Digit·HVM revista digital d'humanitats

Opportunities for language-technology

Piek Vossen

CTO at Irion Technologies
Piek.Vossen@irion.nl

Abstract: The Internet is about communication and information. It is not simply a provider of data but also implies a real need for information. In other words, there is an growing necessity to cope with huge quantities of information, a large proportion of which is stored in text form. This article describes the opportunities offered by linguistic engineering for developing high quality products applicable in information society. A general model is proposed to deal with the different stages of data analysis, storage and access and an analysis is made with regard to the systems most widely used today. It is not an exhaustive list of possibilities and solutions. Nor is it completely up-to-the-minute, or is it intended to be. Technology is developing at great speed and any attempt to cover a particular field is already outdated by the time it is published. Rather, the author wishes to provide a perspective on the main current developments and attempts to stir the reader's interest in the present and the future of linguistic engineering.

1. Introduction

Ten years ago, the term language-technology did not exist, and even today, it is a completely new concept for many people. Soon, however, everybody will have more than a vague idea what it is. In the near future, most of our technology will be operated and even based on language-technology. Just like the remote control and the computer-mouse are now basic concepts in our society, so will speech-operated devices and patiently reading intelligent assistants. My little daughter used to hide the remote control to stay in charge of the television. Her daughter will have to come up with a different plan because devices will only listen to her voice on certain hours each day.

Such a plea for language-technology made ten-years ago, would have aroused a lot of scepticism, laughter and unbelief. Is it yet another desperate attempt to get funding for a (hopeless) project to develop language understanding systems? In the eighties, many funders and politicians just recovered from long-term projects to develop machine-translation systems, resulting in a handful of sentences translatable in a handful of language-pairs. In these days, computational linguists mostly participated in academic exercises to model linguistic formalisms or theories on computers. There hardly was any practical use for these models, nor any spin off. Gradually, some applications did develop, like spelling checkers, automatic indexing and summarization tools, but most of these do not involve any language-technology at all. Most people believed (and still do) that human language is too fuzzy and complex, too un-logical and vague, too ambiguous and implicit to be comprehensively captured in any model.

What has changed since then? Certainly not the languages. In these ten years a silent revolution took place. A revolution that is caused by another more loudly revolution: the Internet. Via the Internet, computers can now access a continuously growing amount of text in any conceivable language. With the Internet, language-engineers suddenly got massive empirical data. Before the Internet, linguists spent years of research to build (medium-size) language-corpora: i.e. carefully selected and monitored pieces of text to verify linguistic claims. Now, they have several magnitudes of text on the Internet, not just in English, but in most languages of the world.

However, it is not just the data that makes the difference. The Internet is all about communication and information. With the Internet, there is not just more data, but there is also a real information-need. To put it more strongly, there is a growing need to digest massive amounts of information, mostly stored in textual form. If you cannot find your way in the Internet to the relevant information, somebody else will. For some people it is more than a need, it is an information-problem.

Suddenly, there is a market and a critical mass of data, and everybody can play with it and develop solutions. Solutions are indeed being developed, almost every day another one. Many of them are bad, and most of them do not use any or much language-technology. But the good thing here is that they all create a market and that they establish an expectation and a baseline that can be used by applications that do make use of language-technology to prove their use. A statistical based summarizer selects sentences on the basis of the frequency of the words, which gives a certain quality sufficient for some purposes. It is then not that difficult to add some linguistic analysis on top of it and improve the result, for example by counting lemmatized words.

In this paper, I hope to explain the opportunities for language-technology to develop high-quality products for the information society. In the next section, I want to present a general model for discussing different layers of information analysis, storage and access. In the subsequent sections, I will discuss each of these layers. I will refer to the baseline systems that are now commonly used, to cheap and quick solutions that anticipate a human need to deal with information or knowledge in a more convenient way and, in so far available, to well-designed language-technology systems that can be used or developed to do a better job and still live up to the expectations. This description is not a comprehensive list of all possibilities and solutions, nor is and can it be completely up to date. Technology is developing rapidly and every attempt to cover the fields is directly out of date when it is published. Instead, I hope that it will put some major developments in perspective and give people a feel of what can be done or what is happening with language-technology.

2. Layers of information analysis, storage and access

The simplest way of access on the Internet is the WWW itself. You either directly type in an address or you follow a link, which also brings you to the address that it stands for. The Internet itself, however, only gives access to information, it does not digest it or tries to understand it in any way. Therefore, people still have to read it (if it is textual) and see if it is relevant. Since the Internet is so big, it changes constantly and the linking makes you get lost instantaneously, people that only browse are pretty helpless to find what they want (unless they know where it is).

Computer systems can help human Internet users by partially digesting the information for them and to provide access to this digested information, in some way or another. There are different levels of analysis possible, resulting in different representations of information, and, consequently, providing different ways of accessing and exploiting it. In Figure-1 below, you see a schematic representation of these solutions. At the left side a collection of HTML documents represents information on the Internet. The individual documents on the Internet can directly be accessed by the user with some Internet browser, one document at a time. In addition to HTML, there are of course many other textual documents represented as DOC, PS or PDF that cannot even be accessed with a browser. Going down, we see different ways of digesting the same information and giving access in alternative ways. Going to the right, there is first an analysis phase that results in a representation of digested information in the middle. Different sophistication levels of digested information are stacked on each other, indexes, hierarchies, facts and knowledge, resulting from different analysis processes: plain indexing, classification, data-mining and learning. At the right site, we see that ways of access to information are dependent on the sophistication of the analysis.

