Semi-automated Thesaurus Enrichment

A Case Study in the Economic Domain

Jaron van Boheemen
4-8-2005

Table of Contents

1. Introduction

1.1 Information Overload

Progress usually is a good thing. It makes life easier for most of us. One of the domains where the biggest progress has been made in the last few years is the Information Technology. Thanks to new technologies information can be available world-wide in just a few seconds. Anybody can access all available information around the world.

A lot of information is available, actually too much information is available. It is quite impossible to read all information available on a certain subject. When you think you have finished reading about a subject, new information is available. It gets harder and harder to keep up with the latest research, or to get an overview of the state of the art. It seems that information overload is a bigger problem than information shortage these days. Information overload is reaching, as long forecasted, new heights. It is becoming increasingly difficult to manage it all, or even to decide what to discard and what not. In the end, lots of relevant information is lost or it never reaches its ideal destinations. [23]

1.2 Organising Information

To create order in the chaos of the information overload, information has to be organised. Ways have to be found to get the right information to the right place at the right time. Some solutions are already available; others are waiting to be discovered. Existing solutions are search engines, intranets of companies, websites where documents on a certain domain are published and ordered, or electronical agents search a network for certain information.

Some more specific examples of systems that organise information are a semantic publishing system and an associative search engine. A semantic publishing system is a system which organises articles. When an author offers an article to this system, the system analyses the article. Based on a structured vocabulary and information from other documents, the system tries to retrieve semantic information from the article. This semantic information serves as a label for the article. This way of publishing makes it easier for people to find the article they are looking for. Based on the semantic information, it even might be possible to judge the quality of the article. The second example, an associative search engine, is an engine that searches for information, based on associations between (parts of) documents. This is the kind of search engine which the VICORE project [22] is trying to develop. This engine searches for associations between documents and tries to visualize these relations. This way, one can get a clear overview of a specific domain, or discover new relations between certain topics.

More and more we let computers do, or assist in, our search for information. One of the

biggest problems is that most information is written in human language. Computers don't deal well with human language. Therefore it is necessary that we either present our information in a computer readable form, or teach computers how to 'understand' our language. An example of the latter scenario is Natural Language Processing (NLP). Here, rules are created by which computers can analyse human language. In certain research domains researchers are quite far in enabling computers to process human language. In the medial domain, for example, most relevant words can be recognized by computers. In the economic domain, on the other hand, much less progress has been made on this subject. Therefore there is much need for methodologies that make computers process human language in the economic domain. An example of a way to present information in a computer readable form is the Semantic Web [24]. The purpose of the semantic web is to add computer readable semantics to all information on the Internet, in order to make the information processable for computers. The ultimate goal is to make computers process information in such a way that it comes as close as possible to human understanding. The semantics are added by adding labels to pieces of text that describe the specified text in a computer readable format, named the Resource Description Framework (RDF) [25]. Another way to enable computers to determine the contents of a document, is to index all documents by representing them, for example, with a number of keywords.

All examples have something in common. To enable computers to process information in a way that is useful for us, the same terminology has to be used. If two articles about the same subject are represented by different keywords, a computer won't be able to see that the articles are about the same subject. So if a computer searches for the keyword by which the first article is represented, it won't find the second article. A way to solve this problem is by creating a vocabulary. Different kinds of vocabularies exist. Some only indicate which words can be used to describe the same 'thing' (synonyms). Other, more advanced, vocabularies, like thesauri, also describe one or more kinds of relations between words. These vocabularies can be very helpful for representing information in a standardised form. However, vocabularies that are created today can be outdated tomorrow. Therefore they should be updated frequently.

1.3 Goal and Structure

My goal is to create a structured vocabulary and develop a methodology for enriching and updating this vocabulary. This vocabulary can be used as a building block for applications that help people to find the information they need. To make this possible, the applications must be able to use the vocabulary to adequately represent the semantics of a document and the relations between the terms used to represent the documents. Because my study has an economic background, I shall do this in the economic domain.

First I shall describe some common structured vocabularies and discuss the use and construction of those vocabularies (chapter 2). Next, in chapter 3, I shall describe and analyse some existing methodologies for construction of vocabularies. In chapter 4 I shall create my own methodology.

Finally the results of my methodology and a conclusion shall be available in respectively chapters 5 and 6.

2. Structured Vocabularies

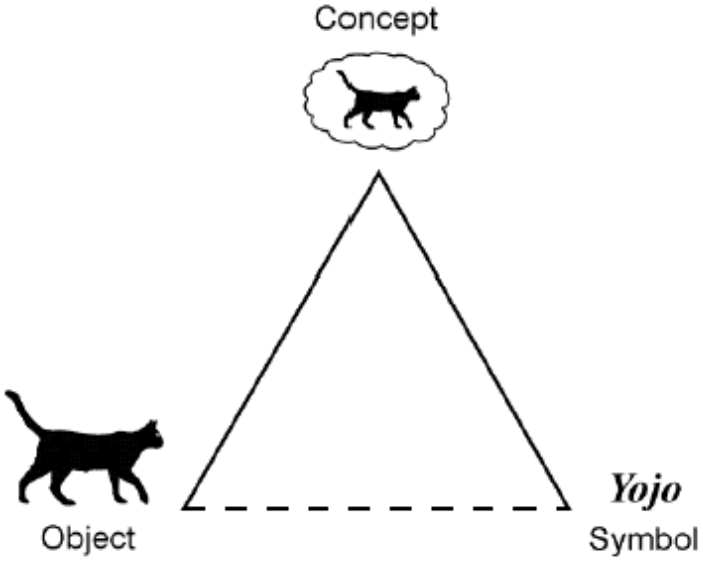
Several different structured vocabularies exist. In this chapter I shall describe what they can be used for and discuss some issues that arise when creating a certain type of vocabulary. But first I shall describe some of the best-known types of vocabularies.

2.1 Terminology

The best-known and most used types vocabularies that can be used to enable computers to analyse human language are taxonomies, thesauri and ontologies. These types are often mixed up with eachother. To avoid any confusion I shall explain the meaning of these terms and describe their differences and similarities. I shall also explain what a ‘concept’ is, since all mentioned vocabularies make use of it.

2.1.1 Concepts

A concept is an unambiguous representation of an object or 'something' in the real world in someone‘s mind. It can exist of words, descriptions, symbols, real objects and thoughts. The concept can represent any concrete or abstract 'thing' from the real world that you can think of. It can best be explained by the Ogden-Richardson meaning triangle [8].



The Ogden-Richardson meaning triangle

The triangle relates objects in the real world, concepts that correspond to these objects, and the symbols that languages use to refer to them [9]. The relation between the symbol and the object itself is dashed, because it is indirect and mediated by an interpreter. On hearing, seeing or reading the symbol, the interpreter compares this symbol to the conceptual knowledge that it has present, hereby developing a thought that links the symbol to the thing in the real world. Only through an interpreter a mapping can be made between the symbol and the object [10].

A human being can obviously develop a thought to relate the object to the symbol and vice versa. Due to our complex brain, background information is immediately available to process information delivered to our brain by our senses. Unfortunately, current computers can't. Although there are many ways to represent information, like databases, first order logic, Bayesian networks, Associative Concept Spaces (ACS), or decision trees, computers still have poor conceptual representation and reasoning abilities and deal with semantics in a limited way. An important subclass of the knowledge representation domain is that of structured vocabularies. In this domain, at this moment, the digital representation of a concept can contain most information when the concept is part of a semantic network that describes a part of the real world. However, this is also a representation which is quite hard to create. A digital concept in a structured vocabulary can exist of not much more then a collection of words and descriptions (and in some cases graphical representations) that have the same meaning as the 'real' concept in the real world. Therefore, it is quite hard to let computers make a match between real objects and symbols by using the digital information that is stored on their harddisks.

In the rest of this text, when I use the word 'concept', I shall refer to a digital representation of a concept, unless indicated otherwise. These digital concepts will consist of a set of synonyms. All equivalent words in a certain context will be part of the same concept. Of all the synonyms in a concept, one is chosen as the label of the concept. This synonym will be referred to as the preferred term. The other synonyms in the concept are given the status of non-preferred term.

2.1.2 Taxonomy

A taxonomy is a structured list of concepts, which shows hierarchy. The hierarchy of these concepts can be shown in a tree-structure, in which broader terms are closer to the top [1]. The concepts in a taxonomy are always connected by 'is-a' relations. All concepts must be unambiguous and mutually exclusive. For each concept, except for the top concept, there is one broader concept and none or more narrower concepts. Each concept exists only once and in one place and all relations are vertical [1]. Polyhierarchical taxonomies also exist. Here, a concept can have more than one broader term (or 'parent') or can appear more than once in the tree-structure.

2.1.3 Thesaurus

A thesaurus, just like a taxonomy, is a structured list of concepts. The concepts in a thesaurus, however, are not just connected by 'is-a' relations. In a thesaurus associative relations are also permitted. Associative relations are relations that indicate a related concept. They connect a concept to another concept in which someone who is searching for the first concept might be interested. A thesaurus can be considered as a taxonomy with extra relations.

The National Information Standards Organization, in their *Guidelines for the Construction, Format, and Management of Monolingual Thesauri* [Z39-19-1993], define a thesaurus as:
a controlled vocabulary arranged in a known order in which equivalence, hierarchical, and associative relationships among terms are clearly displayed and identified by standardized relationship indicators.

In addition, a thesaurus has the following characteristics:
- If the same concept is expressed by two or more terms, one of these is selected as the preferred term. The relationship between preferred and non-preferred terms is an equivalence relationship.
- A thesaurus is distinguished from an unstructured list of terms through use of hierarchical relationships based on degree or level of superordination and subordination of terms, where the superordinate preferred term represents a class or a whole, and the subordinate preferred terms refer to its members or parts.
- associative relationships are links between preferred terms that are semantically or conceptually associated to such an extent that links between them should be made explicit. This means that the related terms share in a way the same meaning (semantically associated), but not enough to put the two terms in the same concept, or that the related terms represent two concepts that are related (conceptually associated), like a ‘car’ and a ‘driver’.

[4]

2.1.4 Ontology

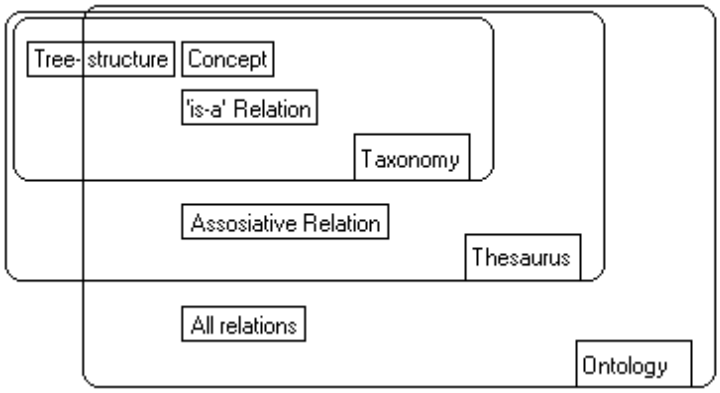
The word 'ontology' in philosophy refers to a systematic description of a minimal set of concepts that a language needs to express all its other concepts [7]. Artificial intelligence workers introduced the word into computer science, where it stands for a conceptualisation, or computer readable representation, of (a part of) the 'real world'. An implementation of an ontology is also called a semantic network.
In an ontology all words and objects from a certain knowledge domain are made into concepts. All relations between these concepts are shown. Just like a taxonomy and a thesaurus, an ontology exists of concepts and relations. An ontology, however, cannot only contain associative or 'is-a' relations, but any imaginable standardised relation, like 'is-child-of', 'works-at', 'is-author-of', or 'lives-at'. This can lead directly from one document to others written by the same person, or by others working for the same organisation as the author. Ontologies can be machine generated from good metadata. The semantic web will be built on ontologies [1].

