Mining Answers in German Web Pages

Günter Neumann and Feiyu Xu
Language Technology Lab, DFK
D-66123 Saarbrücken, Germany
{neumann, feiyu}@dfki.de

Abstract

W[1]e present a novel method for mining textual answers in German Web pages using semi-structured NL questions and Google for initial document retrieval. We exploit the redundancy on the Web by weighting all identified named entities (NEs) found in the relevant document set based on their occurrences and distributions. The ranked NEs are used as our primary anchors for document indexing, paragraph selection, and answer identification. The latter is dependent on two factors: the overlap of terms at different levels (e.g., tokens and named entities) between queries and sentences, and the relevance of identified NEs corresponding to the expected answer type. The set of answer candidates is further subdivided into ranked equivalent classes from which the final answer is selected. The system has been evaluated using question-answer pairs extracted from a popular German quiz book.

1. Introduction

Recent advances in the area of information extraction (IE) have demonstrated successfully that domain-specific processing of large free text collections is achievable by using shallow NLP components; it has been shown that shallow technology can provide sufficient information for highly accurate and useful tasks to be carried out. The success in IE as well as in the area of Information Retrieval (IR) also have led to a very recent development of open-domain textual question answering systems (abbreviated as TextQA). They combine IR and IE technology in order to handle user queries of the sort ”Where is the Taj Mahal?” by producing answer strings extracted from relevant paragraphs of retrieved free NL documents, e.g., ”…list of more than 360 cities throughout the world includes the Great Reef in Australia, the Taj Mahal in India, Chartre’s Cathedral in France, and Serengeti National Park in Tanzania.”. The majority of TextQA handle a limited sort of NL questions, namely fact-based, short-answer questions, where answers are usually simple slot-oriented entities (e.g., when, where, who, what…). They are evaluated systematically as part of the text retrieval conference series (TREC). Evaluation is performed by means of a large corpus of textual documents from which a set of question-answer pairs is mainly pre-selected as reference material.

Currently, there is also an increased interest to consider the Web as an attractive resource for seeking answers for simple, fact-based questions in order to explore open-domain web-based question answering systems (abbreviated as WebQA), e.g. [7], [12], and [13].

Although the basic functionality between TextQA and WebQA is similar, there are some differences that make the development of WebQA a worthwhile venture. One important aspect of the Web is that of data or answer redundancy: the more frequently an answer appears, the easier it is to find it (cf. [2], [5], [9], [10]). The core assumption here is that with the massive amount of data on the Web it is more likely that a QA-system can find statements that answer the question in an obvious way, using simple pattern matching-based NLP methods (cf. [3], [5]). For example, if we consider a question like ”Who won the Nobel price 2000 in chemistry?” then a WebQA would consider all Web pages that contain the content words ”won”, ”Nobel price”, ”chemistry” and the time expression ”2000” and any person name. The candidate answer person name could then be the person name with relevant distance to the other terms and the one with the highest likelihood. In this case the redundancy of the Web is used as a substitute for

[1] The work presented in this paper has been funded by the BMBF projects Whiteboard and Quetal. Thanks to Olga Goldmann for her implementation support. The authors’ names are alphabetically ordered.

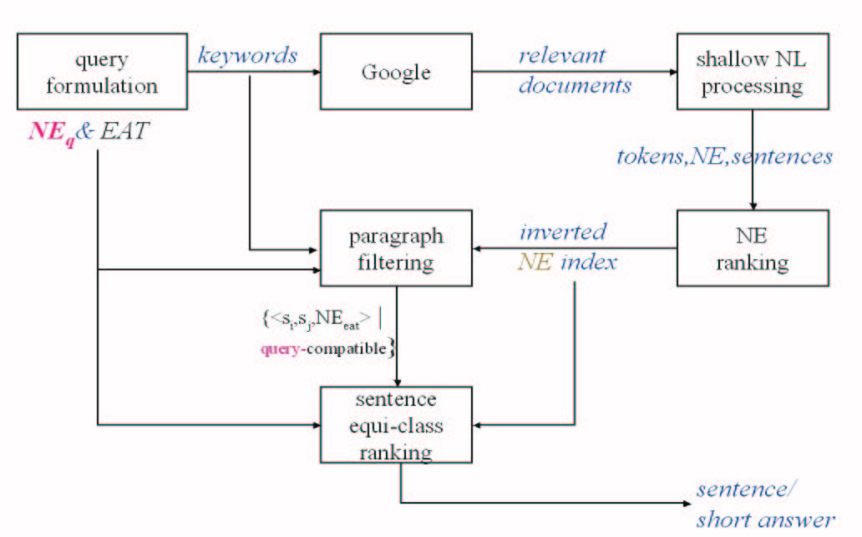

Figure 1. WAG's system architecture

a deeper structural analysis that would probably be more successful in case of small data sources and hence less redundancy (cf. also [5]).