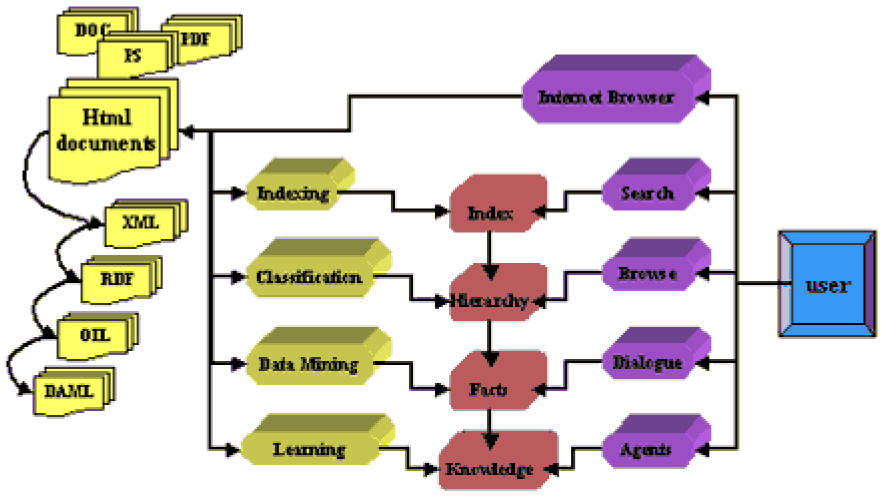


Figure 1. Layers of analysis

Indexes can be accessed with key word retrieval or search engines. You type in a few key words and the documents or HTML pages that have the strongest score with respect to these key words are returned. The output is a ranked hit list of documents or URLs. There are many different ways in which these indexes can be built or in which key words can be analyzed, combined, and expanded. Linguistic techniques can easily be integrated with the basic search engine technology. This can be at a fundamental level by improving the indexing and query analysis, but also as add on in the form of summarizers, recognizing languages, supporting multiple languages, matching queries to documents in other languages, etc.

An index can be seen as a flat list of normalized terms. Such a list can be the starting point for developing a hierarchy or classification tree. A hierarchy is some kind of classification of data or documents that can be browsed as a tree, going from more general to more specific concepts or classes, e.g. going from Sports to Ball Sports or Water Sports. At each node of the tree you may find a collection of documents that relate to the concept. There are different kinds of hierarchies as we will see later, e.g. thesaurus, taxonomy, ontology. A thesaurus often is a more global clustering of words, documents or objects that are related to some category (a so-called facet). Taxonomies and ontologies are more strictly defined hierarchies of all possible objects and their properties. Linguistic technology can be used to map words and expressions to concepts in the ontology: i.e. find the right meaning of the words, as well as to minimalize the ontology and automatically link the documents or terms to the nodes in the ontology. Ontologies and classifications are already present on the Internet in many ways, mostly as static resources, e.g. Yahoo classifications or product catalogues.

An ontology always captures generic relations between concepts or classes, but it does not capture specific facts on instances of these concepts. An ontology may state that a company has employees but a fact states that a particular company has certain people as employees. Although an ontology may help the extraction of facts (it defines all possible facts) it does not state what is the case at a certain moment of time. Ontologies are more persistent through time (e.g. a state can have a president), facts are time-bound (Clinton is the president of the US). Both facts and ontologies can be extracted from textual data but the process of extraction is very different. Ontological relations may follow from the analysis of large quantities of data in which certain frequent patterns are discovered, whereas a fact may be expressed only once, and even then, still not be true or outdated.

The nice thing of facts is that you can store them in a relational database. A relational database can be accessed using SQL queries. An SQL query consists of a command and certain references to items in tables, e.g. SHOW+PRODUCT(CAR)+TYPE(ASTROL)+HAS(AIRBAG). An SQL query is complex to formulate and it therefore makes sense to develop Natural-Language-To-SQL modules that map queries such as: Do you have any Astrol VANs with airbags?, into SQL requests. Complex questions can be cut up in smaller questions and this makes it possible to develop simple dialogues, where you first state what your interest is and next specify further properties and features. E-commerce is quickly pushing the representation of facts and the ways in which these can be accessed. It is a small step from a product catalogue to a relational database. E-commerce is a worldwide enterprise: there is no physical limitation of the clientele to your neighborhood. This requires that dialogues or questions in many languages can be handled. Furthermore, quality of service is more essential than in usual businesses because also your competition is not limited by physical boundaries either. Ease of access and communication are two important ways in which you can distinguish your business from others and this will drive the businesses of the future and, consequently, the development of language-technology.