An ontology is implemented as a datastructure. What distinguishes the ontology from the data structure is semantics: that it talks about something in the world [3]. Technically, an ontology is a thesaurus in which all possible relations are permitted. A side effect is, that the tree-structure that was visible in a thesaurus might be lost.

2.1.5 Putting it all together

The terms described above are often confused with eachother. They are also often used to describe other types of structured lists of words. This causes much confusion about the exact definition of the words. There are two other causes for this confusion. First of all different people have different ways of describing, or interpreting something. Where someone describes the technical side of a term, someone else might give a description from a more semantic point of view. Secondly, people who use a taxonomy, thesaurus, or ontology often use one that is a little different from the exact definition, because it fits their purpose better. They create some intermediate structure, but still use the same name for it.
A thesaurus, for example, might have so little associative relations, it closely resembles a taxonomy. On the other hand, so many associative relations could be allowed, that the difference between the thesaurus and an ontology is hardly perceptible. This makes clear that it is important to make good agreements on the definition and, for example, what associative relations can be used in a thesaurus.

A taxonomy, thesaurus, or ontology always belongs to a certain knowledge domain, like economy, or medicine. In every domain a word is interpreted in it's own way, so the same word in different domains can be in a different concept. There doesn't exist a single conceptualisation that covers everything. When multiple conceptualisations exist, an associative relation in a thesaurus could also refer to a related concept in a thesaurus in a different domain. Within a domain a connection can be seen between taxonomies, thesauri and ontologies. Just like a thesaurus is a taxonomy with extras, an ontology is a thesaurus with extras. So a thesaurus can be created by removing all relations from an ontology that shouldn't be in a thesaurus. Adding associative relations to a taxonomy results in a thesaurus. The image below shows similarities and differences between an ontology, a thesaurus and a taxonomy.



Relation between vocabularies

Technically, the differences between a taxonomy, thesaurus and ontology aren't that big. The biggest difference is in what these collections of concepts and relations represent and what they are used for. In the next part I shall describe what they can be used for.

2.2 Use of vocabularies

The described vocabularies can be very useful when trying to find information in a large text corpus. They can also be used to represent documents in such a way that computers can actually work with the semantics of the documents.

2.2.1 Representing Documents

With the tremendous amount of information available today, it becomes crucial to have fast and accurate retrieval systems. Traditional Information Retrieval (IR) and knowledge discovery systems that operate on a fixed set of documents, as well as the Internet search engines extract documents by finding keywords in documents [26]. The main problem of the traditional keyword-based approach to IR concerns the quantity of the results. This approach, while sometimes valuable to experts trained to search collections from a specific discipline, often returns too much information to the user to be useful [27]. Another problem concerns the quality of the retrieved information. Commonly used measures to determine the quality of the information retrieved by IR systems are called *recall* and *precision*. The recall percentage indicates what part of the documents that should have been found, have actually been retrieved. A low recall can be caused by terms used in the query that differ from the terms used in the document. It indicates that a lot of relevant and valuable information is never found in the document collection and therefore never reaches the user. The precision percentage is just the other way around. It indicates what part of the documents that has actually been retrieved, should have been found. A low precision indicates that many documents that don't fit the user's wishes are actually returned. A lot of this 'noise' distracts the user from relevant information.

Some of the reasons related to the mentioned difficulties are obvious: natural language is

fuzzy. In particular, one problem is synonymy; for example, when an article contains the word “automobile” and the query the word “car”, exact word matching fails to identify this article [28]. Another example is homonymy. This is when one word can have multiple meanings. The word ‘country’ for example can be used to indicate a nation, like Holland, but it can also appear in an article about music. A search using this word would retrieve articles about both subjects, decreasing the precision of the searchresults. This example shows that the context of the article, i.e. the topic that an article is about, can help to determine which meaning of a homonym is the correct one. Other phenomenon that can result in retrieval of irrelevant information, or omission of relevant information, is metaphorical writing. When words in an article are not used for what the literally mean, this article has a bigger chance of being retrieved while it should not have been retrieved, or vice versa.

Efficient document retrieval can be achieved by indexing. IR problems are characterized by a collection of documents and a set of users who perform queries on the collection to find a particular subset of it. This differs from database problems, for example, where the search and retrieval terms are precisely structured. In the IR context, indexing is the process of developing a document representation by assigning content descriptors or terms to the document. These terms are used in assessing the relevance of a document to a user query [29]. This way less irrelevant documents will be returned.

The content descriptors that form the representation of a document can have any desired form. An example is the vector space model (VSM) representation. In a vector space model a matrix is used to represent a corpus of documents. The row and column vectors of this matrix represent respectively the words and documents in the corpus, and each cell is a nonnegative real number intimating the degree of relevance between the i-th term and the j-th document. This matrix gives rise to a space of term vectors and document vectors, wherein computing the cosine of the angle between two document vectors approximates the semantic relevance between the two documents. During retrieval, a query is treated like a document vector [28]. This way the relevance between a query and each document can be calculated to retrieve the most relevant documents.

Although indexing is a big improvement in IR, queries are still ambiguous. Search terms still don’t retrieve documents that are indexed using synonyms. A solution to this problem can be found by indexing documents using concepts.

2.2.2 Using Concepts

Concepts can be used to represent in an unambiguous fashion the conceptual content of the documents in a documentary system and of the queries addressed to that system. The ordinary language used by the authors of documents and by users who want to retrieve these documents is in fact often ambiguous. It may be possible to express the same concept using a number of synonyms. A document indexed using one synonym would not be found on the basis of a query that uses a different expression [6].

For indexing, that is representing the conceptual content of documents, only preferred terms can be used. Documents are indexed so they can more easily be retrieved. The representation of the documents can exist of more than only concepts. Also other information about the documents, or ratings that indicate how much a concept relates to the document can be used. If a user wants to retrieve one or more documents that could be of interest to him, he can submit a number of keywords to the document retrieval system to find the indexed documents that fit these keywords best. If the user uses a non-preferred term in a query, the search-engine knows it should look for the preferred term of the related concept. This way it doesn't matter which synonym the users uses.

2.2.3 Using Vocabularies

When representing a document by a set of concepts, the problem of recognising synonyms has been solved. However, the representation doesn't contain much semantical information. If the concepts are part of a structured vocabulary, much more can be said about the document. The relations between the concepts say something about the concept and therefore also about the documents that are represented by these concepts. This way more semantical information is added to the representation of the documents. Of course an ontology can add much more semantics than a taxonomy. Therefore the different vocabularies often have different uses.

A taxonomy is often only used for indexing information. When looking at a representation of a document which exists of the concepts from a taxonomy, one can quickly determine what a document is about. Broader terms of the concepts used for a representation, i.e. concepts that appear nearer to the top of the tree-structure, give a more general description of the contents of a document and can, for example, indicate a group of documents about the same subject.

Where a taxonomy is designed to classify things, a thesaurus is designed to help you find the right words or phrases to describe what you are ultimately looking for [1]. A document in a document retrieval system can be represented by a number of concepts from a thesaurus. If someone is searching for documents about a specific topic, and uses some of these concepts as keywords for his search, the document will probably be found. Even if concepts that are related to the concepts that represent the document are used in the search, the searcher should be able to find the document. Using the semantic relations between the concepts, a user can select the concepts that describe his search best. This way a query can be formulated as detailed as possible. The concepts from a thesaurus do more than just describe what a document is about. They add some more information. If a document is represented by a number of concepts, these concepts also determine a number of related subjects. Using these relations, associations between documents can be determined. This way, a person who is looking for an article can be redirected to other articles that might be of interest of him, even if the person didn't think of an article about that subject before. Such an application is not easy to implement, but research projects like the VICORE project (see below) should make such an application possible in the near future. This example is to make clear that not only the concepts that directly

represent the document determine the semantics of the document. The concepts that are related to those concepts can also say something about the semantics of the document.

Of all types of structured vocabularies, ontologies or semantic networks offer most semantical information. They can be designed to improve communication between computers. They give a representation of a part of the real world that is essential for effective communication between electronic 'agents'. Agents are software programs that search an electronic network for information. The information is made ‘understandable’ for these agents. If documents are indexed using an ontology, such a representation of a document will also be a small ontology. Agents can analyse the concepts and their (standardised) relations. The agents can then aggregate information that they extract from different documents to provide answer to user queries or use the aggregated information in other applications [30]. For example, an agent can be set to find a location where US dollars can be ordered. One document could contain the information that US dollars are foreign currency. Another document could contain information about local banks. If the ontology contains a relation between foreign currency and banks, which indicates that banks sell foreign currency, the agent can return the address of the nearest bank. Now the user can go there to buy the dollars.

There are different kinds of applications that can use a thesaurus for retrieving concepts and its semantics. An example of such an application is a semantic publishing system. This is a system in which documents can be published and represented by their semantics. When a document is added to the system, a conceptual representation of the document will be created. This representation, or ‘fingerprint’ represents the semantics of the document. In searches by users the representation of the document will be used instead of the document when the best results are being determined. The more the search terms entered by a user correspond with the concepts of the representation, the better the chance that this is a document the user is looking for. The system can also redirect the user to related documents. Documents are related if their semantics are much alike, that is if many of the concepts that represent the one document also represent the other document and preferably appear frequently in both documents. This publishing system makes sure that documents are published in a consistent way and that users can easily find the document they are looking for.

Another example is an associative search engine like the VICORE [22] project aims to develop. This project is intended to find and visualize relations between concepts, instead of between documents. This system first represents the documents from a text corpus like in the previous example. Then the user can specify the concepts of interest. Next, the system tries to find relations between the selected concepts, identifies other concepts that link the selected concepts together and tries to formulate hypotheses that describe the discovered relations. These relations, concepts and hypotheses are then presented to the user. This way a researcher can easily gain insight in a domain that he is not specialised in, or discover relations in his own domain that he didn’t think of before.

There are many ways to use a structured vocabulary. A lot of research is being done on extracting and representing knowledge. The most common vocabulary to do this, is the

thesaurus, so there is a general need for rich thesauri. The methodology I am trying to develop is primarily ment to enrich general thesauri for a predetermined purpose. Therefore I shall from now on write about thesauri instead of vocabularies, keeping in mind that most information also applies to taxonomies and ontologies.