Another important aspect of the Web concerns multi-linguality [16]. The growth of multilingual users and content on the Web in recent years is impressive. According to the newest data in 2002 provided by Global Internet Statistics[2], the English-speaking users are 36.5%, while the non-English users play now a dominant role with 63.5%, in which the European languages contain 35.5% (German: 6.7%). At the same time, the content distribution of non-English Web pages has increased too: there are 68.4% English Web pages, among other languages, the distribution of German Web pages is at the third position with 5.8% following the Japanese pages.

In this paper we describe WAG, an experimental search engine for performing answer extraction from German Web pages, allowing semi-structured queries for asking, e.g., who, when, where, how many questions. WAG uses Google for performing an initial Web search. The N-best Web pages found by Google are linguistically processed by a shallow NL processor, which among others recognizes Named Entities (NE such as person, company, location names, time, date expressions) and maps each document into a stream of sentences.

The core idea of WAG is to combine:

- an NE-directed voting technique by ranking all found NE (independently from the fact whether they are relevant for the answer using its term and document frequency, with
- an answer extraction and ranking strategy using word/NE overlap between query expression and answer candidates as scoring function extending the approach of [9].

The answer extraction process as a whole realizes a kind of text zooming method in the sense that we first identify the relevant paragraphs, then the relevant sentences, and then the relevant NE, i.e., the exact answer. In all of the subsequent steps NE's serve as anchors for the selection of the relevant textual window. WAG has fully been implemented and evaluated using a set of question-answer pairs from a popular German quiz book, [14].

2. System description

Our scientific view on the development of a generic question-answering system is that of a heterogeneous system architecture. The idea is that depending on the complexity of the query information (from simple fact-based questions, to relational template-based questions, to thematic-oriented questions, see also [4]). Shallow or deep QA strategies should be selected (or even mixed) which might involve different degrees of linguistic processing, domain reasoning or interactivity between a user and the system.

In case of using the Web as answering source for simple-fact based open domain questions, our current system design is focused on robustness, data directness, and scalability. For that reason we are using a simple query formulation process and few shallow NLP components. In the future we will scale up the system by integrating more advanced NLP components (as described in [11]).

2.1. Overview

[2] http://global-reach.biz/globstats/index.php3

Figure 1 displays the current system architecture of WAG. We will now describe briefly the major steps.

Query formulation. We are using a semi-structured template-based query formulation process. The internal representation of an NL question consists of the set of relevant content words, named entities and the expected answer type (EAT). Thus for a question like "Who won the Nobel price 2000 in chemistry?" the internal query object is as follows:

$$\left\{\begin{bmatrix} T:tok \\ V:\text{won} \end{bmatrix}, \begin{bmatrix} T:tok \\ V:\text{NobelPrice} \end{bmatrix}, \begin{bmatrix} T:DNE \\ V:2000 \end{bmatrix}, \begin{bmatrix} T:tok \\ V:\text{chemistry} \end{bmatrix}, \begin{bmatrix} T:PNE \\ V: \ ? \end{bmatrix}\right\}$$

In our current system the mapping from an NL question to such an internal query object is actually bypassed. Instead of parsing an NL expression, the user fills in a simple form that more or less corresponds to the structure of such an internal query object, e.g., the user enters the set of relevant content words, and selects named entities from a pop-up menu and fills in desired values (of course, freely selectable and including "?"). An additional motivation for this kind of query formulation is that it also supports transparent adaptation of the system to additional or new types of named entities or even domain specific ontologies, see also [15].

Document retrieval and shallow processing. All content words (all values from V) are passed to the Google search engine and the N-best documents (currently, N=50 is used) are further processed by our shallow NLP system, cf. [11]. In WAG we are only making use of the system's tokenizer, POS-tagger (which also recognizes sentence markers) and NE-recognition component. The NE-recognition component applies a set of regular patterns manually specified for handling German NE expressions (12 different types including company and person names, and locations). The coverage is state-of-the-art, e.g., 93%/80% precision and recall for company names, 92%/90% for person names, and 98%/86% for locations.