The final layer looks at the same data from a very different perspective. Instead of directly interacting with users that want to access information, the user may have an assistant that operates in his place. With agent technology, we enter a whole new dimension of information access. Now a piece of software is trying to make sense of the information. Obviously, it can access exactly the same indexes, ontologies and facts as humans can (although more consistently and in much larger quantities), but it has much less capacity to know what is useful. Agents need some intelligence so that they can take decisions. Therefore, an agent or assistant does not just access facts but has to acquire knowledge. A user may tell the assistant to find the cheapest computer for the best service. The agent has to develop a plan to gather sufficient knowledge on the subject so that it can come back to you with the knowledge about computers that is requested (or even buy the computer if it is sufficiently reliable). If all information would be stored in digested form, an agent would not need language-technology to learn. However, since most information is still in pure textual form agents need to be able to understand as much language as needed. Furthermore, humans still have to be able to communicate with agents, and thus language-technology is eventually required after all.

There is still another tendency that will have an impact on how information in the Internet will be accessible. Instead of HTML, new markup languages are being developed. XML (http://www.w3c.org/XML/) is a more explicit format than HTML. XML not only gives a common representation for the make up of documents, but also for stating the kind of content. RDF (www.w3.org/RDF/), OIL (http://www.ontoknowledge.org /oil/) and DAML (http://www.daml.org/) go even a step further to formerly define the content itself. RDF (Resource Description Format) integrates a variety of web-based metadata activities including sitemaps, content ratings, stream channel definitions, search engine data collection (web crawling), digital library collections, and distributed authoring. RDF uses XML as an interchange syntax. OIL (Ontology Interchange Language) tries to combine Web standards with Description Logics and Frame representations in ontological approaches. OIL will make it possible to make inferences on content represented in its language. DAML (Darpa Agent Markup Language) is a formalism to help software agents interact with each other. The DAML language is also an extension to XML and RDF.

Each of these standards is necessary to exploit information and resources at their respective levels of analysis. Partially, it will make the above information dissemination schema obsolete but at the same time it will require tools that automatically map text or speech into these representations. As such, the formats do not really replace the text to information analysis phase, but they can be seen as representation formalisms for the storage of the analyzed information, as represented in the middle of Figure-1. In that sense, they will certainly have an effect on the ways of getting access to this knowledge. They will make it easier to develop software to access the digested information because developers can anticipate the common format in which it is represented.

I will not further discuss the developments of these markup systems below. I will also not discuss futuristic scenarios where agents mine the Web for knowledge and form communities to solve problems. In the next sections, I will mainly look at index and search, classification and browse, and question and answering systems. For each of these I will look at the current practice and discuss some examples. In addition, I will try to indicate the opportunities to build-in language-technology and improve these systems.

3. Index and search

Everybody is familiar with the first generation of search engines on the Web such as Yal (http://www.yahoo.com/) and Alta Vista (http://www.altavista.com/). These search engines in portions of the Internet and provide access via a key word search. The focus of these search engine on coverage of the Internet and actuality. They try to provide access to as many Web pages as t can and up date these links regularly.

It is important to realize what they actually index and how they map key words to indexes. In most cases, Web titles and index pages are used for building the index. They thus do not have direct access to the content of the Web pages or to other pages and certainly not to documents that are linked to these pages. Furthermore, they index strings and do not look at inflection, word function and certainly not at the compositional structure of the word combinations. To see the limitations of these search engines we will look at the following example queries:

poisonous medication; poisonous medicine; poisonous medicines; toxic medication; toxic medicines; medicine for toxication; medicines for toxication; medicines against poisoning; medication for toxication; Help my kids took poison, show me medication?; medicamento tóxico; medicamento intoxicación;medicina ponzoñoso;fármaco tóxico;

A number of things can be said about these queries:

1. They include plural/singular variants.

2. They include similar queries with different synonyms.

3. They include two compositional variants: one in which the medicines are poisonous and one in which you are looking for medication against poisoning.

4. The queries can be formulated in different languages.

You would expect the following behavior from a search engine:

1. It neglects inflectional variants such as plural and singular and gives the same results.

2. It neglects usage of synonyms and gives the same results.

3. It takes compositional differences seriously and returns different documents for each interpretation.

4. It can find information regardless of the language of the query.

When you look at the search engines on the Web, none of them do this. I included a list of query results below so that you can compare the results. You can also directly go to the Web sites and try it yourself. You will see that using a plural or singular or a synonym makes a lot of difference for the result list. None of them is the same. The indexing is purely based on strings. No normalization, stemming, compound analysis, or derivation analysis takes place. Furthermore, the exact compositional meaning is not taken seriously at all. Incidentally the same words co-occur as index items for the same documents, but even that is not always the case. The relation between the items is completely neglected:

WebSamples\Yahoo!_toxic_medication.htm
WebSamples\Yahoo!_poisonous_medicines.htm
WebSamples\Yahoo!_poisonous_medicine.htm
WebSamples\Yahoo!_poisonous_medication.htm
WebSamples\Yahoo!_medicine_for_toxication.htm
WebSamples\Yahoo!_medication_for_toxication.htm
WebSamples\Yahoo!_medication_against_poisoning.htm
WebSamples\AltaVista_toxic_medication.htm
WebSamples\AltaVista_poisonous_medicines.htm
WebSamples\AltaVista_poisonous_medicine.htm
WebSamples\AltaVista_medicine_for_toxication.htm
WebSamples\AltaVista_medicines_for_toxication.htm
WebSamples\AltaVista_medicines_against_poisoning.htm
WebSamples\AltaVista_medication_for_toxication.htm