2.3 Construction Issues

As explained above, there are good reasons to use a thesaurus, but to do so, it has to be constructed first. Thesaurus construction requires collecting a set of terms from a specified domain and determining relations between these terms. Some of these will end up becoming preferred terms and others may not appear in the thesaurus at all in their original form, but they may bring to mind concepts that need to be in the thesaurus [12]. Sources from which terms can be collected are wordlists, like other vocabularies or dictionaries, people, such as subject specialists, or a large text corpus from which terms can be extracted. If possible words in a thesaurus should be nouns. These terms should be general enough so they can be used to index a number of documents, but not so general that they can be used to index too many documents [12]. There are a lot of reasons why a certain term should or shouldn't be in the thesaurus. When using a corpus, this can, for example, be determined by looking at how often a term appears in the corpus. When a term collection is complete, or seems complete, since there is no way to be certain, synonyms have to be found and a hierarchy of the terms should be constructed. Next, the terms should be connected by semantic relations like, narrower/broader term (hierarchical), preferred/non-preferred term (equivalence), and related to (semantic). Related terms can be found by considering the following categories: Time, Place, Product, Cause, Agent, Device, Application, Part, and Complement [12]. When considering the product category, for example, one can think of a concept that creates, or is created by the concept it could be related to, like photograph and camera, the place category could bring ship and harbour to mind, or the complement category could link the concepts parents and children.

When a new thesaurus is going to be built, an important question that arises is which methodology to use in the development process. When determining the answer to this question, some other questions have to be taken into account, like:

- should the thesaurus be built from scratch or by reusing other thesauri,
- which activities must be performed when building a thesaurus with the selected methodology,
- what is the life-cycle of the thesaurus,
- how should maintenance be done,
- which tools support the development process,
- how will the thesaurus be stored,
- which language should be used to implement the thesaurus [11],
- should the thesaurus be constructed automatically or by hand?

The answer to some of these questions can be dependent on the application the thesaurus will be used for, the available resources and technology, and the preference of the developer. In the next chapter I shall describe and compare some existing methodologies

for thesaurus construction. I shall also determine which practical questions have to be answered when creating a new methodology.

3. Methodologies for Thesaurus Construction

One important difference between a technical field that is in its "infancy" and another that has reached "adulthood" is that the mature field has widely accepted methodologies, while the emerging discipline usually does not [14]. Since thesaurus construction is a relatively new research area, there are as yet no standardised methodologies for building thesauri, but a series of approaches have been reported. The first guidelines for a methodology were proposed in 1995 in [15]. Later several methodologies have been published. A summary and analysis of the best-known methodologies is given in [14].

3.1 Existing Methodologies

Some well-known methodologies are developed by Uschold and King [15], Grüninger and Fox [16], Bernaras [17], Gómez-Pérez [18], and Swartout [19]. The Uschold and King's method gives a very general description of what a methodology should be like. It describes what should be done, but not how it should be done. This is because the details of a construction process depend too much on the intended use of the thesaurus. Uschold and King give a good basis for anyone who wants to develop a new methodology for thesaurus construction. A Methodology according to Uschold and King uses the following steps:

- First it identifies the purpose of the thesaurus, because it is important to be clear why the ontology is being built and what its intended uses are [14]. This is important, because the choices that are made in the first two steps of the building process rely on this. A thesaurus can for example be intended for some manner of re-use, or the construction can depend on the nature of the software with which the thesaurus will be used.
- Next, the building process starts. This process consists of three parts:
  1. In the first part the knowledge is captured. Here the key concepts and relationships in the domain of interest will be identified and will be given an unambiguous name and description.
  2. Next, the collected information will be coded. This involves the explicit representing of the acquired information in a formal language.
  3. The third part takes place during either or both of the previous two parts, and deals with the question how existing ontologies can be integrated in the ontology that is being built.
- Finally the ontology will be:
  - evaluated, to check if the result matches the specifications,
  - and documented, which is important for use and maintenance of the ontology.

The methodology by Grüninger and Fox is a methodology that is based on the development of knowledge-based systems using first order logic [11] and essentially involves building a logical model of the knowledge that is to be specified by means of the

ontology. The development of ontologies is often motivated by scenarios that arise in the applications [16]. The meaning of the words in the ontology, and therefore their location in the ontology, largely depends on the applications. First these scenarios are captured. A scenario is a description of a problem that should be solved by using a software application. Every scenario should also contain a solution to the problem. From this solution a set of informal intended semantics for the objects and relations, that will later be included in the ontology, can be extracted. The scenario gives a motivation to use these objects and relations in the ontology. A motivating scenario can, for example, be a user's whish to determine how many dollars he can buy for a certain amount of euros. The answer to this question is "by combining the amount of euros with the exchange rate between these currencies." From this answer can be concluded that the concepts 'Currency' and 'Exchange rate' should be part of the thesaurus, and should have a relation that connects different kinds of currencies to their exchange rates. Given these motivating scenarios, a set of queries will arise which place demands on the ontology. These queries are the encoded solutions to the scenarios and need to be supported by the ontology. They represent a part of the functionality of the application and can be considered requirements, in the form of questions that an ontology must be able to answer [16]. These questions are called competency questions. The competency questions serve as constraints on what the ontology can be. The questions are used to evaluate the ontological commitments, the assumptions that have been made to translate reality into an ontology, to see whether the ontology meets the requirements. An ontology must be able to represent these questions using its terminology [14]. This final step validates the ontology. In the example above the ontology should be able to be queried for two currencies and their exchange rate. The competency questions formed by all motivating scenarios form the requirements for the ontology.

The approach of Bernaras was proposed at the KACTUS project [34]. This project investigates among others feasibility of knowledge reuse in complex technical systems and the role of ontologies to support it [35]. The methodology is conditioned by application development. Every time an application is built, the ontology that represents the knowledge required for the application is built. This ontology can be developed by reusing others and can also be integrated into the ontologies of later applications [14]. The building process is repeated each time an application is developed and involves the following steps:

- For each application the concepts that the application tries to model are collected.
- Then the ontologies that have been built for earlier applications are generalised and the top-level parts that can be used for the new ontology are collected.
- Finally, these parts are put together and will be refined and specified until all concepts used by the application are included.

The METHONTOLOGY approach is a methodology for building ontologies either from scratch, reusing other ontologies as they are, or by a process of reengineering them [11]. This methodology describes the following phases:

- A construction phase in which first the specifications are given, based on for what and by whom the ontology will be used.
- Then domain-specific terms will be collected and structured into concepts.

- Next, these concepts will be formalized. This means that the concepts will be translated to a formal model. This model should be clear to the human programmers, so they can implement these models in a computational language, so the models become machine-readable.
- Finally, maintenance updates and corrects the ontology. The life cycle of the methodology is based on evolving prototypes. This means that the different phases of this process will be repeated until the ontology is complete enough to be used.

Besides the construction phase, the methontology approach describes other management and support activities that must be performed while building the ontology, like scheduling, control, quality assurance, knowledge acquisition, integration, evaluation documentation and configuration management.

Swartout created a methodology based on SENSUS. The first step is to take a series of 'seed' terms that are important to the ontology under construction. These terms should come from a large, general ontology, like SENSUS (which has over 70.000 concepts.) Then all the concepts in the path from the seed terms to the root of SENSUS are included. Terms that could be relevant within the domain and have not yet appeared in the ontology are added. Finally, for those nodes that have a large number of paths through them, the entire subtree under the node is sometimes added, based on the idea that if many of the nodes in a subtree have been found to be relevant, then the other nodes in the subtree are likely to be relevant as well. Obviously, very high-level (general) nodes in the ontology will always have many paths through them, but it is hardly ever appropriate to include the entire subtrees under these nodes [14]. In short, pieces of a large, general ontology are put together. The result will be pruned and enriched.

3.2 Evaluation

Some methodologies are somewhat alike, or are in some phases equal to others, but there are also some important differences. Some used techniques are complementary; some are even eachothers opposite. As a result of this, some choices have to be made when creating a methodology. Here, I shall discuss some differences and choices to be made.

3.2.1 Comparing Existing Methodologies

Methodologies broadly divide into those that are stage-based and those that rely on iterative evolving prototypes. These are in fact complementary techniques [13]. Stage based methodologies work step by step. The thesaurus construction is divided into several subtasks. Each subtask starts after the previous task has been finished. After the final task, the thesaurus is ready. Iterative methodologies perform the same series of tasks over and over again. After each iteration a new prototype thesaurus is ready. These iterations go on until the thesaurus is good enough to be used. Most methodologies distinguish between an informal stage, where the ontology is sketched out using either natural language descriptions or some diagram technique, and a formal stage where the

ontology is encoded in a formal knowledge representation language that is machine computable [13].

When searching for different terms or concepts, this can be done in at least four different orders. One can start with the most general concept and work down from there, by looking for more specific concepts. This is known as the top-down approach. This can also be done in exact the opposite order. The most specific concepts that are needed for the thesaurus can be searched first. Then more general can concepts can be looked for, until all branches meet in one most general concept. This approach is called bottom-up. The other two approaches are variations on top-down and bottom-up. They are also opposite and are called inside-out and outside-in. Inside-out starts somewhere in the middle, and works its way to the most general and the most specific concepts, and outside-in starts at the most general and the most specific concepts and goes on, until the two meet somewhere in the middle.

An important decision when creating a methodology is if the thesaurus should be constructed from scratch or by reusing one or more other thesauri. Reusing other thesauri has as main advantage that a lot of work is already done. On the other hand, when depending too much on other thesauri, it can be harder to meet the specifications of the thesaurus that is being constructed. The extending and pruning of the used thesauri can also be a lot of work. In general, it is better to construct an application dependent thesaurus from scratch, so you can better meet the requirements of the application. The two approaches can also be combined. A general framework can for example be constructed from scratch, and can be filled in by using other thesauri, or vice-versa. Another critical argument to determine if the thesaurus should be constructed from scratch or not, is the availability of other thesauri. If no other thesaurus on the selected domain exist, it is obvious that they can't be used to construct a new one. But also the quality and reusability of existing thesauri play in important role in this decision.

The final step in each methodology is the evaluation of the thesaurus and the methodology itself. In this phase shall be tested if the right words have been added to the right concepts and if the concepts have been placed in the right location in the thesaurus. The resulting thesaurus should be able to meet the demands placed on it, like enabling an application to give the right results. In "Towards a protocol for validating thesauri and ontologies" [35] a guideline is given to come to such an evaluation. It describes the key steps to be performed in order to enrich knowledge structures, like thesauri and ontologies, in an efficient, effective and easy-to-do way such that the resulting knowledge structures can be considered as valid.

3.2.2 Practical choices to be made

Clearly, a lot of choices have to be made before a thesaurus can be constructed. Most choices rely on the developer's preferences and the available resources. However, these choices can also be influenced by the application in which the thesaurus will be used. When the future use of the thesaurus has a lot of influence on the methodology, this

methodology is application dependent. This approach is often used if the thesaurus is only ment to be used by one, or one type of, application. In this case the thesaurus can be constructed in such a way that it meets the specifications set by the application, and that it's easy to use by the application. If a thesaurus will be used by multiple, quite different applications, or if the developers do not know the use of the thesaurus at construction time, the thesaurus construction will probably be application independent.

Several aspects of an application can influence a number of stages of an application dependent methodology. The questions that an application must be able to answer can be considered when creating concepts, or when implementing the thesaurus in a computational language. Concepts aren't just lists of synonyms. They are lists of terms that are synonyms in the scope of the posed question, e.g. the terms have the same meaning in the used context. So, when the type of questions is known in advance, this information can be used when conceptualising the collected knowledge. In the same way the information can be used to determine other than hierarchical relations and the type of semantic relations that need to be used. This way the results of the application will fit the user's wishes better. Using this approach, the application is actually used to specify the domain of the thesaurus. The way in which the questions are posed, determined by the user-interface, can be of influence on the implementation of the thesaurus. It could influence the way in which the thesaurus is represented, since the representation can be adapted to possible queries. This way, queries can be made as small and efficient as possible. In order to adapt to the questions that the application must be able to answer, use-cases for the application should be made. This way most possible questions can be formulated from a user's point of view.