2.2. Mining and extracting answers

Finding the answer of a simple fact-based query basically means finding a single named entity – an instance of the expected answer type of the question, e.g., a person name for a who-question. We assume that a single sentence will contain the answer. However, since the Web pages returned by Google and further processed by the NLP system will contain many, many NE expressions in a single Web page as well as in multiple Web pages, simply iterating through all sentences to look for candidate answers is not appropriate.

In order to take advantage of the redundancy of NE expressions, we compute a weight for each recognized NE term. In later steps, we use this list of weighted NE terms as anchor for the retrieval of relevant paragraphs and sentences, and for the ranking of candidate answers.

NE ranking. The weight of an NE (and hence its relevance) is computed as follows:

$$(1)\quad NE = |DF| \times (\alpha * TF) + \sum_{i=1}^{|DF|} (1 - \frac{r_i}{N})$$

where

- *DF* is a list of documents containing the NE and ordered according to Google's ranking,
- *TF* is the frequency of the NE,
- α is a smoothing factor,
- r_i is the Google-rank of the ith document in *DF*

This means that an *NE* that occurs in more different documents will receive higher weight than an *NE* that occurs in fewer documents. Furthermore, NEs that occur in documents ranked higher by Google receive a larger weight than NEs occurring in lower ranked documents.
It might be the case that an NE is expressed using different forms, for example, "Heeger", "Alan Heeger", "Alan J. Heeger". We are able to recognize these variants and treat them as synonyms before the NE-ranking is computed. [7] also clusters such name variations together and provide them as alternative answers.

Paragraph filtering. Once the weight for every recognized NE is computed, we construct an inverted index from the individual NEs to their positions in the original Web page. We further subdivide these indexed NEs by collecting all NEs of the same type into an individual list (e.g., a list of all found person names). These NE-lists are used for paragraph selection, which works as follows: for each NE from the NE-list which is type-compatible with the expected answer type of the current question (e.g., person) we determine each of its position P_{NE} in the original Web pages and extract an P_{NE}-centered window $S_1S_2S_{NE}S_3S_4$, where S_{NE} is the sentence containing the NE, and S_i are the adjacent sentences. For each triple $S_1S_2S_{NE}$, $S_2S_{NE}S_3$ and $S_{NE}S_3S_4$, a weight is computed based on the number of containing content words of the question and the distance of each identified content word from P_{NE}. The highest scored triple is selected as a candidate paragraph.
The major reason for computing a candidate paragraph first instead of directly returning S_{NE} is that of increased robustness: it might be that the sentence S_i which contains the highest overlap with the question does not contain an instance of the expected answer type, but a generic phrase

(e.g., in case of referential expressions), then we are still able to consider S_i for the answer extraction process (see next section). In a similar way, we are able to handle named entities which have not been recognized by our NE-finder, but which co-occur with NE.

Sentence equi-class ranking. Note that we consider each occurrence of NE in the selected document set. Redundancy comes into play here, because it might be that different paragraphs from different documents which contain NE, differ only in view wordings because they contain the same or very similar sentences. We now describe, how we can collapse similar sentences into a single equivalence class. These classes are then used as ranked answer candidates.

The scoring function *EC* used in our approach for building and ranking sentence-based equivalence classes is an extension of the one described in [9]. *EC* is defined as follows:

$$(2)\quad EC = (olToken + olNE) * (1.5 + \frac{\sum_{i=1}^{n} EAT_i}{n})$$

We consider both the number of overlapped tokens (*olToken*), and the number of overlapped named entities (*olNE*).[3] Additionally, we consider the weight of each NE that is compatible with the expected answer type (EAT_i). Thus, a sentence has higher relevance than another one, if it shares more common words and named entities with the question, and if it contains instances of the expected answer type with high weights. For *n* instances of the expected answer type occurring in the sentence (which means that the sentence has ambiguous answers), we use their average weight. The EAT_i's are chosen as exact answers ranked according to their weights. For example, for the question

(3) *Welches Pseudonym nahm Norma Jean Baker an?*
Which pseudonym did Norma Jean Baker use?

WAG returns a list of equivalence classes of sentences (abr. eclass), and ranked answer type named entities. (4) shows one equivalence class for question (3). It contains only one answer candidate, where two ranked person names ("Marilyn Monroe" and "Norma Jean Mortenson") can be potential exact answers. The person name with higher weight is "Marilyn Monroe", which is selected as the best exact answer in this case.