Obviously, because indexing is string-based, a Spanish query will return Spanish documents. However, unless the words are spelled in the same way in both English and Spanish, there is no way in which you can find English documents with a Spanish query:

WebSamples\AltaVista_medicamento_toxico.htm
WebSamples\AltaVista_farmaco_toxico.htm
WebSamples\AltaVista_medicina_ponzoñoso.htm

There are other search-engines that try to be a bit more precise about the interpretation of the query. Below are the results for the same queries returned by Oingo and by Google. Oingo tries to present categories of information but also offers the option to further narrow down the meaning of the query terms. The meanings are based on the Wordnet database. Wordnet (http://www.cogsci.princeton.edu/~wn/w3wn.html/) is a freely available lexical semantic network. It contains a fund of concepts with semantic relations between them, and a mapping from English words to these concepts. Synonyms are mapped to the same concepts, forming so-called synsets. In Wordnet, *medicine and medication* are for example synonyms for the same concept, just like *poisonous* and*toxic*. You would thus expect that an expansion of the query words to their synonyms would result in similar hits regardless of the original words used in the query.

In the Oingo interface, you have to select the meanings of the query words manually. Once a meaning is selected, Oingo can find documents in which either the actual query word is used or another synonym, e.g. *toxic* instead of *poisonous*. As we can see in the next output pages, the results are not that spectacular. The hit lists are still very different when you use synonyms in the queries:

WebSamples\Oingo_toxic_medicine.htm
WebSamples\Oingo_toxic_medication.htm
WebSamples\Oingo_poisonous_medicine.htm
WebSamples\Oingo_poisonous_medication.htm
WebSamples\Oingo_medicine_for_toxication.htm
WebSamples\Oingo_medicines_for_toxication.htm
WebSamples\Oingo_medication_for_toxication.htm
WebSamples\Oingo_medication_against_poisoning.htm

Apparently, expansion to synonyms is not always helpful. As argued by Voorhees (1999), synonym expansion with Wordnet can even have a negative effect on the results, especially if there is no selection of meanings. That it can help, however, can be seen in the next examples. Selecting no meaning for *organ* or selecting either a*musical* or a *body part* sense does make a huge difference:

WebSamples\Oingo_organs.htm
WebSamples\Oingo_musical_organs.htm
WebSamples\Oingo_body_organs.htm

Unfortunately, you have to select the meanings by hand. There is no automatic disambiguation possible. It does not make much sense to develop such a disambiguation system at the query site, because most queries consist of one or two words. One or two-word queries do not provide sufficient context to disambiguate.

Google neglects different meanings. Instead it launches a meta-search to several search-engines and applies a document analysis to find the query terms in close proximity of each other. Google also shows the text fragments in which the words co-occur. Since the recall is enormous it can still generate sufficient hits.

WebSamples\Google_toxic_medicine.htm
WebSamples\Google_toxic_medication.htm
WebSamples\Google_poisonous_medicine.htm
WebSamples\Google_poisonous_medication.htm
WebSamples\Google_medicine_for_toxication.htm
WebSamples\Google_medicines_for_toxication.htm
WebSamples\Google_medication_for_toxication.htm
WebSamples\Google_medication_against_poisoning.htm

As you can see, the constraint that both words must co-occur leads to good results. Apparently, it is not always necessary to select a specific meaning. Google exploits the enormous redundancy of the information on the Internet. Most information is stored many times, formulated in many different ways and languages. The change that the information is stored once in exactly the same wordings as your query, is pretty high. Rather than diffusing the query by expanding it to synonyms or other phrases, it thus makes more sense to restrict to the literal matches only. Obviously, things get very different when retrieval is applied to small document collections or intranets. In that case, the information may be expressed only once in a single document. Query expansion is then essential to guarantee retrieval.

Both Google and Oingo try to give the impression of precision, but still do not really take the query seriously. They do not deal with phrasal variation nor do they look at the relations between the query terms and therefore cannot deal with the compositional differences in meaning. This is not surprising if one realizes the consequences of making such an analysis. Not only do we have to know the language of each document, we have to find out the beginning and ending of sentences (tokenization), parse the sentences to get the lemmatized words and the compositional structures, analyze compounds and derivations, detect multiword expressions, discover cross-sentence relations, determine the meanings of the words or phrases, etc. All this needs to be done for every language that is supported. The above search engines try to cover large portions of the Internet and need to update their indexes constantly. A linguistic analysis of documents and queries of that scale would require enormous processing time.

Still there are information providers that do try to deliver more specific answers to questions. AskJeeves has created a hype by their illusion that they can handle true natural language questions. Unfortunately, this is not achieved by analyzing and understanding the question but by simply looking up the question in a database where lots of questions are listed with the answer. These questions and answers are put in the database by hand. The results of the above queries are not very astonishing, but incidentally, we can get a very good effect, as can be seen for the call for example *help1* below. The query “Help my kids took poison, show me medication?” actually results in the rephrasing: What should I do if my child swallows posion?

