A Heuristic Approach for Recognizing a Document's Language Used for the Internet Search Engine GETESS

A. Düsterhöft

University of Rostock,
Computer Science Department
A.-Einstein-Str. 21
18051 Rostock, Germany
duest@informatik.uni-rostock.de

S. Gröticke

SFG Consulting

Ebereschenweg 16
18209 Bad Doberan, Germany
SFG.Consulting@t-online.de

Abstract

In this paper, we illustrate how Internet documents can be automatically analyzed in order to identify the document's language. This language knowledge is then used for the Internet search engine, GETESS. The aim of the language–classification heuristics is to ensure that documents with the same content, but different languages (e.g., in German and English), will not simultaneously presented to the user as search results. The GETESS search engine only provides the results in the language relevant to the user. Consequently, the search–result set is narrower and more appropriately fits the needs of the user.

1 Introduction

Growing amounts of information in cyberspace make it increasingly difficult for network users to locate specific information for certain themes. Even experts sometimes experience the "joy" of becoming "Lost in Hyberspace".

In contrast, a wide variety of tools and services exist that are useful for information searches in the Internet, but whose efficiency is somewhat limited. The services tend to supply unsatisfactory search results, which are either too extensive, inapplicable, or incomplete. The majority of tools used for information searches in the Internet concentrate primarily on so-called syntactical attributes, such as TITLE or DESCRIPTION, without considering the actual meaning of the information (cf. [4], [5]).

The bulk of available information in the Internet is provided in natural- language format and supplemented with graphics. Furthermore, user queries are typically formulated using natural–language words and phrases. However, none of the well-known search tools take advantage of the use of a natural language combined with graphics, pictures, icons, and menus. This, despite the fact, that during the past few years the computer linguistic field has developed a wide variety of tools and mechanisms for partially automatic, natural-language processing taht could be employed as intelligent search support.

Internationally, the English language has established itself in the Internet. The majority of information in the Internet in Germany is presented in both German and English. Even in German-speaking areas, the German language plays an increasingly subordinate role in the Internet. Currently, a typical Internet user is either an information expert, student or computer freak with some command of the English language. However, as access to the Internet increases, the circle of users is also expected to become more multifaceted. At some point, knowledge of the English language can no longer be assumed .

The project GETESS (http://www.getess.de) [1] [3] focuses on the development of an Internet search engine which analyzes information in German within a defined domain.
The project began with the idea of combining Internet techniques with database and knowledge representation methods, as well as results from the computer linguists. The GETESS architecture subsequently integrates these aspects in order to give more detailed information to the user.

Therefore, the starting point is the idea to combine Inter-

[1] GETESS is funded by the German Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie (BMBF) under grant number 01IN802. The partners of the project are AIFB, the University of Karlsruhe, DFKI Saarbrücken, Gecko mbH Rostock, and the University of Rostock.

net techniques with database and knowledge representation methods as well as results from the cmputer linguists. The GETESS architecture integrates the different aspects in order to give more detailed information to the user.
Users can formulate their queries using natural–language phrases, and they will be supported by an ontology. A query's result set consists of so–called *'abstracts'* that are summaries of the contents of web documents.

In this paper we discuss the primary ideas behind GETESS, the functionality of the GETESS Gatherer, and the language classification heuristics for constructing a search context.

2 GETESS - GErman Text Exploitation and Search System

The GETESS project focuses on developing an intelligent Internet search site. It will offer the user an easy–to–operate system and will provide for the description of search requirements in a natural language. Intelligent systems situated between search sites and information will be responsible for condensing extensive and complex data into language-independent, content-weighted summaries (*abstracts*). The sum total of *abstracts* will provide a base and will subsequently be used as results for queries. Results will then be translated into the user's mother tongue.
A fundamental requirement for developing of an Internet search site is an extremely large amount of data. Accordingly, the amount of information allocated to the *abstracts* will also be substantial. Therefore, databases will be used for efficient storage and quick access to *abstracts*.

The core of the system features a search site that, like other search services, will provide availability to conventional information (e.g., HTML-Documents, etc.) However, it will also have the capability to access *abstracts* in order to supply satisfactory responses to search queries. Therefore, the GETESS search engine must also be capable of accessing database information. A response to a query will consist of a presentation of one or more results, as well as a ranking of the information to provide a plausible sequence for the user.