(4)

```
<eclass rank='7.254032855348923'>
    <sentence url='http://www.beatlesfan.de/alle.htm'>
        Marilyn Monroe war der Kuenstlername von Norma
        Jean Mortenson, auch bekannt als Norma Jean Baker
        Marilyn Monroe was the stage name of Norma Jean
        Mortenson, also known as Norma Jean Baker
        <exact-answer type ='PERSON'>
            <name rank='0.6029282321968642'>
            Marilyn Monroe
            </name>
            <name rank='0.024088195477597402'>
            Norma Jean Mortenson
            </name>
        </exact-answer>
    </sentence>
</eclass>
```

3. Evaluation

We have performed an initial evaluation of WAG by comparing it to Google. Our German question-answer pairs have been extracted from a popular quiz book written by [14]. Currently, we have considered questions of two answer types: person and location, although our system can deal with other answer types too (e.g., date time, organization, number, address and currency). For each type, we have chosen manually the first 20 person-related questions and the first 19 location-related questions.

Here are some example question-answer pairs from the corpus:

- Welches Pseudonym nahm Norma Jean Baker an? (person)
  Which pseudonym did Norma Jean Baker use?
- Wer wurde 1949 erster Ministerpräsident Israels? (person)
  Who became Israel's first prime minister in 1949?
- In welcher ehemaligen Sowjetrepublik befand sich das Kernkraftwerk Tschernobyl? (location)
  In which former soviet state was the nuclear power plant Chernobyl?
- In welcher europäischen Stadt nennt man die Altstadt Alfama? (location)
  In which European city is there an old city part called Alfama?

In our evaluation scenario, we choose the top 50 documents found by Google. For each processed question WAG returns the following information:

- the number of documents found by Google;
- the top five Google snippets; we treat them as answers extracted by Google.
- the top five sentence-based equivalence classes;

[3] Note that this cannot include NE that have the same type as the expected answer type (EAT) because EAT actually serves as a typed variable in the question.

- each equivalence class contains the list of weighted named entities that are found instances of the expected answer type;
- the time point when Google was called for that question; this is important because this can also help us to track changes in the answer results caused by dynamically changed Web content (when trying to answer the same question later).

Given these parameters, we have manually evaluated the results. We consider two metrics: *recall* and *mean reciprocal rank* (MRR). Recall is the percentage of the questions answered correctly by WAG compared to all questions of the test corpus, cf. [7].[4] We consider two cases: the recall of the first relevant answer (top1) and the recall of the first three relevant answers (top3). The value of MRR is the mean of the reciprocal values of the rank of all correct answers among the top N:

$$MRR = \frac{\sum_{i=1}^{N} \frac{1}{rank_i}}{N}$$

In our current experiments (see next section), we have chosen N=5.

3.1 Evaluation of person questions

Figure 3.1.1 shows the distribution of the number of documents retrieved by Google using the person questions. For each of the 20 questions, we choose the top 50 documents. For 3 questions, Google cannot find any documents. The average number of retrieved documents is 28. In Table 3.1.1, Table 3.1.2 and Table 3.1.3, the average performance of Google snippets, WAG exact answers and WAG sentences is listed. In general, WAG has both a better recall and a better MRR value than Google, regarding both exact answers and sentences.

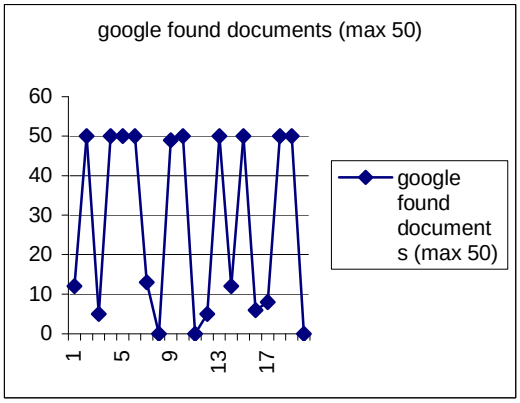

Figure 3.1.1 documents found by Google (person questions)

Google snippet	MRR (N=5)	Recall top1	Recall top3
all questions	0.103	0.3	0.35
excluding zero document cases	0.122	0.35	0.41

Table 3.1.1 the average performance of the Google snippets (person questions)

WAG exact answer	MRR (N=5)	Recall top1	Recall top3
all questions	0.212	0.45	0.55
excluding zero document cases	0,236	0.53	0.64

Table 3.1.2 the average performance of WAG exact answers (person questions)

WAG Sentence	MRR (N=5)	Recall top1	Recall top3
all questions	0,216	0.5	0.55
excluding zero document cases	0,254	0.59	0.64

Table 3.1.3 the average performance of WAG sentence (person questions)

3.2 Evaluation location questions

We have also evaluated 19 location questions. Google has found documents for all these questions. The distribution of the retrieved documents is shown in Figure 3.2.1. The average number of retrieved documents is 39. Table 3.2.1, Table 3.2.2 and Table 3.2.3 show that WAG has generally better performance than Google except for one case, namely the recall of top3.