The way in which terms for the thesaurus will be collected can also be influenced by an application. If the thesaurus is being designed to be used by one application, only the terms that are relevant to the application need to be in the thesaurus. So, if first the terms that will probably be used by the application, or the users of the application can be collected, the rest of the thesaurus can be constructed by a process of abstraction. This is the bottom-up approach. This works best if the domain in which the terms are sought is relatively small. Otherwise it will be quite hard to collect all required terms.

The question if a thesaurus should be considered from scratch or by reusing other thesauri, is a question that can be answered by combining the specifications of the application with the available thesauri. If the quality of existing thesauri is very poor, and the application requires a rich and correct thesaurus, it is obviously not a good idea to reuse the existing ones. On the other hand, if only a global thesaurus is required, or a 'backbone' that will be enriched later, even a low quality thesauri can be useful.

Finally, the performance of the application should be kept in mind when constructing a thesaurus. A very deep and rich thesaurus will take a lot more time to process then a simple one, so a thesaurus shouldn't have more concepts then is required for the application. The same goes for the number of semantic relations between the concepts. A thesaurus with many relations will give a better result, but will make the application slower.

Another choice to be made is the question if the thesaurus should be constructed by hand or automatically. The future use of the thesaurus isn't of any influence here, so other arguments have to be found. Collecting terms and constructing the thesaurus is quite a time-consuming and subjective job, especially when the construction is done manually. When constructing a thesaurus manually, the human developer can easily forget a lot of terms, synonyms and relations. The result is subjective, so the thesaurus would probably differ if it was constructed by another developer and, as mentioned before, the manual way of constructing a thesaurus is quite a lot of work. Unfortunately there is a bigger problem. Terminology in a research domain changes and expands from time to time. Therefore the thesaurus should also be expanded. When this happens, the whole process can start over again. This time only terms that aren't in the thesaurus yet should be looked for. So not only the construction of a thesaurus is a lot of work, but also keeping it up to date.

Thesaurus construction could be much faster and less subjective, if the construction could be fully automatic. Computer programs could scan text, to find the right terms and relations. Other programs could check if the thesaurus is good enough. Once the thesaurus is complete it could also be kept up to date automatically. When new documents have to be indexed, a computer program could scan them and find new terms and place them in the thesaurus. This approach seems too good to be true and unfortunately, at this moment, it is. The problem is that computers don't understand human language. This is actually why we need a thesaurus in the first place. A thesaurus that has been constructed using a fully automated process can't be trusted as much as thesaurus created manually by domain experts.

3.2.3 A Generalization of Existing Methodologies

To use the advantages of both automated (speed and accuracy of computers) and manual (semantic knowledge) construction, for now a semi-automated construction process in which the domain experts are aided by computer programs seems the best option. Computers will do work like parsing and representing documents, creating statistics, finding new terms and, if possible, giving suggestions about what new concepts are, to what other concepts they are related and where they should be placed in the thesaurus. The human domain experts will be there to check the computers results and make the final decisions.

No matter what approaches are used and what choices are made when developing a thesaurus, in general seven stages can be identified that can be part of the life-cycle of a methodology. These stages are:

1. Identifying the purpose and scope of the thesaurus. Here the specifications are determined
2. Terms acquisition, which is the process where terms from the selected knowledge domain are collected
3. Conceptualisation, where the terms are translated into concepts and relations between

the concepts are identified
4. Integrating, which stands for combining the concepts with existing thesauri
5. Encoding, where the thesaurus will be implemented
6. Documentation, which is essential for use and re-use of the thesaurus
7. Evaluation, where the resulting thesaurus is compared with the specifications from the first stage [14].

Not all of these stages have to be in a good methodology. The ‘integrating’ stage, for example, shouldn’t be in a methodology if no existing thesauri are used. These stages are the most commonly used phases from a thesaurus construction methodology.

4. Choosing an Approach

There are a lot of different ways to construct a thesaurus. From all these approaches, we have to choose the approach that fits our purpose best. This can be an entire existing methodology, or use pieces of different approaches. Before deciding how the thesaurus should be created, we should first determine why is should be created.

The main reason for the construction of the thesaurus is to analyse the semantics of documents and create a compact representation of these documents. Based on such a representation a computer should be able to calculate what a document is about and what it's related topics are. If a document is represented by a number of concepts, instead of a number of keywords, these concepts not only indicate what the article is about, but also a number of related subjects can be determined. Using these relations, associations between documents and concepts can be determined. As described in section 2.2, there are many forms that an application that uses a thesaurus can have. Therefore it wouldn't be wise to make the thesaurus too application dependent. It is more important to create a thesaurus that can be used as a building block for multiple applications and can be used to determine which documents, or concepts, are closely related to the concepts a user might be searching for. In this case the quality of the thesaurus is more important then making sure applications can use it fast and easy enough. Therefore, it would seem wise to construct a thesaurus where future applications of the thesaurus won't be taken into account in the construction process. The result of such a building process would be an independent thesaurus, representing the economic domain, where any kind of application can be built on. On the other hand, the future applications of a thesaurus are too important too ignore. An application can be seen as the context of a thesaurus. The exact meaning, and therefore also the location, of each concept in the thesaurus is determined by this context. A better idea is to start with a general thesaurus in the economic domain. This general thesaurus can then be enriched to create a more specific thesaurus, which is dedicated to a specified application.

Since constructing a thesaurus from scratch is a lot of work, the easiest way to create a new thesaurus is by reusing other thesauri. Unfortunately there aren't many good thesauri in the economic domain. There are however some thesauri, like the EUROVOC [20], or the JEL classification system [21], that cover the economic domain good enough to form a solid base for a new thesaurus. This thesaurus can then be refined, until it becomes specific enough to index or represent the literature in the specified domain.

Since computers are still unable to perfectly understand the semantics of the human language, it is for now hardly possible to create a reliable fully automated process for thesaurus construction. Ironically, a good thesaurus is needed to make this possible. A manual methodology, on the other hand, is obviously a lot of work, subjective and quite sensible to human mistakes. This is why the refinement process will become a computer-aided process in which we will enrich the thesaurus.

An automated process shall try to find new concepts in a corpus containing economic literature. By storing as much semantic information as possible, this process shall try to indicate new concepts, and discover relations between existing and new concepts. Then it shall use this information to suggest a place in the thesaurus for each new concept. These suggestions, including all used information, shall be presented to a domain expert. The human expert shall then make the final decision about if and where the new information shall be added to the thesaurus.

In short, we will first reuse other thesauri into a general economic thesaurus and then improve the result using a semi-automated enrichment process.

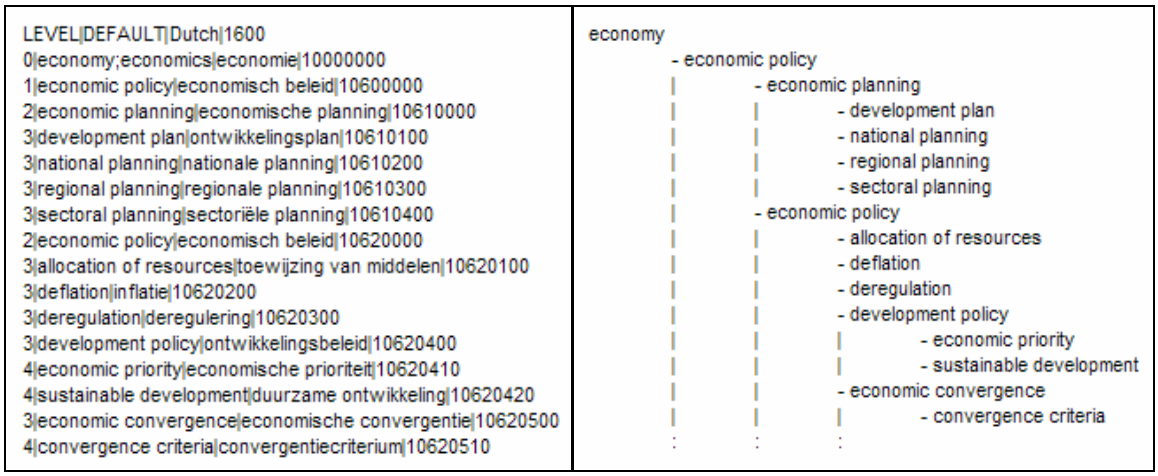
4.1 Preparations

Before an enrichment process can be constructed, several preparations have to be made. First of all we need a thesaurus to start with. This thesaurus should give a good, but not perfect, coverage of the economic domain. The coverage should not be perfect, because otherwise the thesaurus could not be enriched anymore. To find new concepts, we need a text corpus, containing economic literature. These documents should be quite general. Otherwise it won't be possible to give a good representation of the documents using the general thesaurus. A good representation is needed, because otherwise the context of new concepts, i.e. existing concepts surrounding the new concepts, will be too shallow. An extensive and representative context of new concepts is needed to find a good location for the new concepts in the thesaurus. Finally a developing environment is needed to develop the enrichment process in. With this environment I should be able to represent documents, find new concepts and search for a location in the thesaurus where the new concepts can be placed.

4.1.1 A General Thesaurus

As mentioned above, to develop an enrichment process, we first need a general thesaurus to start with. This thesaurus should give a good coverage of the economic domain, so documents can adequately be represented. As starting thesaurus, I selected the EUROVOC thesaurus [20]. I selected this thesaurus because it seems to be the most extensive thesaurus that covers the economic domain. For the enrichment process I will be using software from Collexis [31] (see chapter 4.1.3), therefore the thesaurus must be formatted in a way that is readable for this software.

Collexis uses the Pipe Separated Format (PSF) to represent a thesaurus. A thesaurus in this format is stored in a plain text document. Each line, except for the header, represents a single concept and is divided into multiple columns. Columns will be separated by a pipe (the '|' sign). The figures below show a part of a thesaurus in PSF and in a tree structure.



PSF format **Tree structure**

The header of a PSF file consists of the keyword “LEVEL”, followed by the names of the languages used in each column, and finally a number. The first language should be represented by the keyword “DEFAULT”. This column represents the default language of the thesaurus.

The first column indicates the depth (level) of the concept in the thesaurus. The lines must be ordered in a depth-first order. Concepts with a low level are general terms that appear near the top of the thesaurus. Those with a high level are more specific terms and appear more at the bottom of the thesaurus. In the picture above can be seen how a thesaurus in PSF format can be shown in a tree structure. The final column of the thesaurus contains the conceptnumber of the concept on that line. This conceptnumber uniquely represents the concept represented by that line. If two rows contain the same conceptnumber, then the two rows are treated as the same concept. The header of this row indicates the highest conceptnumber that can be used.