WebSamples\AskJeeves_toxic_medicine.htm
WebSamples\AskJeeves_toxic_medication.htm
WebSamples\AskJeeves_poisonous_medication.htm
WebSamples\AskJeeves_medication_for_toxication.htm
WebSamples\AskJeeves_help1.htm
WebSamples\AskJeeves_help2.htm

Obviously, this approach is limited. The number of questions and answers is infinitive and the stored information is difficult to maintain and control for humans without any further support. It is just a matter of luck that the above cry for help is matched with a previously stored question that covers the same content. As you can seen in *help2*, you will not always be that lucky.

Clearly, all the larger systems lack language-technology and none of them is cross-linguistic, e.g. can match a query in Spanish to documents in English. There are commercial systems that try to improve the search technology with linguistic techniques for many languages and across different languages (Irion, Sail Labs,Textwise, Lexiquest). Most of these solutions are still being developed for smaller intranets and specific domains. They aim at higher precision, in other words: the answer should be in the first 10 hits, and preferably the sentence with the answer should be highlighted in the document. These next generation retrieval systems also handle different inflected forms and in some cases resolve compounds and multiword expressions. Because they are often applied to smaller homogenous document collections, there is also less ambiguity of meaning. For example, if the documents are all about music, then the organ-query does not need to be disambiguated. The word can only match with one meaning in the index. High-precision retrieval gives a feel of understanding but these systems do not really understand the question either. Compositional differences in the above example queries can still not be detected. A demo system with multilingual search for a specific document collection (environmental documents in Europe) can be seen at work at:http://dis.tpd.tno.nl/21demomooi/. The TwentyOne restrieval system developed by TNO, also uses fuzzy-matching. This means that spelling mistakes, derivations and compounding in the query can still be matched with index terms. For comparison, you can also look at Autonomy, who explicitly claims to be independent of languages and not to use language-technology and who also develop solutions for small intranets and portals.

Cross-lingual retrieval is usually achieved via bilingual dictionaries or using a multilingual semantic network. The EuroWordNet project developed such a network for 8 different languages: English, Spanish, Italian, Dutch, French, German, Czech, Estonian. Other languages are still being added to EuroWordNet. In the EuroWordNet model, synonyms are not only linked to concepts in each language but also across languages via an Inter-Lingual-Index. With such a multilingual wordnet database you can expand to synonyms within a language (*medicine* to *medication*) but also across languages (*medicine* to *medicamento* and *medicina*). Similar resources are being used or developed by the above companies.

4. Classify and browse

One of the disadvantages of search engines is that you will never get a good feeling of what is present. A hit list may give you some likely documents but you will never know what you have missed, and you do not know what documents are present. For the whole Internet this may make less sense because it more ore less contains everything, but for smaller document collections it does make sense to classify the information and present it by giving some tree of categories. Yahoo was the first bigger search engine that also uses categories that function as major topics within which you can search for further information. Another obvious example are electronic version of the Yellow Pages, e.g. http://www.yellowpages.com.au/. Yahoo classifications and the Yellow Pages are built manually. The coverage is necessarily limited and therefore still does not give you a good feel for what can be found.

Other companies develop systems that automatically categorize documents. Adams(2001) makes a distinction between 3 classification technologies:

1. Classification by example: the user makes a training set by manually assigning documents to categories. New documents are classified on the basis of their similarity with the training set. Companies: Mohomine, Inxight,Autonomy.

2. Statistical classification by key word extraction: Linguistic techniques are used to extract key words and documents are clustered that have similar key words. Companies: Semio, Cartia

3. Rule-based: explicit rules capture criteria on the basis of which documents are classified as A or B. Companies: Verity.

In contrast to query-document retrieval, purely statistical classification and classification by example seem to work quite well without language-technology. A document usually contains sufficient text to determine similarity to another document. Variation in words is found throughout the document and general a-specific words can be neglected because they occur in all documents.

Key-word extraction is obviously helped with a linguistic analysis and some of the above companies heavily rely on the extraction of the most salient key words for the classification. As a general rule, we can state that more and more linguistic analysis is needed the smaller the documents are. For example, classification or filtering of e-mails or URLs is more difficult without linguistic or semantic mapping. The topic must be recognized from a single subject line. Classification can only be done if individual word meanings are related to domains and these word meanings can be selected with a disambiguation method.

A specific problem for classification of documents is the visualization and access-method. A common way of visualizing the classification is a tree. These structures can however become quite large and complex, which hampers their use. Various technologies are being developed to deal with this dynamically. The following links show some nice dynamic examples:

Reuters: http://reuters.medialab.nl/aqua.htm

WebBrain: http://www.webbrain.com/open_IE.htm

Inxight: http://www.inxight.com/products_wb/tree_studio/tree_studio_demos.html

A general drawback of any classification is that it forces users to access information from a specific viewpoint. If the classification is large and complex, the user may get lost. He may be looking for the wrong distinction (it is not made or the desired information is classified differently), or looking for the correct distinction on the wrong place. To solve this, either the user should be able to organize the classification according to his personal wishes, or the classification can be complemented with a search option. In the former case, it must be possible to extract multiple views of classification maps and the user can select one. The underlying structure could include multiple classifications of the same documents and multiple relations between classes. Alternatively, a user may type in a class, which may be redirected to a categorization that is actually used. In that case, there is a separate index from words to categories. There are some initiatives to develop standardized representations of the same information in different ways. So-called Topic maps are used to display the same information from any perspective. More information can be found at: http://www.gca.org/papers/xmleurope2000/papers/s22-04.html. Visualization software can be developed on the basis of these standards.