Linquistic and domain-specific information (e.g., in an ontology) will be employed when initiating a search query and for a response. Accordingly, one objective will be to guide users in preparing search descriptions. This means preparing search requests that specifiy parameters with regard to scope and quality of information. By offering content–dependent generic terms or concepts, it will be possible to limit the search domain. Additionally, a search process is better specified when relevant generic terms are negotiated. This is particularly important in situations where users have difficulty describing desired information or if they are unable to determine relevant search terms.

Through a specific linguistic and ontology–based analysis using information- extraction methods, structural links will be determined in natural–language queries (e.g., modifications). This will make it possible for searches considering relational circumstances between individual words in a natural–language query.

Language restrictions. The GETESS-System commences with the German language and notably offers users German-language WWW documents as search results. Furthermore, when users formulate natural-language queries, they are supported in the German language. [2] However, language autonomy will continue to be observed as much as possible during the development of the GETESS search site.
An additional step will consider making *abstracts* available for an English translation using an existing WWW automatic–translating tool. Furthermore, a study of the japanese language is planned to examine the use of the GETESS search site for Japanese documents.

Domain restrictions. An indepth, automatic natural-language analysis and classification of WWW documents initially requires a defined domain. Tourism was chosen for the GETESS search site that will subsequently enlist tourism documents from Mecklenburg-Western Pomerania as data. Tourism data from other German states will also be analyzed in later steps.
In order to investigate its adaptability and cost, the GETESS system will also be applied to a second domain in a subsequent step.

2.1 GETESS Architecture

The front end of the GETESS system (cf. a depictation of its architecture in Figure 1) provides a user interface that is embedded in a dialogue system controlling the history of interactions. Single interactions are handed to the query processor that selects the corresponding analysis method, viz,. the natural–language processing module, or the information retrieval and database query mechanisms. While the latter can be directly used as input to the search system, the natural–language processing module first translates the natural–language query into a corresponding database

[2] In this case, natural language means natural–language phrases because experience indicates WWW users rarely use full or complete sentences.

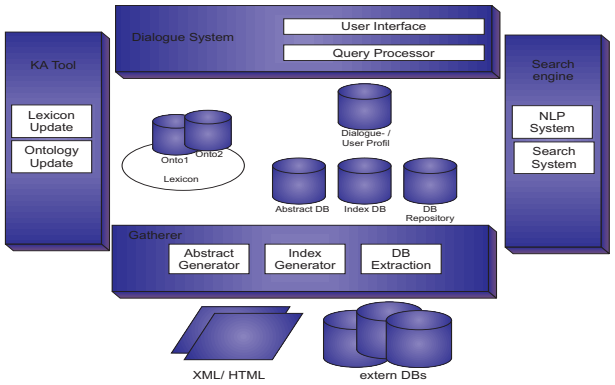

Figure 1. GETESS Architecture

query before it sends this formal query to the search system.

In order to process queries and search results, three kinds of resources are provided by the back end of the GETESS system. First, archieved information is available in several content databases (the *Abstract* DB, the Index DB and the DB Repository), the function of which is explained below. Second, the lexicon and the ontology provide metaknowledge about the queries, viz., the grammatical status of words and their conceptual denotations. Third, a database incorporating dialogue sequences and user profiles, provides control over dialogue interactions.

While dialogue sequences and user profiles are acquired, the course of interactions and the metaknowledge is provided by the human modeller with the help of knowledge acquisition tools (KA tools).The content databases must be filled automatically, because the contents of typical web sites change almost on a daily basis. For this task, the *gatherer* regularly searches through relevant XML/ HTML pages and specified databases in order to generate corresponding entries in the *abstract* database, the index database, and the database repository.

The content in the *abstract* database is derived from a robust, though incomplete, natural–language understanding module that parses documents and extracts semantic information, subsequently building an *abstract* for a document or a set of documents. These *abstracts* are sets of facts, i.e. tupels such as hasChurch(Rostock, Church-1), that could be extracted from natural–language text, like ”Rostock’s major church was built during medieval times.” The index generator builds access information for full–text searches with information retrieval methods, while the DB Repository offers relevant views into external databases.