[4] Note that the test corpus consists of the questions manually tagged with their correct answers. Thus if it contains N questions, then a recall of 0.5 means that WAG can answer half of them correctly, i.e., we do not consider « wrong answers ».

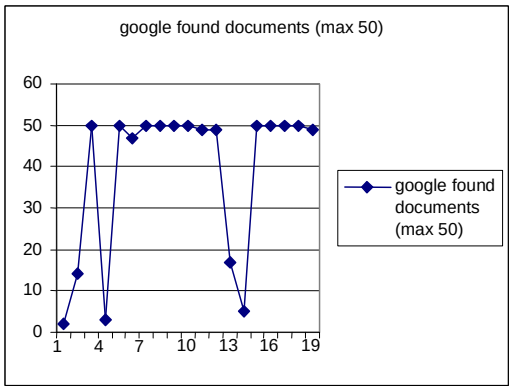

Figure 3.2.1 documents found by Google (location questions)

Google snippet	MRR (N=5)	Recall Top1	Recall top3
all questions	0,092	0.21	0.52

Table 3.2.1 the average performance of the Google snippets (location questions)

WAG exact answer	MRR (N=5)	Recall top1	Recall top3
all questions	0,135	0.31	0.42

Table 3.2.2 the average performance of WAG exact answers (location questions)

WAG sentence	MRR (N=5)	Recall top1	Recall top3
all questions	0,126	0.26	0.37

Table 3.2.3 the average performance of WAG sentences (location questions)

3.3 Summarization of the whole experiments

The following table summarizes the performance of WAG given the total 39 questions:

Metric	exact answer	Sentence
MRR (N=5)	0.174	0.171
Recall top1	0.38	0.38
Recall top3	0.48	0.46

Table 3.3.1 average values of all 39 questions

4. Discussion and related work

To best of our knowledge, WAG is the first publication of an open-domain "German Speaking" WebQA. AnswerBus developed by [13] also allows German input queries. However the queries have to be translated (using the translator provided through Babelfish[5]) into English because answer extraction is only performed for English Web pages.

Our experiments have approved the experiences reported in [2], [5], [7], [9] and [10], that the redundancy plays an important role for WebQA. The NE-ranking measure that we have developed is similar to that described in [5], which weights candidate answer terms by the integration of corpus frequency into the scoring function. The main differences are 1) that they count the passage frequency instead of the document frequency and 2) that they do not take into account the relevance of the document or passage given through the document or passage retrieval.

All published WebQA systems that are known to us are "English Speaking" systems (e.g., START, [6]; Ionaut, [1]; MULDER, [7]; AnswerBus, [13]; NSIR, [12]). A direct comparison of the corresponding reported results with our result is difficult, because our evaluation data set contains only German Web pages and German questions. Furthermore, currently we are using a small corpus (39 questions), where these other systems mostly used a corpus from TREC-8 of around 200 questions. However, we feel that our system has yielded encouraging results (cf. Table 3.3.1). [13] has compared a number of systems also using the TREC 8 corpus. Relevant numbers he reports are (considering recall on "top1"): AnswerBus=60%, LCC[6]=37.5%, START=14.5%. We obtain a value of 38% for top1 (with the restriction mentioned above, of course).

From a more qualitative point of view, the major performance influence of our methods can be summarized as follows:

- Redundancy: NE-ranking,
- Relative score is better than absolute score: the integration of document ranking into NE-ranking and further the integration of NE-ranking into scoring function,
- Building equivalence classes based on scoring function for ranking the answer candidates,
- The ambiguity of exact answers is handled in the scoring function.

The positive effect of these issues has also been highlighted by [9].

5. Conclusion and future work

We described and evaluated WAG, a Web-based answer extraction system for extracting sentences and short answers from German Web pages using semi-structured

[5] http://uk.altavista.com/babelfish

[6] Cf. [8]

fact-based queries. WAG uses ranking of named entities and statistical based text zooming as major strategies. It can handle ambiguous answers and performs a multi-document answer extraction process.