The rest of the columns all contain a list of synonyms in a certain language. The first language-column, that is the second column of the PSF file, should be the default language. The second language-column can, for example, contain a list of synonyms in Dutch, while the third could contain the same synonyms in the German language. The synonyms, of which the concept consists, are separated by a semicolon. The first synonym will be the preferred term, and will therefore be used to represent the concept. Note that the synonym will not be used to uniquely identify the concept. For this, the conceptnumber will be used. The economical branch of the used thesaurus in PSF format is available in appendix A.

There are two other lists of words that in a certain way belong to the thesaurus. These are the stopword list and the remove-word list. The stopword list contains words that are ignored by Collexis. When creating a fingerprint, the words on this list will never appear in it. Neither individually, nor as part of another concept. Words on the remove-word list won’t be ignored. However, these words won’t appear as individual words in a fingerprint. These words are only used if they are part of a concept. For example, a

thesaurus contains the concept ‘new economy’. When creating a fingerprint of the sentence “The new economy is new to me”, the concept ‘new economy’ would be in the fingerprint if there is no stopword list and no remove-word list. However, if the word ‘new’ is on the stopword list, the concept won’t be in the thesaurus. This is because the word ‘new’ is ignored; so only “The economy is to me” remains. If the word ‘new’ is on the remove-word list, the concept will be recognized, because ‘new’ is part of a concept. The second appearance of ‘new’ will be ignored, because this time the word isn’t part of a concept.

4.1.2 An Economic Corpus

In order to find new concepts, we need a corpus of documents about an economic subject. These documents will be searched for words that aren’t part of the thesaurus. Using the words from the documents that are already part of the thesaurus (the context of the new words), a possible location for the new word in the thesaurus will be determined.

To find and download these documents from the Internet, I used the Google Web API [32]. This API allows programmers to use Goole’s functionality in their programs. The program I wrote and used to search for, and download the documents is included in appendix B. The program is written in the Perl programming language. To find documents about economic subjects, I used this program with searchterms like “economy”, or “economic news”.

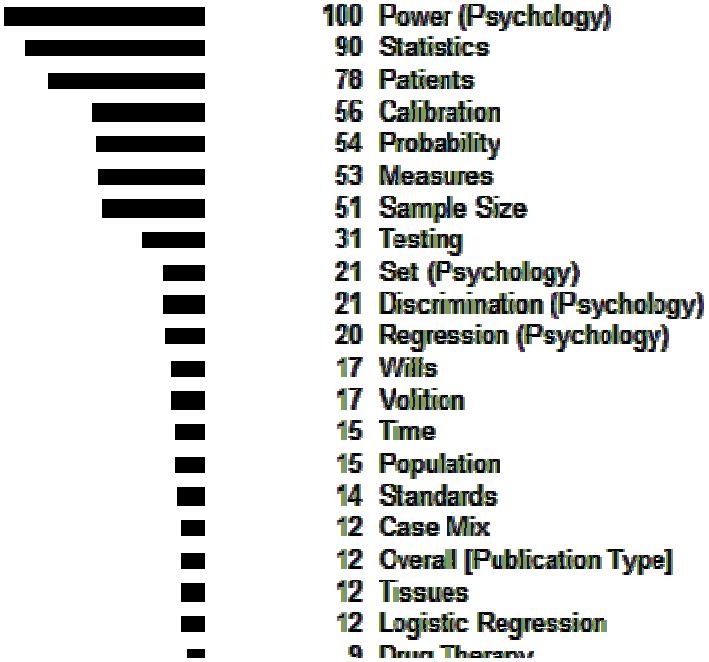
Not all documents returned by the program were good enough to be used. Some documents hardly contain any information. They only contain headlines, or links to other sites. Others do contain enough information, but not on the right subject. After separating the documents about an economical subject from the rest, a corpus of useful documents remains.

4.1.3 A Development Environment

To develop an enrichment process, a development environment is required, that meets certain demands. First of all it must be able to handle thesauri. The environment must be able to use a thesaurus to find concepts from the thesaurus in documents and find words in documents that don’t exist in the thesaurus. There also has to be a possibility to add new concepts to the thesaurus. Secondly, the environment must be able to handle documents. It must be able to store and search documents and represent them using the concepts from a thesaurus.

A development environment that has all these features and more is the Collexis ® Engine from Collexis B.V. [31]. It also has an API, so programmers can use Collexis’ features in their own programs. Collexis is based on the principle of fingerprinting. A fingerprint is a small and unique representation of a text. Collexis uses a thesaurus to find keywords in a text. It exploits the used synonyms to recognize the keywords in the text and to estimate

the relevance of the keywords for denoting that text. A series of keywords with their relative weights, together representing a text, are referred to as a fingerprint [33]. The system can create a fingerprint for each piece of text that contains relevant information, such as competence sheets, project descriptions or web pages.



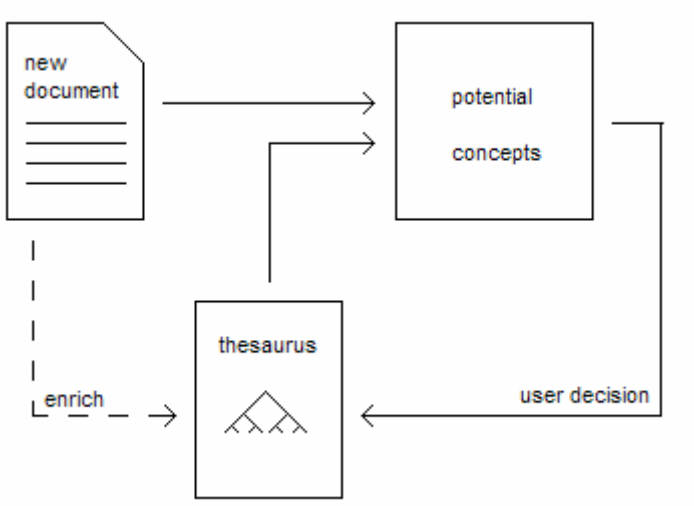
A fingerprint of a psychological document

The fingerprinting process makes use of a structure of professional terminology of a particular field (essentially a thesaurus). By doing so it embodies the way humans understand those terms and concepts. Collexis fingerprints, like human ones, are a very small but still unique representations of their source [31]. When creating a fingerprint from a text, it doesn't matter if the text is large or small. Even from very small texts, like a query by a user, a fingerprint can be made. A Collexis catalog contains only fingerprints, not the original information. This makes the process of matching a catalog with a fingerprint that represents a user query extremely fast and the results very relevant. Entire documents can also be used as a query. In fact, using a document as a search command (asking for 'more like this') will define the search topics much more accurately than one or two keywords and it will yield better results [31].

4.2 Refining the Thesaurus

With all the requirements ready, we can try to enrich the thesaurus. I've developed a semi-automated enrichment process to add new concepts to the thesaurus. This process first searches a document for words that could possibly be a new concept in the thesaurus. I shall refer to these words as *potential concepts*. Next the process will give suggestions on where these potential concepts might be placed in the thesaurus. The suggestions are

concepts from the thesaurus which might be a parent, child or synonym of the potential concept, i.e. the suggestions are concepts that should be closely related to the suggested concept. I shall refer to these suggestions as *suggested concepts*. This process is illustrated in the figure below.



The enrichment process

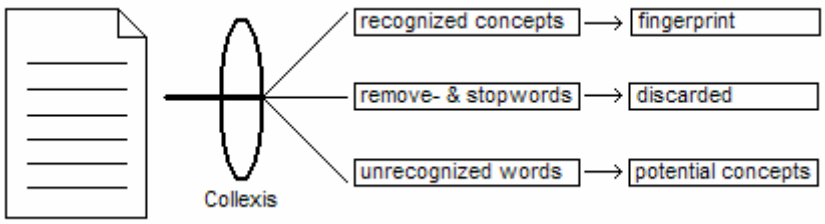
The enrichment process is based on the Collexis engine. The process uses Collexis' feature to compare the fingerprint of a (piece of a) document with a large collection (spelled as 'collexion' by Collexis) of fingerprints stored in the database of Collexis. This collection of fingerprints is obtained by indexing large set of documents. These documents, which have been collected in chapter 4.1.2, should give a good coverage of a certain domain. In this case the domain is the economic domain.

4.2.1 Finding potential concepts

The first step in the enrichment process is to find potential concepts in a document. These potential concepts should be words that can not (yet) be identified by the thesaurus. Here Collexis is quite helpful. A fingerprint of a piece of text can be created in a number of different formats. Each format gives a certain amount of information. One of these formats has an option to return the entire text, in which each word has been prefixed. There are three different prefixes, which put a word in one of three categories.

1) If word (or group of words) has been recognized as a concept from the thesaurus, the word is replaced by its conceptnumber. The concept is prefixed by the '^' sign. The sign indicates that the word (represented by its number) that follows this sign is a concept from the thesaurus.
2) The '-' sign in front of a word also indicates that the word it is in front of has been recognized. This word, however, is not recognized as a concept, but as a remove- or stopword.
3) The '+' sign indicates that the word followed by it is not recognized. The word is

neither in the thesaurus, nor on the remove- or stopwords list.



Collexis splits a document in three categories

This representation of the original text is called *unexplained text*. Each word is marked as a concept, a remove- or stopword, or an unrecognized word. For the sentence "In today's economy, a lot of money is spent" the result will be:

```
-in +today +s ^9240 -a +lot -of ^9745 -is +spent
```

Here the recognized concepts 'economy' and 'money' are replaced by their conceptnumbers 9240 and 9745. The words 'in', 'a', 'of', and 'is' are marked as a stopword. In this example the words 'today', 'lot', and 'spent' are not recognized as an existing concept, a remove-word, or a stopword. The recognized concepts and their score form the fingerprint of the document. All unrecognized words can be proposed as potential concepts, however, to reduce the amount of not very useful potential concepts it could be better to use only a selection of the unrecognized words as potential concepts. This selection can for example be based on the number of times a word is used in a document. Unfortunately, this way only potential concepts that exist of one word can be recognized.

In order to retrieve this marked result, the option to return unexplained text must be set when loading the thesaurus into Collexis. This option can be set in the thesaurus' settings file, called the 'ika.ini' file. Next, the document should be indexed using the most verbose output (output format 4). The unexplained text will now appear in the fingerprint.

Once these potential concepts have been retrieved, they can be offered to the enrichment process. A decision has to be made if the words will be added to the thesaurus or not. And if a word will be added to the thesaurus, a location for the new concept has to be determined.