The above products classify documents. A document classification is not really an ontology. There are however also other related ways of structuring information. Within E-commerce, we see that many companies provide catalogues for their products. Catalogues can also be seen as a kind of classification, but these are not necessarily associated with documents. To automate the building of catalogues is more difficult. Often, product descriptions are short and the categories do not necessarily follow from the descriptions. Still, some of these catalogues contain millions of products and access is often poorly supported. Furthermore, companies may want to determine the exact way in which the classification is organized. Compared to information in documents, catalogues are more poor but also more systematic. They usually cover only a few types of concepts with a limited number of properties or features. An obvious way to deal with catalogues is to convert them into relational databases. This will be discussed in the next section.

5. Data-mining and question-answer systems

A catalogue may have a hierarchical structure, just as a classification, but the hierarchies are less deep and complex. What is more interesting for catalogues are the features of the products. E-commerce sites often have feature descriptions (prices, delivery date, colours, sizes, amounts) and there can be a limited number of choices. Such a structure lends itself for storage in a relational database. Once this information is stored in a database, we can ask very specific questions for products with certain features. This kind of information is both ontological and factual. Ontological constraints dictate what properties or features each product or product type can have: this is the data model of the relational database. The actual products (serial numbers) and their status (actual properties) can then be seen as the facts that are expressed in the tables of a database.

Many companies are developing systems to store sophisticated ‘knowledge’ in databases to provide access to this knowledge. To the extend that information is present in the form of documents, general information and support on products can be given by indexing and classifying as described above. This does not however lead to specific knowledge. To acquire more detailed knowledge, some companies therefore store specific questions and answers in databases. They offer solutions to specific problems. Different types of knowledge are built up in various ways. I will not discuss them all but mention some examples to give an impression.

The easiest solution is to store or ‘can’ questions-and-answers just as AskJeeves does for general information. Some companies do this by storing specific problems and solutions of problems that are known for a product. This information is sometimes manually extracted from documents and manuals, sometimes based on FAQs, sometimes by logging user queries and answers, or by some kind of diagnostic dialogue that extracts knowledge when answering questions and that generates possible questions related to that knowledge. Because they develop these systems for specific customers they can still build highly sophisticated support systems that can be integrated in e.g. call-centers or help-desks. Two examples of companies that use this method can be found at:

ServiceWare http://www.serviceware.com/

Demo: http://www.serviceware.com/solutions/essdemo.asp

Primus http://www.primus.com/

Demo: http://www.primus.com/search.asp

Because they focus on questions-and-answers these companies still give the impression that they automate knowledge exploitation. However, all they try to do is to cut down costs for companies by automating some of the support services in a clever way.

Obviously, the above companies do not necessarily rely on language-technology. That is different for companies that actually try to extract knowledge from structured (database) and unstructured (text) data that is provided by customers. The key process is information extraction. Information extraction is partially based on language-technology and partly based on domain-knowledge. The domain knowledge functions as an ontology that limits the possible information that one is looking for. The language technology is used to extract information from text that fits this ontology. The process basically consists of a template-filling task, where the ontology defines the possible templates and the text analysis results in fillings. Because the ontology is small and explicit, the language-understanding part is able to extract reliable data. It will only interpret phrases that make sense within the interpretational frame of the ontology. It will be clear that compositional differences, such as*poisonous medicine* and *medicine for poisoning*, are essential for information extraction. For a discussion on such information extraction systems see Gaizauskas and Humpreys (1997).

Two examples of commercial systems that heavily rely on information extraction are:

iPhrase: http://www.iphrase.com/

ClearForest: http://www.clearforest.com/

Both companies use linguistic techniques to interpret text and phrases to fill templates on products and extract ontologies. In Figure-2 below, you see the architecture used by ClearForest. A definition of concepts and relations in the form of a 'rulebook' is used to guide the extraction of content from text. The rulebooks are pre-built for domains.