The focus of our research so far is based on simplicity and robustness following a data-driven, bottom-up system design. Of course there is enough room for increasing the performance of WAG. For example, currently we do no query expansion or re-formulation which would surely help to increase at least the recall. Furthermore, our semi-structured query format allows us very easily to express multi-fact (or template-based) questions (by specifying more than one expected answer type of different type). This could also be viewed as specifying a kind of on demand template (who did what when where). In this case our paragraph selection and sentence ranking process would be extended to return partially filled templates that have to be merged in a later step to find candidate templates. In some sense, the whole approach performs a kind of "template mining". We have already implemented a first prototype, however not yet evaluated.

References

[1] Steven Abney, Michael Collins, and Amit Singhal. Answer Extraction. In Proceedings of ANLP, 2000.

[2] Eric Breck, Marc Light, Gideon S. Mann, Ellen Riloff, Brianne Brown, Pranav Anand, Mats Rooth, and Miachael Thelen (2001). *Looking under the hood: Tools for diagnosing your question answering engine*. In Proceedings of the ACL-01 Workshop on Open-Domain Question Answering.

[3] Eric Brill, Jimmy Lin, Michele Banko, Susan Dumais, and Andrew Ng (2001). *Data-intensive question answering*. In Proceedings of TREC 2001.

[4] J. Burger, C. Cardie, V. Chaudhri, R. Gaizauskas, S.Harabagiu, D. Israel, C. Jacquemin, C. Lin, S. Maiorano, G. Miller, D. Moldovan, B. Ogden, J. Prager, E. Riloff, A. Singhal, R. Shrihari, T. Strzalkowski, E. Voorhees and R. Weischedel (2001). *Structures to Roadmap Research in Question & Answering (Q&A)*, In *NIST DUC Vision and Roadmap Documents*, http://www-nlpir.nist.gov/projects/duc/roadmapping.html.

[5] Charles L. A. Clarke, Gordon V. Cormack and Thomas R. Lynam (2001). *Exploiting Redundancy in Question Answering*. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New Orleans. September.

[6] Boris Katz (1997). *From Sentence Processing to Information Access on the World Wide Web*. AAAI Spring Symposium on Natural Language Processing for the World Wide Web. Stanford, California.

[7] Cody Kwok, Oren Etzioni, and Daniel S. Weld (2001). *Scaling question answering on the Web*. In Proceedings of the Tenth International World Wide Web Conference (WWW10).

[8] Sanda Harabagiu, Dan Moldovan, Marius Pasca, Mihai Surdeanu, Rada Mihalcea, Roxana Gîrju, Vasile Rus, Finley Lăcătuşu, Paul Morărescu and Răzvan Bunescu (2001). *Answering complex, list and context questions with LCC's Question-Answering Server* In Proceedings of the TREC-10.

[9] Marc Light, Gidcon S. Mann, Ellen Riloff and Eric Breck (2001). *Analyses for Elucidating Current Question Answering Technology*. Natural Language Engineering 1(1), Cambridge University Press.

[10] Jimmy Lin (2002). *The Web as a Resource for Question Answering: Perspective and Challenges*. In Proceedings of LREC 2002, Spain.

[11] G. Neumann and J. Piskorski (2002) A Shallow Text Processing Core Engine. In Journal of Computational Intelligence, August 2002, vol. 18, no. 3, pp. 451-476(26).

[12] Dragomir Radev, Weiguo Fan, Hong Qi, Harris Wu, and Amardeep Grewal (2002). *Probabilistic question answering on the web*. In Proceedings of the Eleventh International World Wide Web Conference.

[13] Zhiping Zheng (2002). *AnswerBus Question Answering System*. In Proceedings of HLT Human Language Technology Conference (HLT 2002). San Diego, CA. March 24 – 27.

[14] René Zey (2001). *Quiz für Millionen*. Falken Verlag. Niedernhausen

[15] A. Maedche, G. Neumann, S.Staab (2002) *Bootstrapping an Ontology-based Information Extraction System* In Szczepaniak, Piotr S.; Segovia, Javier; Kacprzyk, Janusz; Zadeh, Lotfi A. (eds), Intelligent Exploration of the Web, Springer, ISBN 3-7908-1529-2, 2002, pages 345-360.

[16] F. Xu (2003) Multilingual WWW - Modern Multilingual and Cross-lingual Information Access Technologies. In "Knowledge-Based Information Retrieval and Filtering from the Web". Witold Abramowicz (Ed.), Kluwer Academic Publishers, to appear.