4.2.2 Finding suggestions

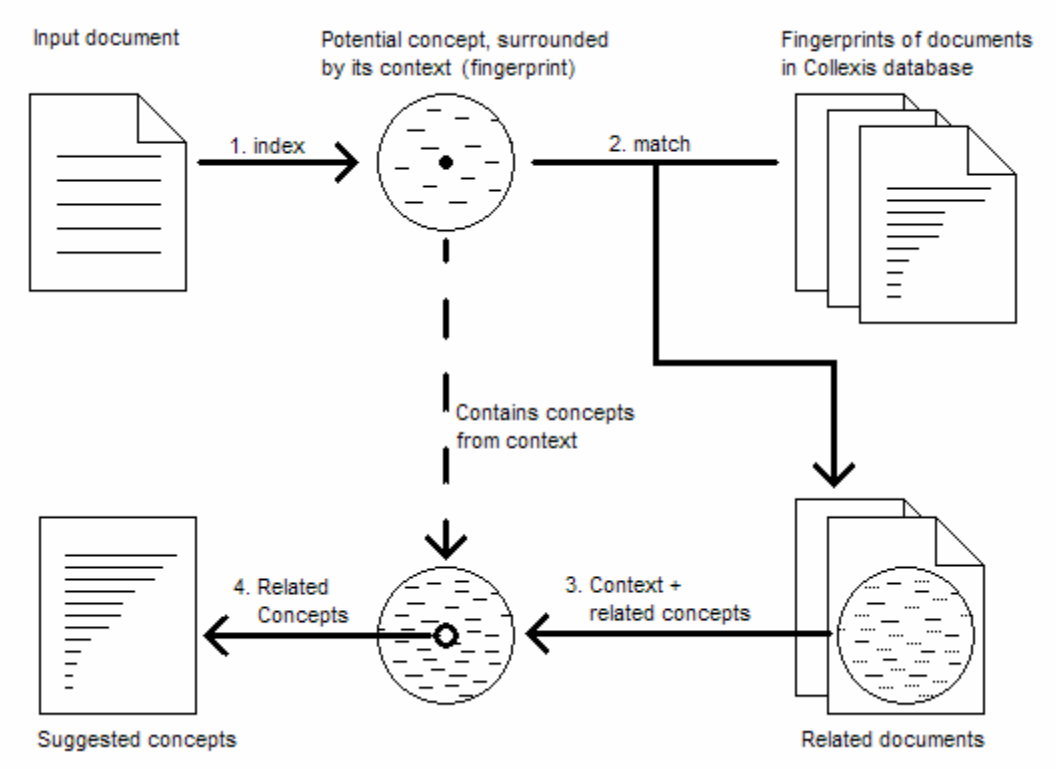
Once a word has been offered to the enrichment process, a decision has to made be about if the word should be added to the thesaurus. If the word shouldn't be added to the thesaurus, it can be added to the remove- or stopword list, or it can be ignored. But if the potential concept is a word that should be in the thesaurus, a location in the thesaurus should be determined where the concept can be added. At the moment, this decision can't

be made by a computer, but has to be made by one or more domain experts. This enrichment process will try to make it easier for the human expert, by giving some suggestions. In this process the suggested concepts will be based on concepts that are already in the thesaurus and are found near the potential concept in the document. I refer to the collection of concepts that appear near the potential concept as the *context* of the potential concept. For now, I shall use all concepts that appear in the same sentence as the potential concept as the context of the potential concept. This method for finding suggested concepts is based on the idea that concepts that are related to the concepts that are found near the potential concept, could very well be related to the potential concept.

To find suggested concepts for a potential concept, first the context of the potential concept is represented as a fingerprint (the first step in the image below). This fingerprint consists of all recognized concepts that appear in the same sentence as the potential concept, and their relative weights. It is necessary to turn the context into a fingerprint, because otherwise Collexis wouldn't be able to do handle this context. This context fingerprint will be merged with a very small fingerprint, which consists only of the potential concept. Since the potential concept is, by definition, not recognized by the regular thesaurus, the second fingerprint is constructed indexing the potential concept with the 'freetext' thesaurus. This thesaurus is no real thesaurus, but treats each word that isn't a remove- or stopword as a concept. The result of using this fingerprint is, that if the potential concept exists in other documents that have been indexed, concepts that are in the same document as the potential concept will have a better chance of being selected as suggested concept.

Obviously, the created fingerprint might contain some very general concepts (indicated by a high weight), and some irrelevant concepts (indicated by a low weight). These concepts do not represent the potential concept very well. Therefore they shall be removed from the fingerprint. This way, the context is a better representation for the potential concept. There is no concrete measure to calculate how well a concept can be used to represent the potential concept. In Collexis, weights are always normalized, so the highest weight is always one, but the lowest weight can be as close to zero as possible. Therefore, the 'cut-off' value for which concepts remain in the context and which are removed is quite arbitrary.

The next step is to use the fingerprint of the context of the potential concept in a search through the fingerprints that have been stored in Collexis (step two in the image below). By using this fingerprint to search for documents, documents will be retrieved that have a fingerprint that is similar to the fingerprint of the context of the potential concept. This means that the retrieved (related) documents contain a pattern of concepts which is similar to the context of the potential concept.



The associations based enrichment process

From the related documents, Collexis can retrieve related concepts. In the image above this is indicated by the arrow from the lower right of the image to the lower center. These concepts are concepts that exist in the fingerprints of the related documents, but not in the fingerprint of the context of the related concept. For example, a sentence from the input document could be "Because economics is all about money, the value of the dollar is quite important for the American economy." Concepts recognized by the thesaurus are underlined. The first word from this sentence that is not in the thesaurus and should not be in the stopwordslist is 'money'. Therefore 'money' is the first potential concept. The context of this potential concept exists of the concepts 'economics', 'value', 'dollar', 'American' and 'economy'. Now, a document that has already been indexed in the Collexis database, contains the sentence "The economic welfare of the United States depends on what their currency, the dollar, is worth." Again, concepts from the thesaurus are underlined. The words 'economics', 'economy' and 'economic' are words from the same concept and so are the words 'value' and 'worth'. The words 'American' and 'United States' are also from the same concept. The fingerprint from the second sentence closely matches the context from the potential concept, because all concepts that appear in the context also appear in the fingerprint. As a result, in step two in the image above, the document that contains the second sentence is likely to be retrieved as a related document. The only concept that does appear in the fingerprint of the second sentence, but not in the context of 'money' is the concept 'currency', so this concept could be one of the related concepts that are retrieved from the related documents.

Finally, the concepts that are most closely related to the related documents will be used

by the enrichment process as suggested concepts and will be presented to the user. In the example above, the concept ‘currency’ has a big chance of being one of the suggested concepts for the potential concept ‘money’. This final step is indicated by the arrow from the lower center to the lower right in the image above.

4.2.3 Making a decision

Now we have a proposed concept and a list of suggested concepts. These concepts will be presented to one or more experts on the specified domain. All the expert has to do, is indicate what has to be done with the proposed concept. If he thinks the proposed concept should not be in the thesaurus, there are two options. The proposed concept can be added to the stopwords list, or it can be added to the removewords list. One should be careful before adding a word to the stopwords list, because words on this list will be ignored by Collexis in the future. Of course it is also possible to do nothing with the proposed concept, but this does not seem a good option. If a proposed concept is not good enough to be added to the thesaurus now, it probably won’t be in the future. In this case, the removewords list is probably the best option, because, although the word by itself is not a good enough concept for this thesaurus, it might be good enough in combination with another word. If the proposed concept would end up in the stopwords list, the combination can no longer be added as a concept.

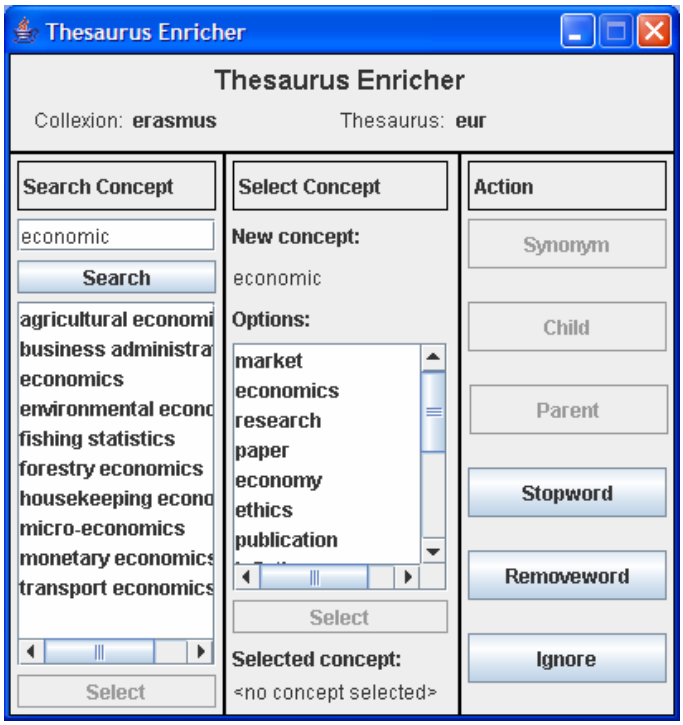
If the proposed concept is a concept that should be added to the thesaurus, the expert should select a location for this concept. This can be done by selecting a concept from the thesaurus and selecting the relation between the selected and the suggested concept. When selecting a concept, the expert is not restricted to using the suggested concepts. He can also choose to search for a concept in the thesaurus himself. I shall refer to the concept selected by the expert as the *selected concept*. Once a concept has been selected, a relation between the selected and the suggested concept can be indicated. The suggested concept can be a child, parent, or synonym of the selected concept. Again the expert can choose to ignore the suggested concept. This seems only an option if the suggested concept should be in the thesaurus, but no suitable location can be found for it as yet.

All choices made by the experts shall be logged. When all proposed concepts have been handled by all experts, the proposed concepts choices can be analysed. If most experts have made the same choice for a proposed concept, the concept can be added to the thesaurus, removeword-, or stopwordlist. If the experts disagree, some discussion might be required, before making the final decision. This indicates that the human part of the enrichment process still is more important than the automated part of the process. The part done by computers is only to assist thesaurus builders and domain experts in making their choices.

5. Experimental

5.1 Testresults from the enrichment process

To test the enrichment process described above, I implemented this process in a Java application. The input parameters of this application are the used collexion and thesaurus in Collexis and the document in which the potential concepts will be searched.



The thesaurus enrichment tool

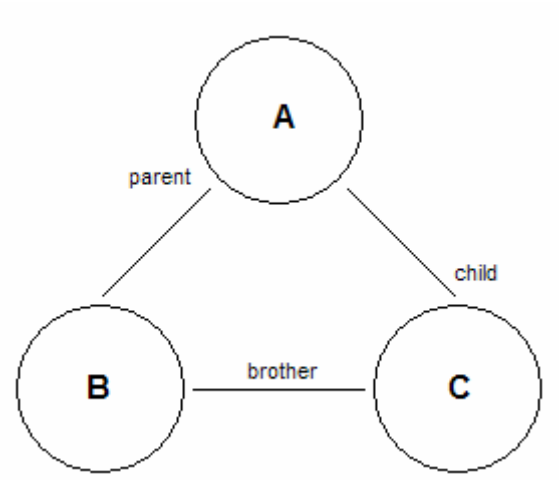
The image above is a screenshot of the used application. Each potential concept is offered to the user in the column in the middle under the label “new concept”. The suggested concepts for this potential concept are in the same column under the label “Options”. Now, the user can select a suggested concept and press a button in the column on the right. The text on the button indicates the position in the thesaurus of the potential concept relative to the selected suggested concept according to the user. To put a potential concept in the remove- or stopwordlist, it is not necessary to select a suggested concept first. If no fitting suggestion is available, the column on the left can be used to search an existing concept from the thesaurus by name.

The first thing that leaps to the eye, when testing the application that implements the described enrichment process, is the large number of irrelevant potential concepts. This is

no big surprise, because I used all words unrecognized by the thesaurus as potential concepts. Part of those potential concepts are words that should be on the remove- or stopwords list. These are mainly words that don't mean anything, but are needed to form a good sentence. When these words are placed in the correct list, they won't reappear when, in the future, other documents are used in the enrichment process. The other part of the irrelevant potential concepts consists of general words, like most verbs, that appear frequently in documents. These are words that are too general, or don't mean anything by themselves and should therefore not be in the thesaurus. It can also be words that should be in **a** thesaurus, but not in **this** thesaurus, because they don't mean anything in the used domain.