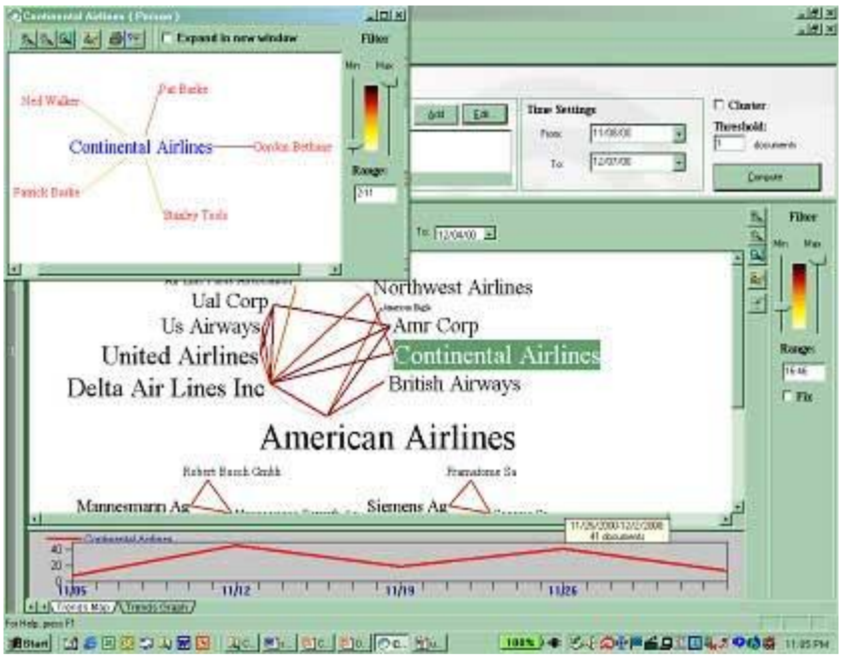
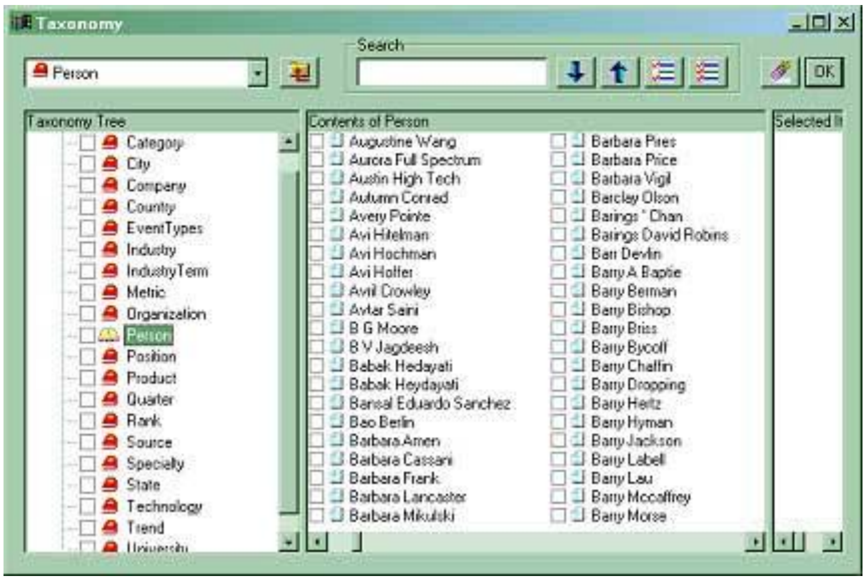


Figure 2. ClearForest Relations

Figure-3 shows how taxonomies are extracted from especific documents. In this example you see extracted persons. For each person different data may be found and expressed.



Category
City
Company
Country
EventTypes
Industry
IndustryTerm
Metric
Organization
Person
Position
Product
Quarter
Rank
Source
Specialty
State
Technology
Trend
University

Augustine Wang
Aurora Full Spectrum
Austin High Tech
Autumn Conrad
Avery Pointe
Avi Hitelman
Avi Hochman
Avi Holfer
Avril Crowley
Avtar Saini
B G Moore
B V Jagdeesh
Babak Hedayati
Babak Heydayati
Bansal Eduardo Sanchez
Bao Berlin
Barbara Amen
Barbara Cassani
Barbara Frank
Barbara Lancaster
Barbara Mikulski
Barbara Pires
Barbara Price
Barbara Vigil
Barclay Olson
Barings ' Chan
Barings David Robins
Barr Devlin
Barry A Baptie
Barry Berman
Barry Bishop
Barry Briss
Barry Bycoff
Barry Chaffin
Barry Dropping
Barry Hertz
Barry Hyman
Barry Jackson
Barry Labell
Barry Lau
Barry Mccaffrey
Barry Morse

Figure 3. Taxonomy ClearForest

At the iPhrase site you can see demos of the way they give access to information:http://www.iphrase.com/demo. Their data analysis makes it possible to handle complex questions and iterations of questions such as:

- What vans have airbags?
- Does the Astro also have a CD-player?

They can also generate overview tables with prices and properties and present these to users on request. After the first question they can give a table with all available*vans* that have *airbags* and specify further information such as *brands* and *prices*. The second question is then interpreted within the context that is created by the first question. Thanks to the rich database, iPhrase can handle question at the level of an SQL query.

EasyAsk is a company that specializes in just that. They have developed a complete E-commerce system where relational databases are extended with a natural language to SQL interface. The system works because it will recognize certain words in the query as SQL commands and others as names for tables. A query such as “Show me all vans with airbags?” can be handled because *show* is the command and*vans* and *airbags* are items belonging to certain tables. The system will look for products that are related to both table items and will show a list or a result table. There is not much processing of the query needed to get at such an analysis of the query. A simple list of commands, table names and some synonyms is sufficient. You can see a demo at: http://www.easyask.com/demo. Where as iPhrase puts focus on the data mining and the linguistic analysis of the queries and responses, EasyAsk puts more effort in a generic solution that can be applied to any relational database. The advantage of EasyAsk is that it is easy to apply to any existing relational database without much customization.