The primary focus of the GETESS search site is the Internet search engine that makes it possible to store information in a database and then makes it quickly available for user inquiries. Above all, the storage of information in a database assumes the development of database solutions for the relevant data - in this case *abstracts*, a knowledge base and dictionaries. Furthermore, a search engine has to be developed with the capacity to initiate preparation of *abstracts* from Internet information and with the ability to access *abstracts* and relevant information from a database. The search engine also has to have the capacity for allocating user requests to database–stored *abstracts*.

3 The GETESS Gatherer

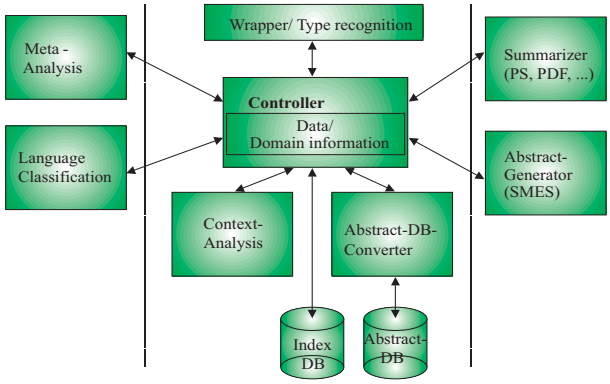

Figure 2. GETESS Gatherer

The GETESS Gatherer (cf. Figure 2) is one of the main components of the GETESS architecture. The Gatherer works as an Internet agent that periodically collects data from the Internet. After recognizing a particular data type, the Gatherer’s controller activates the summarizers and the SMES *abstract* generator in order to analyze the documents. The different summarizers, e.g., HTML, XML, PS, RTF, parse documents and extract keywords using information retrieval techniques. (The summarizers are mainly based on the Harvest system [2]). The *abstract* generator creates *abstracts* and uses the meta-analysis, where especially HTML meta tags and the HTML structure are analyzed, as well as the language classification and the context analysis, where knowledge about the structure of an complex Internet site will be extracted.

4 Language Classification

The language classification defines the document's language, i.e. the language identification problem (LID)[3]

The linguistic analyses SMES, combined with an Internet agent (cf. [1]), requires information about the document's language when an *abstract* of a document is being constructed. The appropriate German or English *abstract* tool is then be activated.
Otherwise, documents with the same content, but in different languages, should be arranged together via the constructed *abstracts* in the database. This results in equal *abstracts* representing documents with the same content but in different languages. Lastly, the GETESS search engine choosed only one of these *abstracts* as a query result – the document's *abstract* that is in the language the user desires.

It is a fact that using automatic natural–language processing for defining an Internet document's language is a time consuming and expensive task. Because it wasn't possible to combine our natural language abstract construction with a language classfication, we decided to realize a heuristic approach.

4.1 The classification algorithm

In order to define the language of a document, the following sources are used:

1. The HTML meta tag Language that explicitly defines the language of a document.

2. The Internet domain knowledge.
   The GETESS Context analysis attempts to find a general language structure of an Internet domain before the Gatherer starts to analyze single documents [1]. A domain's language structure is given when, e.g., a root node of an Internet site has two or more trees of documents and a tree of documents exists for each language. Another case of having a domain structure is when each leaf node of an Internet site has documents of the same theme in different languages.
   Such structure cases are expected in order to build a domain discription. The domain description is then used for the Gatherer's language classification.
   The URL of an Internet site will be heuristical analyzed for language implications. Parts of a URL name, such as 'en' or 'german' etc., are assumed to imply a document's language.

3. The key identification.
   If the number of counted words in the document is greater than 100 a key identification will be used to determine the document's language.

When defining a document's language, the heuristics outlined above will be weighted. Heuristics 1 and 2 provide definite indications for a specific language. In these instances, other heuristics are not required. If heursitics 1 and 2 are unable to define a language, heuristics 3 and 4 will be combined. In the event all of the heuristics are inapplicable, the German language will be assumed.

4.2 Key identification

In general, a language can be recognized by its phonetics and the corresponding spelling. Key identification can also provide clues to the origin of a language. Therefore, the use of certain letter combinations, or keywords particular to a language, can provide a heuristic approach for recognizing and classifying a document's language.