The potential concepts that remain, are the words that should become a new concept in the thesaurus. Here, the suggested concepts are important. These should be concepts from the thesaurus which can be directly linked to the potential concept with a hierarchical or synonym relation. The quality of the suggested concepts seems not as good as intended. Most suggestions do seem to be related to the potential concept, but there doesn't seem to be a direct relation (i.e. parent, child or synonym) between the two. And since a thesaurus in Collexis' format doesn't support semantic relations other than parent, child or synonym, these suggestions can't be used.

Some of the suggested concepts mentioned above have a special type of semantic relation with the potential concept. This relation can be described as a *brother*-relation. In this relation both concepts hare the same parent, but have no direct relation. In the image below, Concept A is parent of child-concepts B and C. The concepts B and C have a brother relation with eachother.



Concepts B and C are 'brother' concepts

This type of relation cannot be made directly in the thesaurus. However, it is possible to add the potential concept as a child of the parent of the selected suggested concept. Unfortunately, adding this type of relation as a new option creates a new problem. Without more information, it is quite hard to judge if two concepts really should have a

brother relation in the thesaurus. In the test application it is not known what the parent concept of a suggested concept is. As long as this is unknown, it is just as well possible that the two concepts don't share the same parent, but only the same grand-parent. For example, "The Netherlands" and "South Africa" might seem to have a brother relation, because they are both countries. One could assume that both concepts have "Country" as their parent concept, so creating a brother relation would seem a good option. However, if the parent concept of "The Netherlands" turns out to be "European countries", and the parent concept of "South Africa" is "African countries", this option would not be the right choice.

In the example above, a wrong choice might be made due to lack of information. Because of this, a certain option may seem the right choice, but turns out to be not so good at all. This brings us to another problem: the suggested concepts can be too suggestive. If the potential concept is "Farming sector" and one of the suggested concepts is "Economic sector", the first would seem a child concept of the latter. But if one would know that in the thesaurus "Primary sector" is a child concept of "Economic sector", it would be better to add "Farming sector" as a child concept of "Primary sector". If the 'best' option is missing in the list of suggested concepts, one can easily be misguided by the suggested. This way one can choose the second best option (or worse) and place the potential concept in the wrong place in the thesaurus. A mistake which may have easily be prevented with a little more information on the structure of the thesaurus.

Another problem that arises, concerns the potential concept. This potential concept consists of one word. Since the sentence around the word is no longer visible, the word has a big chance of being ambiguous. For enrichment purposes this is not a big problem. Even if the potential concept is added to the thesaurus with a different meaning than it had in the document, it is a welcome addition to the thesaurus. For testing purposes, however, this is a bigger problem. If the potential concept is ambiguous, it is harder to judge if the suggested concepts fit the intended meaning of the potential concept.

As described earlier, most of the potential concepts that are offered to the user are irrelevant. This becomes quite clear when looking at the log-file generated by the enrichment process. After a few test sessions 86% of the potential concepts has been added to the removeword- or stopwordlist by the experts. Of the remaining potential concepts, only 13% has been given a place in the thesaurus. The other 87% should have been added to the thesaurus, according to the experts. However, for this last group of potential concepts no suggested concept has been given that was good enough to directly link the potential concept to. Regarding these percentages, the number of new concepts that can be added to the thesaurus seems quite low. Does the enrichment method not work? Does the used corpus not cover the domain well enough? Or are there other reasons why the percentage of concepts added to the thesaurus is so low? These questions and others shall be discussed in the next paragraph.

5.2. Discussion

During the enrichment process a number of choices have been made that might be improved. As a context for the proposed concept, I chose the sentence in which the proposed concept is located. A smaller context might not give a representation that is more or less unique for the proposed concept. A larger context might be a too general representation. The fingerprints to which the context will be compared during the enrichment process are the fingerprints of entire documents. The difference between the two sizes could decrease the quality of the resulting suggested concepts. The fingerprint of a document can be seen as the context of all the words that are important in that document, while the context of the suggested concept is the context of only one word. This can only lead to a reliable result if the potential concept is an important concept in the selected documents. Also the fingerprint of a document represents the entire document and not only the concepts that might be related to the potential concept. This way, part of the suggested concepts will probably be irrelevant.

To solve this problem, it seems better to compare the context of the potential concept with fingerprints of small pieces of documents. Since the context is created from a single sentence, the fingerprints should in this case also be the representation of sentences from the documents. This way the represented texts are about the same size. Unfortunately only the fingerprints of entire documents are stored in Collexis. It is not possible to retrieve the fingerprints of pieces of a document. And because it is not realistic to index each documents in Collexis by sentence, this solution is not an option.

Another way to make sure that the texts, from which the context and the fingerprints are created, are of a comparable size, is to use the entire document in which the potential concept has been found as context for this potential concept. In this case, all potential concepts in a document would have the same context. This is a big disadvantage. It would suggest that potential concepts in the document have the same representation and are therefore at least closely related. Although it is very well possible that all potential concepts from a document are related to the same subject (i.e. the subject of the article), it is quite unlikely that they are all closely related in a semantical way. The only way this would lead to a reliable result, is when the potential concept is an important concept in both the document is has been retrieved from and the documents to which fingerprints it's context will be compared. Another option is to still use the entire document in which the potential concept has been found as context for the potential concept, but give concepts that appear closer to the potential concept a higher score. This way, the entire document is used to represent the potential concept, but the representation is made unique for this potential concept by putting the emphasis on the concepts in the neighborhood of the potential concept. Unfortunately, the problem with the fingerprints in Collexis that represent entire documents remains.

Another problem described in the results is the lack of options and information in the test application. The possibility to add potential concepts as 'brother' or 'related concept' of a suggested concept would be welcome. Since Collexis does not support related concepts, this option drops out. To be certain if a potential concept can be added as a brother (or any other option) of a suggested concept, more information is required. A user needs to be able to see the thesaurus. He needs to see the structure of the thesaurus and the

concepts in it, especially near the suggested concepts. For example, selecting a suggested concept would show a screen containing the thesaurus, focused on the selected concept and the concepts surrounding it. Through this screen, the user would be able to browse through the thesaurus, starting at the selected concept. This way the user has more information on the suggested locations. The user can now, for example, choose to add a potential concept as a child of a suggested concept, or perhaps as a child of a child of the suggested concept. The exact location can now be selected in the graphical representation of the thesaurus. Due to the extra information and possibilities, a user is more capable of finding the ideal location in the thesaurus for a potential concept. Suggested concepts can in this case be seen as entry points to the thesaurus, instead of actual suggestions.

Finally, in the algorithm that searches for suggested concepts some choices have been made that might be improved. The context, which is used for finding related documents, is not the entire fingerprint of the surrounding text of the potential concept. As described earlier, the concepts in this fingerprint that are very general (concepts with a high score), or almost irrelevant (concepts with a low score) are removed from the fingerprint. In fact, the top and bottom of the fingerprint are ‘cut off’. It is not clear, however, how high a score should be to mark a concept as too general, or how low a score should be to mark a concept as irrelevant. The same problem arises when the context of the potential concept is being used to find documents containing a similar context. How well should the fingerprint of a document match the context, to be considered a good enough match? It is not clear what settings are best. The best settings may vary for each domain or even each document.

All choices made, result in a number of suggested concepts. Some choices will have a big effect on the result, and some hardly any effect at all. The size of the context probably has biggest influence. Other choices are for fine-tuning. But in the end, the biggest choice will be made by the human domain expert, who will, or won’t, give the proposed concepts a place in the thesaurus.

6. Conclusion

Motivated by the information overload, my goal was to construct a structured vocabulary and to create a methodology to keep the vocabulary up-to-date by enriching it. Such a vocabulary can assist a computer application in analysing human language. These applications can be used to process scientific articles and other documents and present this information in a human readable form. It would assist humans in finding their way through the huge amount of information which is available world-wide these days. Of all forms of structured vocabularies, which all have their own advantages and complexity, the thesaurus seemed the most useful one for this purpose.

Since information overload is such a hot topic nowadays, is surprised me that there are not many thesauri in the economic domain. In other domains, like medicine, the state of thesauri has much more advanced. It also became clear that, although a lot of research is being done on using thesauri and automatic processing of human language in general, only a few worked out methodologies for construction of thesauri exist.

Along the way it became clear that it would not be wise to construct a new thesaurus from scratch. A few thesauri exist which could be used as a solid, general base, which can be enriched to form a more specific thesaurus, dedicated to a specified application. It would be a lot of useless work to start all over again. I selected the EUROVOC thesaurus to serve as a base for the enrichment process.

Besides selecting a base thesaurus, some other preparations were needed. First of all, I needed a corpus of economic literature. I used a search tool based on the Google API to download articles from the internet. I used this corpus to find new potential concepts in and to find environmental information on these potential concepts, which exists of concepts that are already in the thesaurus and appear near the potential concept in the corpus.

As a base for the enrichment application, I used the Collexis engine. The main functionality of this engine is to create a unique representation, a fingerprint, of each document and to compare these fingerprints. A fingerprints exists only of a list of concepts from a thesaurus and a relative weight for each of the concepts. Using this fingerprint technology, I made a fingerprint of the concepts in the environment of each new potential concept: the context. By matching the context with fingerprints of other documents, I tried to retrieve concepts which might be closely related to the potential concept. The potential concepts were offered to a user, who then could decide if and where the potential concept should be added to the thesaurus.

While testing this enrichment methodology I ran into some problems, which caused a high percentage of irrelevant potential concepts and a low percentage of good suggestions on where to place the potential concept in the thesaurus. To me it is clear that some more research needs to be done to decide if this methodology can effectively be used to enrich

thesauri.