Figure-4 shows the design of the iPhrase system. The domain knowledge base plays the same role as the rulebooks of ClearForest. In addition to the knowledge base, iPhrase offers a sophisticared language interface to anlyze the quieries and to map them on the datbase, and a response generation component:

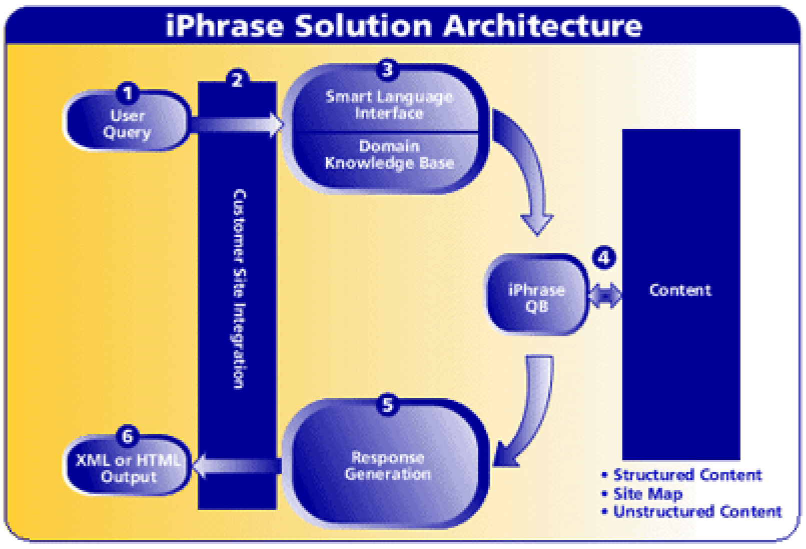


Figure 4. IPhrase System's design.

The next step for commercial systems would be the develop dialogue systems on top of the relational database. Various dialogue systems (commercial and experimental) have been developed during the early eighties. In Jönsson (1997) you can find a good overview. Obviously, a dialogue requires more sophisticated linguistic models and techniques, such as:

- Understand questions at a speech-act-level to make a difference betweenrequests, orders, clarifications, etc.
- Analyze anaphoric references in queries, such as Can I buy it?, where it refers to a previous entity.
- Give feedback on what questions can be answered and what not: Is there a swimming pool nearby?
- Give feedback on the reason for failure to answer a question: processing the language or making sense of the content
- Use clarifying questions in a clever way to resolve ambiguity or limit the amount of information to be given: a list of 200 hotels may be too much.

The development of good dialogue systems is difficult and subtle. Systems that try to mimic human dialogues can very easily become tedious to use. People are very goal-oriented and do not want to waste a lot of communication effort because a machine cannot understand the intention directly. However, if relational database as the above become more and more widespread in E-commerce, there will be a growing need to access these databases with limited dialogue systems. The iPhrase system is already going in that direction and there will be soon many more systems like that.

6. Other developments

There are two interesting developments in information retrieval and information portals that are indirectly related to language-technology:

- Personalization of information
- Authorization of information

Personalization is an AI-related technology to build profiles of users. On the basis of these profiles, users can be served more adequately with only the information that fits their interest. My profile may suggest that I am interested in musical instruments and not in medical subjects. The *organ* query discussed in section 3, can then directly be interpreted as a musical organ query. The same goes for classification systems. Only the categories that are of interest will be displayed. Profiles can be built by explicitly stating certain interests, but also by monitoring someone's Web-behavior. By just counting what somebody looks at, downloads or reads, a very detailed profile may be built. An interesting aspect of profiles is that they can be clustered and can be used to develop or offer specific services for larger clusters of users. Another interesting aspect is that previous interactions with customers on the Internet can be stored in a personal history and this can be used for referring to in the communication.

Authorization of information is becoming more and more important. The more information is available, the more difficult it is to verify the quality of this information. One of the natural developments of the Internet is therefore that people organize themselves in smaller communities (bulletin boards, chat boxes, email-lists). Within such a community they can focus and concentrate the communication and information. This also has the advantage that the community or group can control the information that is exchanged. Such a control can be seen as an authorization process. If a community of Object-Oriented programmers agrees that a certain piece of open-source software is a good database system, then this will have a certain value. It is possible to score usage of information in a community or to recognize evaluative comments within a community (this software is excellent!) and store this information together with the information. Retrieval of information can then be tuned to deliver the best-evaluated answer.

Bibliography:

FELLBAUM, C. (ed.) (1998). *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

GAIZAUSKAS, R. y HUMPHREYS, K (1997). "Using a semantic network for information extraction". *Natural Language Egnineering*, vol. 3, part 3&3, p. 147-169.

JÖNSSON, A. (1997). "A model for habitable and efficient dialogue management for natural language interaction". *Natural Language Engineering*, vol. 3, part 3&3, p. 103-121.

VOSSEN, P. (ed.) (1998). *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*, Dordrecht: Kluwer Academic Publishers.

VOORHEES E. M. (1999). "Natural Language Processing and Information Retrieval". In: *Information Extraction: Towards Scalable, Adaptable Systems*. Springer (Germany): M. T. Pazienza (ed.), p. 32-48.

Related Links:

W3C, World Wide Web Consortium: XML
http://www.w3.org/xml

WordNet
http://www.www.cogsci.princeton.edu/~wn/

Euro WordNet

http://www.hum.uva.nl/~ewn/

Reuters Medialab: Aqua
http://www.reuters.medialab.nl/aqua.htm

[Published on: december 2001]

Recommended citation:

VOSSEN, Piek (2001). “Opportunities for language-technology”. *Digithum*, issue 3 [article online].
DOI: http://dx.doi.org/10.7238/d.v0i3.597