For example, in searching English documents, we could designate particular prepositions, pronouns, conjunctions, or articles typically found in the English language and use these for recognizing English–written documents. Some examples of key identifying words include:

- Prepositions: from, in, on, of
- Articles: the, an, a
- Pronouns: it, she, he, they
- Conjunctions: if, but

Additionally, letter combinations particular to a language are also helpful indicators, as long as the combinations appear frequently in the desired language and less frequently in other languages. Some combinations unique to, or indicating, the English language include: th, cy, ally, oo, ee, wh, and ng.

In general, the use of letter combinations alone is not as reliable as key identifying words because the same combinations may occur, although not as frequently, in other languages. For example, th is prevalent in the English language, but it also frequently appears in the German language in such words as Thunfisch or Theorie or Gasthaus. In any event, the use of key identifying words and letter combinations can be helpful in indicating a particular language.

[3]There are several publications about LID; most of them are on the basis of n-grams e.g. [6] and in the context of speech recognition [7]. The LID focus in GETESS is a very fast algorithm that uses Internet knowledge for a weekly updated search engine.

Table 1. Frequency of keywords and letter combinations

	1165e	1165	1264e	1264	9081e	9081
words	231	194	262	211	1146	1231
in	1	0	3	0	21	12
the	7	0	15	0	51	0
it	1	0	1	0	4	0
-th-	16	0	23	1	76	15
-wh-	0	0	3	0	1	0

4.3 Implementational results

The GETESS language classification uses the classification algorithm. The key identification is implemented with the keywords in, the, it and the letter combinations -th- and -wh-. A ten percent appearance rate of the counted keywords and letter combinations defines an English-language document. The following examples in table 1 show the frequency of the keywords and letter combinations in 6 selected documents at the Internet site http://www.all-in-all.com. The German document can be found under e.g., http://www.all-in-all.com/1165.htm and the English document (1165e) under http://www.all-in-all.com/english/1165.htm.

The rate was statistically defined after analyzing over 20.000 Internet documents of 12 different Internet domains of the tourism area.

The ten percent rate defines the correct language in 92 percent of the 20.000 documents. Shorter documents are problematic because the appearance of a keyword or letter combination could be sufficient to reach the (10per cent) rate.

5 Summary and Conclusions

In this article, we illustrated an approach for heuristically classifying a document's language. The classification is actually implemeted for the German and English languages in the GETESS search site's Gatherer. The GETESS Gatherer collects Internet documents from the tourism area, analyzes them, and constructs so–called *abstracts* of the documents. The *abstracts* are the basis for querying the search engine.
Future work focus on the integration of the third GETESS language, Japanese.

References

[1] M. Becker, J. Bedersdorfer, I. Bruder, A. Düsterhöft, G. Neumann, "GETESS: Constructing a Linguistic Search Index for an Internet Search Engine." In: *NLDB 2000 - Proceedings of the International Conference on Applications of Natural Language to Databases*, Versaille, France, June 2000

[2] Harvest Web Indexing. http://www.tardis.ed.ac.uk/harvest

[3] S. Staab, C. Braun, I. Bruder, A. Düsterhöft, A. Heuer, M. Klettke, G. Neumann, B. Prager, J. Pretzel, H.-P. Schnurr, R. Studer, H. Uszkoreit, B. Wrenger, "GETESS - Searching the Web Exploiting German Texts", In: *CIA'99 - Proceedings of the 3rd International Workshop on Cooperating Information Agents*. Upsala, Schweden, 1999, LNCS, Berlin, Heidelberg, Springer

[4] "Search Engine Watch: Tips About Internet Search Engines", http://searchenginewatch.com/

[5] "Search engines shoot-out - Top engines compared", http://coverage.cnet.com/Content/Reviews/Compare/Search2/

[6] W. B. Canvar, J. M. Trenkle, "N-Gram-Based Text Categorization", In: *Proc. of the 3rd Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, UNLV Publications/ Reprographics, pp. 161-175, 11-13, April 1994

[7] T. Schultz, I. Rogina, A. Waibel, "LVCSR-base language identification", In: *Proc. ICASSP'96 - IEEE Int. Conference on Acoustics, Speech and Signal Processing*, Atlanta, USA, 1996