7. References

1. Tomatoes are not the only fruit
   Maewyn Cumming
   2002

2. Use of keyphrase extraction software for creating of an aec/fm thesaurus
   Branka Kosovac, Dana J. Vanier, Thomas M. Froese
   2000

3. TML: A Thesaural Markup Language
   Maria Lee, Stewart Baillie, Jon Dell'Oro
   1999

4. Relational Data Structures for Implementing Thesauri
   Randy Ballew, Thomas Duncan, Mike Blasingame
   1999

5. Library Tutorial - University of Helsinki
   Searching information - Defining your topic
   http://www.opiskelijakirjasto.lib.helsinki.fi/koulutus/libtut/4searching_7.html

6. Purpose of a thesaurus
   EUROVOC Thesaurus
   http://europa.eu.int/celex/eurovoc

7. Beyond Information Searching and Browsing: Acquiring Knowledge from Digital Libraries
   Ling Feng, Manfred A. Jeusfeld, Jeroen Hoppenbrouwers
   2001

8. C.K. Ogden, I.A. Richards
   The meaning of meaning
   1923

9. Semantic Networks: Visualizations of Knowledge
   Roger Hartley, John Barnden
   1999

10. Knowledge Discovery in Scientific Literature
    C.C. van der Eijk
    2001

11. State of the Art in Ontologies from the Semantic Web perspective.
Gómez-Pérez, R. Arpírez, O. Corcho, Y. Ding, M. Fernández-López, D. Manzano, M.C. Suárez-Figueroa
2002

12. Introductory Tutorial on Thesaurus Construction
Tim Craven
1997

13. Ontology-based Knowledge Representation for Bioinformatics
Robert Stevens, Carole A. Goble, Sean Bechhofer
2001

14. Overview Of Methodologies For Building Ontologies
M. Fernández López
1999

15. Towards a methodology for building ontologies
M. Uschold, M. King
1995

16. Methodology for the design and evaluation of ontologies
M. Grüninger, M.S. Fox
1995

17. Building and reusing ontologies for electrical network applications
A. Bernaras, I. Laresgoiti, J. Corera
1996

18. Knowledge sharing and reuse
A. Gómez-Pérez
1998

19. Toward distributed use of large-scale ontologies
B. Swartout, P. Ramesh, K. Knight, T. Russ
1997

20. EUROVOC Thesaurus
European Communities
2000-2005
http://europa.eu.int/celex/eurovoc/

21. Journal of Economic Literature (JEL) Classification System
American Economic Association
2004
http://www.aeaweb.org/journal/jel_class_system.html

22. VICORE
http://www.vicore.nl/

23. Information Overload: The Future Of Search And Information Access
2004
http://www.masternewmedia.org/2004/03/10/information_overload_the_future_of.htm

24. Semantic Web
W3C
http://www.w3.org/2001/sw/

25. Resource Description Framework (RDF)
W3C
http://www.w3.org/RDF/

26. Document Indexing Using Named Entities
Rada Mihalcea, Dan I. Moldovan
2001

27. Large-Scale Information Retrieval with Latent Semantic Indexing
Todd A. Letsche and Michael W. Berry
1996

28. Automatic Identification in Document Indexing
Feng Tang
2003

29. Information Retrieval on the World Wide Web
Venkat N. G. Udivada, Vijay V. R. Aghavan, William I. G. Rosky, Rajesh Kasanagottu
1997

30. Ontology Acquisition from On-line Knowledge Sources
Qi Li, Philip Shilane, Natalya Fridman Noy, Mark A. Musen
2000

31. Collexis B.V.
http://www.collexis.com

32. Google Web API
http://www.google.com/apis/

33. SHARED Global SHARing Point Server
PAHO/WHO
http://www.nlk.cz/czech/casopis/spoluprace/zpravywho/shar.htm

34. The KACTUS Booklet version 1.0
Esprit Project 8145
1996

35. The KACTUS View on the 'O' World
G. Schreiber, B. Wielinga, W. Jansweijer
1995

36. Towards a protocol for validating thesauri and ontologies
J. Kircz, J. v.d. Berg
2005

Appendix A – Thesaurus in PSF

This appendix contains the economical branch of the thesaurus I used as a starting thesaurus for the enrichment process. The thesaurus is formatted in Pipe Separated Format, so it can be used by the Collexis engine.

```
LEVEL|DEFAULT|218
0|economy;economics|10000000
1|economic policy|10600000
2|economic planning|10610000
3|development plan|10610100
3|national planning|10610200
3|regional planning|10610300
3|sectoral planning|10610400
2|economic policy|10620000
3|allocation of resources|10620100
3|austerity policy|10620200
3|deflation|10620300
3|deregulation|10620400
3|development policy|10620500
4|economic priority|10620510
4|sustainable development|10620520
3|economic convergence|10620600
4|convergence criteria|10620610
3|economic conversion|10620700
3|economic integration|10620800
4|globalisation|10620810
4|industrial integration|10620820
3|economic liberalism|10620900
3|incomes policy|10621000
4|guaranteed income|10621010
4|income stabilisation|10621020
3|intervention policy|10621100
4|support policy|10621110
3|protectionism|10621200
3|reflation|10621300
3|short-term economic policy|10621400
4|anti-crisis plan|10621410
3|structural policy|10621500
4|economic infrastructure|10621510
4|structural adjustment|10621520
2|economic support|10630000
3|aid for restructuring|10630100
3|aid to industry|10630200
3|aid to undertakings|10630300
3|Community aid|10630400
4|ECSC aid|10630410
3|employment aid|10630500
3|export aid|10630600
3|investment aid|10630700
3|modernisation aid|10630800
3|production aid|10630900
```

3|redevelopment aid|10631000
3|regional aid|10631100
3|sales aid|10631200
3|sectoral aid|10631300
3|State aid|10631400
1|economic growth|11100000
2|economic conditions|11110000
3|cost of living|11110100
3|economic activity|11110200
3|economic interdependence|11110300
3|economic resources|11110400
3|economic situation|11110500
4|short-term economic prospects|11110510
2|economic cycle|11120000
3|cyclical fluctuation|11120100
3|economic fluctuation|11120200
3|economic recession|11120300
3|economic recovery|11120400
3|economic stabilisation|11120500
3|economic stagnation|11120600
3|inflation|11120700
3|structural fluctuation|11120800
2|economic development|11130000
3|basic needs|11130100
3|developing countries|11130200
3|development potential|11130300
3|economic disparity|11130400
3|economic growth|11130500
3|economic reconstruction|11130600
3|economic take-off|11130700
3|economic transition|11130800
3|Group of 77|11130900
3|growth point|11131000
3|industrialised country|11131100
3|integrated development|11131200
3|least-developed country|11131300
3|newly industrialised country|11131400
3|underdevelopment|11131500
4|obstacle to development|11131510
1|regions and regional policy|11600000
2|economic region|11610000
3|coastal region|11610100
3|development region|11610200
3|frontier region|11610300
3|industrial region|11610400
4|declining industrial region|11610410
3|island region|11610500
3|less-favoured region|11610600
4|Mezzogiorno|11610610
3|peripheral region|11610700
3|priority region|11610800
3|region dependent on fishing|11610900
3|rural region|11611000
4|agricultural region|11611010
4|mountain region|11611020
3|tourist region|11611100
2|European Region|11620000

3|Alpine Region|11620100
3|Atlantic Arc|11620200
3|EC Mediterranean region|11620300
3|Rhine Valley|11620400
2|regional policy|11630000
3|Community regional policy|11630100
4|Community support framework|11630110
4|eligible region|11630120
4|integrated development programme|11630130
5|IMP|11630131
4|operational programme|11630140
3|region-EU relationship|11630200
3|regional development|11630300
3|regional disparity|11630400
3|regional integration|11630500
3|rural development|11630600
3|town and country planning|11630700
1|economic structure|12100000
2|economic sector|12110000
3|non-commercial sector|12110100
3|primary sector|12110200
4|farming sector|12110210
3|quaternary sector|12110300
3|secondary sector|12110400
3|tertiary sector|12110500
2|economic system|12120000
3|collectivism|12120100
3|common market|12120200
3|concerted economic action|12120300
3|controlled economy|12120400
3|economic reform|12120500
4|transition economy|12120510
5|post-communism|12120511
3|economic union|12120600
3|market economy|12120700
3|mixed economy|12120800
3|planned economy|12120900
2|economy|12130000
3|collectivised economy|12130100
3|housekeeping economy|12130200
3|industrial economy|12130300
3|national economy|12130400
3|post-industrial economy|12130500
3|public economy|12130600
3|regional economy|12130700
3|social economy|12130800
3|subsistence economy|12130900
3|underground economy|12131000
3|war economy|12131100
3|world economy|12131200
1|national accounts|12600000
2|accounting system|12610000
3|standardised accounting system|12610100
4|European accounting system|12610110
2|income|12620000
3|compulsory saving|12620100
3|distribution of income|12620200

4|distribution of wealth|12620210
4|low income|12620220
4|pauperisation|12620230
4|poverty|12620240
5|mendicity|12620241
4|wealth|12620250
3|household budget|12620300
3|household income|12620400
3|overlapping of income|12620500
3|purchasing power|12620600
4|purchasing power parity|12620610
3|redistribution of income|12620700
3|savings|12620800
3|social transfers|12620900
3|standard of living|12621000
2|national accounts|12630000
3|distribution per employed person|12630100
3|economic accounts for agriculture|12630200
3|economic aggregate|12630300
4|domestic product|12630310
5|gross national product|12630311
5|national income|12630312
4|gross domestic product|12630320
4|national expenditure|12630330
3|gross regional product|12630400
3|per capita distribution|12630500
3|regional accounting|12630600
1|economic analysis|13100000
2|economic analysis|13110000
3|econometrics|13110100
4|economic model|13110110
3|economic consequence|13110200
3|economic indicator|13110300
3|economic structure|13110400
3|economic survey|13110500
3|economic value|13110600
3|impact study|13110700
3|input-output analysis|13110800
3|macroeconomics|13110900
3|micro-economics|13111000
2|economic forecasting|13120000
3|forward studies|13120100
3|long-term forecast|13120200
3|medium-term forecast|13120300
3|short-term forecast|13120400
2|statistics|13130000
3|census|13130100
3|Community statistics|13130200
3|economic statistics|13130300
3|geographical distribution|13130400
3|international statistics|13130500
3|national statistics|13130600
3|nomenclature|13130700
3|official statistics|13130800
3|ratio|13130900
3|regional statistics|13131000
3|sample survey|13131100

```
4|sampling|13131110
3|statistical method|13131200
```

Appendix B – Google Tool

The code below is the program I used to download documents from the Internet, to create a corpus (see chapter 4.1.2). The file ”GoogleSearch.wsdl” contains the Google Web API. I replaced the key to use the API with ‘#’ signs. In order to receive a key, you must first register with Google at http://www.google.com/apis/.

```
use SOAP::Lite;

$key = '################################';
$maxResults = 10;
$lr = "lang_nl|lang_en";

print "Enter query (string)>";
$query = <STDIN>;
chomp($query);

while($nrDocs !~ /^\d+$/) {
  print "Number of documents (integer)>";
  $nrDocs = <STDIN>;
  chomp($nrDocs);
}

$googleSearch = SOAP::Lite -> service("file:GoogleSearch.wsdl");
@found = ("");
@titles = ("");

$foundDocs = 0;
print "\nRetrieving URL\'s...";
while($foundDocs < $nrDocs && $foundDocs % $maxResults == 0) {
  $result = $googleSearch -> doGoogleSearch($key, $query, $foundDocs,
  $maxResults, "true", "", "true", $lr, "latin1", "latin1");

  @results = @{$result->{resultElements}} or goto RETRIEVE;

  # Loop through the results
  foreach (@results) {
    # Store the URL of each result
    @found[$foundDocs++] = $_->{URL};
  }
}
print " ok\n\n";

RETRIEVE:
$t = 1;
for (@found) {
  print "$_\n retrieving ...";
  $document = $googleSearch -> doGetCachedPage($key, $_);
  $document =~ s/^.+<html>/<html>/s;
  print " ok\n saving ...";
```

```
  $title = $_;
  $title =~ s/[\/\\]$//;
  $docName = $t++ . "_";
  if($title =~ /\W([\w\.]+)$/) {
    $docName .= $1;
  } elsif($title =~ /^[\w\.]+$/) {
    $docName .= $title;
  }
  $docName .= ".html";

  open(DOCOUT, "> $docName");
  print DOCOUT $document;
  close(DOCOUT);

  print " ok\n";
}
print "$foundDocs documents retrieved\n\n";

system("pause");

EOF:
```