relational learning with statistical predicate invention: Better models for hypertext. Machine Learning, 43(1/2):97-119.

Cunningham, H., Bontcheva, K. (2005) Knowledge Management and Human Language: Crossing the Chasm. Journal of Knowledge Management, this issue.

Duda, R. O., Hart, P. E. and Stork, D. G. (2000). Pattern Classification 2nd edition, Wiley-Interscience.

Fayyad, U., Grinstein, G. G. and Wierse, A. (editors), (2001). Information Visualization in Data Mining and Knowledge Discovery, Morgan Kaufmann.

Fayyad, U., Piatetski-Shapiro, G., Smith, P., and Uthurusamy R. (eds.) (1996) Advances in Knowledge Discovery and Data Mining. MIT Press, Cambridge, MA, 1996.

Fernández, L.M. (1999). Overview Of Methodologies For Building Ontologies. In Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5).

Fortuna, B., (2004). Kernel Canonical Correlation Analysis With Applications. Proceedings of the 7th International multi-conference Information Society IS-2004, Ljubljana: Institut "Jožef Stefan", 2004.

Grobelnik, M., and Mladenić, D., (2002). Efficient visualization of large text corpora. Proceedings of the seventh TELRI seminar. Dubrovnik, Croatia.

Grobelnik, M., Mladenic, D. (2004). Visualization of news articles. Informatica journal, 2004, vol. 28, no. 4.

Hand, D.J., Mannila, H., Smyth, P. (2001) Principles of Data Mining (Adaptive Computation and Machine Learning), MIT Press.

Hastie, T., Tibshirani, R. and Friedman, J. H. (2001). The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer Series in Statistics, Springer Verlag.

Heyer, G., Läuter, M., Quasthoff, U., Wittig, T., Wolff, C. (2001) Learning Relations using Collocations. In Proceedings of IJCAI-2001 Workshop on Ontology Learning.

Hotho, A., Staab, S., Stumme, G. (2003) Explaining text clustering results using semantic structures. In Proceedings of ECML/PKDD 2003, LNAI 2838, pages 217-228, Springer Verlag.

Jackson, P., Moulinier, I., (2002). Natural Language Processing for Online Applications: Text Retrieval, Extraction, and Categorization, John Benjamins Publishing Co.

Koller, D., Sahami, M., (1997). Hierarchically classifying documents using very few words, Proceedings of the 14th International Conference on Machine Learning ICML-97, pp. 170-178, Morgan Kaufmann, San Francisco, CA.

Leskovec, J., Grobelnik, M., Milic-Frayling, N. (2004). Learning Sub-structures of Document Semantic Graphs for Document Summarization. In Workshop on Link Analysis and Group Detection (LinkKDD2004). The Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Leskovec, J., Grobelnik, M., Milic-Frayling, N. (2004). Learning Semantic Graph Mapping for Document Summarization. In Proceedings of ECML/PKDD-2004 Workshop on Knowledge Discovery and Ontologies (KDO-2004).

Lewis, D.D., Gale, W.A. (1994). A sequential algorithm for training text classifiers, Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval.

Maedche, A., Staab, S. (2001). Discovering conceptual relations from text. In Proc. of ECAI'2000, pages
321-325.

McCallum A., Rosenfeld R., Mitchell T., Ng A., (1998). Improving Text Classification by Shrinkage in a Hierarchy of Classes, Proceedings of the 15th International Conference on Machine Learning ICML-98, Morgan Kaufmann, San Francisco, CA.

Mani, I., Maybury, M.T. (editors), (1999). Advances In Automatic Text Summarization, MIT Press.

Manning, C.D., Schutze, H. (2001).Foundations of Statistical Natural Language Processing, The MIT Press, Cambridge, MA.

Mitchell, T.M. (1997). Machine Learning. The McGraw-Hill Companies, Inc.

Mladenić, D. (1998). Turning Yahoo into an Automatic Web-Page Classifier. Proc. 13th European Conference on Artificial Intelligence (ECAI'98, John Wiley & Sons), 473–474.

Mladenić, D. (2002). Web browsing using machine learning on text data, In (ed. Szczepaniak, P. S.), Intelligent exploration of the web, 111, Physica-Verlag, 288–303.

Mladenić, D., Grobelnik, M. (2003). Feature selection on hierarchy of web documents. Journal of Decision support systems, 35, 45-87.

Mladenić, D., Grobelnik, M. (2004). Mapping documents onto web page ontology. In: Web mining : from web to semantic web (Berendt, B., Hotho, A., Mladenić, D., Someren, M.W. Van, Spiliopoulou, M., Stumme, G., eds.), Lecture notes in artificial inteligence, Lecture notes in computer science, vol. 3209, Berlin; Heidelberg; New York: Springer, 2004, 77-96.

Nigam, K., McCallum, A., Thrun, S., and Mitchell, T., (2001). Text Classification from Labeled and Unlabeled Documents using EM, Machine Learning Journal.

Novak, B., (2004). Use of unlabeled data in supervised machine learning. Proceedings of the 7th International multi-conference Information Society IS-2004, Ljubljana: Institut "Jožef Stefan", 2004.

Plisson, J., Lavrac, N., Mladenic, D., (2004). A rule based approach to word lemmatization. Proceedings of the 7th International multi-conference Information Society IS-2004, Ljubljana: Institut "Jožef Stefan", pp. 83-86.

Popescul, A., Ungar, L.H. (2000). Automatic labeling of document clusters. Department of Computer and Information Science, University of Pennsylvania, unpublished paper available from
http://www.cis.upenn.edu/~popescul/Publications/popescul00labeling.pdf

Reinberger, M-L., Spyns, P. (2004) Discovering Knowledge in Texts for the learning of DOGMA-inspired ontologies. In Proceedings of ECAI 2004 Workshop on Ontology Learning and Population.

Rijsberg, C. J., van (1979), Information Retrieval, Butterworths.

Sebastiani, F., Machine Learning for Automated Text Categorization, ACM Computing Surveys, 2002.

Seeger, M. (2000) Learning with Labelled and Unlabelled Data. Technical Report, Edinburgh University.

Steinbach, M., Karypis, G. and Kumar, V. (2000). A comparison of document clustering techniques. Proc. KDD Workshop on Text Mining. (eds. Grobelnik, M., Mladenić, D. and Milic-Frayling, N.), Boston, MA, USA, 109–110.

Uschold, M., King, M. (1995). Towards a methodology for building ontologies. In Workshop on Basic Ontological Issues in Knowledge Sharing. International Joint

Conference on Arti.cial Intelligence, 1995. Also available as AIAI-TR-183 from AIAI, the University of Edinburgh.

Witten, I.H., Frank, E., (1999) Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann.