A Question Answering System for German. Experiments with Morphological Linguistic Resources

Florian Köhler*, **Hinrich Schütze***, **Michaela Atterer****

* Institute for NLP
University of Stuttgart
koehlefn@ims.uni-stuttgart.de, hinrich@hotmail.com
**Institute for Linguistics
University of Potsdam
atterer@ling.uni-potsdam.de

Abstract

Question Answering systems are systems that enable the user to ask questions in natural language and to also receive an answer in natural language. Most existing systems, however, are constructed for the English language, and it is not clear in how far these approaches are also applicable to other languages. A richer morphology, greater syntactic variability, and smaller fraction of webpages available in the language are just some issues that complicate the construction of systems for German. In this paper, we present a modular Question Answering System for German which uses several morphological resources to increase recall. Nouns are converted into verbs, verbs into nouns, and the tenses of verbs are modified. We use a web search engine as a back end to allow for open-domain Question Answering. A POS-tagger is employed to identify answer candidates which are then filtered and tiled. The system is shown to achieve a higher recall than other systems for German.

1. Introduction

In the recent years, the field of Question Answering (QA) has evolved considerably, bringing forth a number of highly interesting systems and methods such as (Pasca and Harabagiu, 2001), (Dumais et al., 2002), (Hartrumpf, 2006), (Harabagiu and Hickl, 2006), (Moldovan et al., 2007). Question Answering systems enable the user to ask questions in natural language and to also receive an answer in natural language. They can be used for both searching the Internet or electronic corpus collections. However, most systems built in the past years are constructed for the English language, and it is not clear in how far these approaches are also applicable to other languages which have a richer morphology and greater variability in their syntactic constructions. English morphological variants as well as synonyms are usually found by web search engines commonly used as system back ends, but this is not necessarily the case for other languages. This fact together with the fact that web pages in other languages comprise a much smaller fraction of the www compared to English web pages[1], leads to a smaller recall; fewer potential answer candidates are found. Some systems such as Answerbus (Zheng, 2002) use a machine translation component to evade the problem above, but this raises a number of other problems, such as low-quality translations and the problem that many language/area specific questions cannot possibly be answered, because English web pages never contain the answers (such as the names of people well-known exclusively in those countries where the language is spoken). Therefore in this paper we aim at two contributions: 1) constructing an open-domain Question Answering system tailored to the needs of searching German web pages for answers (there is very little work in German QA); 2) experimenting with morphological resources to increase the recall of the system.

[1](cf. `http://www.internetworldstats.com/stats7.htm`)

2. Related Work

Currently, there are only few unrestricted Question Answering systems for German. Answerbus (Zheng, 2002) can treat German among other languages but uses the English system in combination with machine translation. ExtrAns and WebExtrAns (Mollá et al., 2003) are not open-domain, and (Neumann and Xu, 2004) require a semi-structured query formulation. The system by (Hartrumpf, 2006) is perhaps the most similar to the system presented here in terms of unrestrictedness and web-based search on German data, but it differs in that it is an adaptation of an original corpus-based system and uses a sophisticated semantic component. The widely known systems of (Dumais et al., 2002) and (Pasca and Harabagiu, 2001) (both built for processing English) provided the basic motivation for the system presented here – (Dumais et al., 2002) in terms of mining data from the web, and (Pasca and Harabagiu, 2001) in terms of using linguistic resources for query expansion/increasing recall. So far our system only uses morphological resources for query expansion but the system architecture is kept modular such that further resources can be tested in the future as in the work of (Pasca and Harabagiu, 2001).

3. System Architecture

The architecture of the system is schematized in Figure 1
First, the user's question is analyzed to render 1) the question type and 2) the relevant parts for query construction. The query then enters the search engine – in our case we used the Yahoo API [2]. The answer snippets returned by the search engine are then further processed: they are part-of-speech tagged to help find the relevant answer candidate,

[2]`http://developer.yahoo.com/download/`

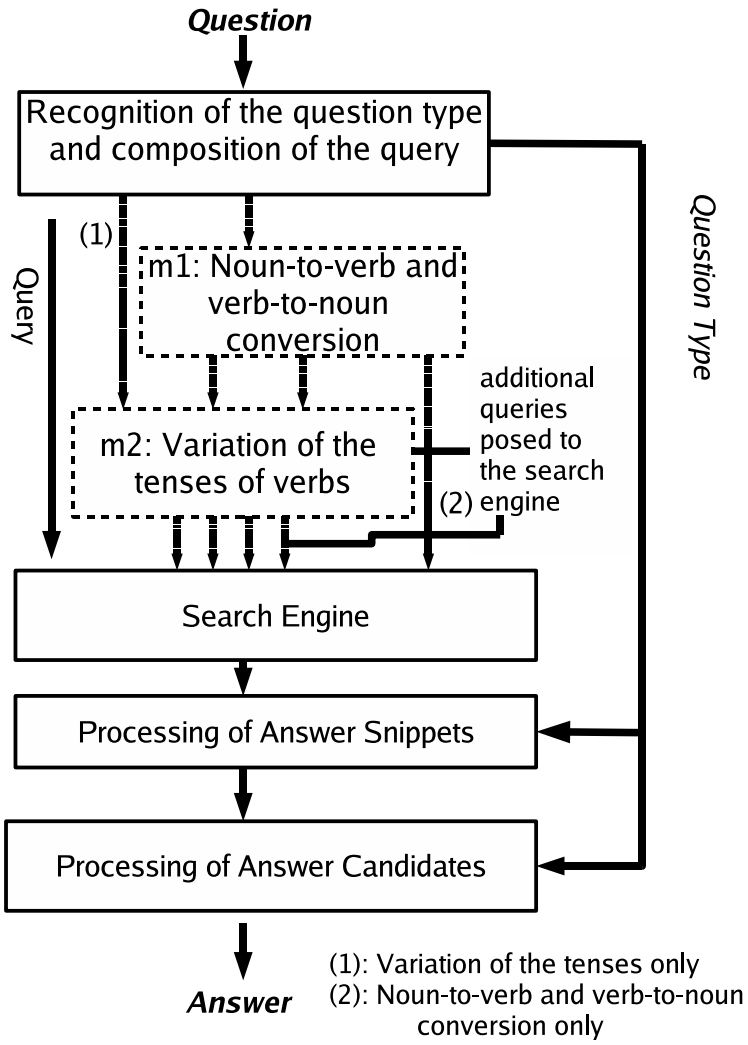

Figure 1: System architecture.

answer-candidates are matched to the question types, and similar answer candidates are merged before the most frequent answer-candidate is selected.

3.1. Question Types

Question types are mainly recognized by identifying the question word. The system so far can deal with the question types shown in Table 1 The POS-tag sequences that are expected as answer types refer to the STTS-tagset (Schiller et al., 1995).[3]

3.2. Query Composition and Query Modification

For composing the query the question words and auxiliaries are erased from the user's question. Two (optional) modules, *m1* and *m2*, are used for query modification. The first module performs noun-to-verb and verb-to-noun conversions in order to achieve a higher recall. The question *Was hat Edison erfunden?* (What did Edison invent?) normally results in the query +Edison +erfunden (+Edison +invented). Using query extension we also obtain the query +Edison +Erfinder (+Edison + inventor). The latter query resulted in 89.200 search results as opposed to 32.800 for the first query. Currently this module only uses a fixed list of words to convert into each other. The second module for query modification uses a more complex morphological resource SMOR (Schmid et al., 2004). This module is used to vary the tenses of verbs. This is useful because in German there are many more irregular verbs than in English. The verb *erfinden*, for instance is changed into past tense *erfanden*, and past participle *erfunden*. Using both modules the question *Wer sang beim Spiel Bayern München gegen die deutsche Nationalmannschaft im Jahre 2005 die Nationalhymne?* (Who sang the national anthem at the soccer game of Bayern München against the German national team in 2005?) results in 16 different queries: the word *sang* (sang) is modified to *singt* (sings), *gesungen* (sung) and *Sänger* (singer) and each of them is combined with the four variants of *Spieler* (player), namely *spielt*, *spielte*, *gespielt* (plays, played, played). The other words in the sentence are not modified.

[3]To compensate for tagging errors, we sometimes also allow for similar tags as in the case of the determiner.

Question type	German key words	expected answer format
Who-question	*Wer, Wie heißt*	sequences of NE-POS-tags followed by NN-POS tag sequences
What-question	*Was, Wonach, Womit*	optional determiner (ART, PRELS, PDS) followed by NN
location	*Wo, Wohin*	preposition (APPRART, APZR, APPO) followed by NE or NN
time	*Wann*	expressions built of numbers, terms describing months, days
abreviations	Words in capital letters	sequences of words whose initials correspond to the capitals
How many-questions	*Wie viele*	cardinal numbers
How [adjective]-questions	*Wie [adjective]*	cardinal number followed by NN or JJ
Why-questions	*Warum, Weshalb*	phrases starting with *weil* (because), *aufgrund* (due to)

Table 1: Question types known by the system.

3.3. Search Engine

As a search engine we use the Yahoo search engine and its API. This API allows us to restrict the number of answer snippets that are returned. As the work of (Dumais et al., 2002) shows, it is not advisable to use too many answer snippets – 200 was the optimal value for their system. Similarly, in our system, for every query, we currently use $x = \frac{200}{\textit{number of queries}}$ snippets.

3.4. Processing of Answer Snippets

The retrieved answer snippets are POS-tagged with the TreeTagger (Schmid, 1994), and the tag sequences are used to identify the words and phrases eligible as answer candidates as shown in Table 1. This strategy is more suitable for taking into account the greater syntactic variability of German, whereas the English AskMSR (Dumais et al., 2002) system relies mainly on the position of the answer candidate in the answer snippet.

3.5. Processing of Answer Candidates

The answer candidates retrieved from the snippets are first filtered and then tiled. The filtering step deletes answer candidates that contain the question, because it is desirable to avoid the answer to *Who discovered America* to be *America*, for instance. We only delete the relevant words in the answer candidates, however, not necessarily the whole candidate. With the question *What is the name of George Washington's father*, for instance, we would retrieve a candidate *Augustine Washington* and only delete *Washington*.

The tiling of answer candidates is similar to AskMSR: We include candidates that are contained in other candidates into the same set of answer candidates, such that the frequency count for *Columbus* (e.g. 50 occurrences) and *Cristoph Columbus* (e.g. 40 occurrences) is summed up in the end. However, we have to avoid cases where 200 occurrences of *Homer* are included into 2 occurrences of *Homer Simpson*. For this reason, we only include a set of substrings into the set of the superstrings when the size of the former set is at least 2/3 of the size of the latter set. Additionally orthographic variants are included in the same sets if for their Levensthein-distance l (Levenshtein, 1966) $l <= 1$ for words of length 8 or less, and $l = 2$ for longer words. Moreover, we use a transducer to transform numbers written in letters to be able to also include those in the corresponding answer sets.

4. Evaluation

Tables 2 through 7 show the evaluation results. *m1* and *m2* refer to the morphology module 1 (modify verbs into nouns and vice versa) and morphology module 2 (modify the verb tenses) repectively. *m12* is the combination of both modules. The basic (*bas*) system does not use any of these modules.

	M	P1	P5	M_2	$P1_2$	$P5_2$	R
m12	30.1	26.0	37.0	36.7	31.7	45.1	82
m2	29.3	25.0	36.0	35.7	30.5	43.9	82
m1	28.0	23.0	36.0	37.9	31.1	48.6	74
bas	28.9	24.0	37.0	38.1	31.6	48.7	76

Table 2: Development Set: 100 questions; questions with obsolete answers were removed and replaced by later questions.

4.1. Corpus

For the evaluation of the system we used 100 question-answer pairs from CLEF (Cross-Language Evaluation Forum) 2004 as a development set, and 100 question-answer pairs as a test set. The development set was used for implementing the question types. The test set was only looked at after the implementation was completed. As some of the question-answer pairs were obsolete (*What is the name of Peruan President Alberto Fujimori's wife?*, *What does Saab call back?*), we employed two different evaluation strategies. First, we replaced obsolete answers with up-to-date answers, wherever it was possible. Second, we removed obsolete-question answer pairs and replaced them with others, that occurred later in the list. To be able to make our work comparable to other work (Neumann and Xu, 2004), (Hartrumpf, 2006),we also employed a third strategy, where we only evaluated questions of answer types our system could recognize. (The system can also treat questions with unknown answer types, defaulting to the most frequent chain of NN and NE-tagged words in the answer candidates.)

4.2. Methodology

During the development phase the system was built using an automatic comparison of answer strings and gold standard answers. In the test phase, however, the evaluation was carried out by a human judge who had not taken part in the development. The judge was shown the question, the gold

	M	P1	P5	M_2	$P1_2$	$P5_2$	ns	R
m12	30.1	26.0	37.0	36.7	31.7	45.1	18	82
m2	29.3	25.0	36.0	35.7	30.5	43.9	18	82
m1	28.0	23.0	36.0	37.9	31.1	48.6	26	74
bas	29.0	24.0	37.0	38.1	31.6	48.7	24	76

Table 3: Development Set: 100 questions; questions with obsolete answers removed and replaced by later questions.

	MR	P1	P5	M_2	$P1_2$	$P5_2$	ns	R
m12	25.4	21.0	33.3	30.3	25.0	39.7	13	84
m2	25.4	21.0	33.3	30.3	25.0	39.7	13	84
m1	26.4	22.2	34.6	32.5	27.3	42.4	15	81
bas	27.0	23.5	33.3	33.1	28.8	40.9	15	81

Table 4: Development Set: 81 questions; questions with obsolete answers were updated where possible. Only questions where the system could recognize the question type were evaluated.

	M	P1	P5	M_2	$P1_2$	$P5_2$	ns	R
m12	23.6	18.0	34.0	27.4	20.9	39.5	14	86
m2	22.3	17.0	32.0	25.9	19.8	37.2	14	86
m1	23.3	19.0	32.0	27.7	22.6	38.1	16	84
bas	22.6	18.0	32.0	26.5	21.2	37.7	15	85

Table 5: Test Set: 100 questions; answers were updated where possible.

	M	P1	P5	M_2	$P1_2$	$P5_2$	ns	R
m12	19.7	14.0	31.0	23.7	16.9	37.4	17	83
m2	19.3	14.0	29.0	23.3	16.8	34.9	17	83
m1	20.6	17.0	29.0	25.1	20.7	35.4	18	82
bas	20.7	17.0	29.0	25.2	20.7	35.4	18	82

Table 6: Test Set: 100 questions; questions with obsolete answers were removed.

	M	P1	P5	M_2	P_2	$P5_2$	ns	R
m12	24.6	18.8	35.4	28.7	22.0	41.5	14	85
m2	23.2	17.7	33.3	27.2	20.7	39.0	14	85
m1	24.2	19.8	33.3	29.1	23.8	40.0	16	83
bas	23.5	18.8	33.3	27.8	22.2	39.5	15	84

Table 7: Test Set: 96 questions; questions with obsolete answers were updated. Only questions where the system could identify an answer type are considered.

standard answer and the system's 5 best answers via a computer program. He could then type in which of the answers was correct, or whether none of them was. This seemed a fairer way of evaluating the system, because in some cases the system provided good answers that didn't either exactly correspond to the gold standard answers (in terms of different spellings which could not easily be corrected using Levenshtein-distance etc.) or were completely different, but still correct. The gold standard answer for *What does FTP mean?*, for instance, was *a communist guerrilla organisation*, whereas the system returned *File Transfer Protocol*.

System	Prec1	Recall	unrestricted Q types
Neumann, Xu (2004)	29.0	?	no
Hartrumpf (2006)	22.9	76.5	yes
present system	20.9	86.0	yes

Table 8: Comparison with other systems

4.3. Evaluation Measures

For evaluating the system we measured the following:

- The Mean Reciprocal Rank (**MRR**, abbreviated to **M**): The sum of the relative rank of the correct answer (where the rank must be at least 5): MRR $= \sum_{i=1}^{n}((\frac{1}{rank_{(i)}}$ if $i <= 5; 0$ otherwise $))$
- percentage of correct answers given as the first answer of the system: **Precision1** or **P1**
- percentage of questions where the correct answer was among the first 5 answers: **Precision5**/**P5**
- the number of questions where the search engine did not return any snippets: **ns**
- the percentage of questions where at least one snippet was returned: **Recall** or **R**
- **MRR_2, Prec1_2, Prec5_2** (**M_2, $P1_2$, $P5_2$**) are values where the questions that didn't at least result in one snippet were not considered.

4.4. Results

The results are shown in Tables 2 through 7. *m1* refers to the morphological module 1, which modifies verbs into nouns and nouns into verbs, and *m2* refers to the morphological module 2, which modifies the tenses of the verbs. *bas* refers to the basic system without any of the two modules, and *m12* is the system with both modules.
In Table 8 we show a comparison of our system with two other German systems, giving *$Prec1_2$* and *Recall*. We used *$Prec1_2$* because the other systems were also evaluated without considering questions where the search engine did not return any results.

4.5. Discussion

Regarding the results in Tables 2 – 7, we can confirm that extending queries via morphological variation results into a higher recall. However, the tables also show, that in the case of the test set this does not necessarily lead to a better overall performance. We attribute this partly to the fact that our test set has a different distribution than the development set and is not representative for the whole question corpus. An evaluation carried out on the whole question corpus of 700 questions confirmed this notion – none of the evaluation measures was lower for the *m12* system than for the *basic* system. Another reason is that our system as presented here is not yet optimized with respect to many parameters, such as the optimal number of answer snippets to be returned,

weighting of queries etc. As Table 8 shows, the system is however comparable to other German Question Answering systems. Note that a fair comparison is not possible due to the different data that were used. One of the systems also was not evaluated considering unrestricted question types, and that increasing recall was not always tried or evaluated in the manner we do.

5. Conclusion

We built a highly modular question answering system for German that mines data from the Internet for answer extraction. The system is a highly extensible prototype which in its current version is tuned to the fact that German web pages comprise only an small fraction of the Internet compared to English web pages. The system uses morphological data resources such that it achieves a higher recall than comparable systems.

6. References

S. Dumais, M. Banko, E. Brill, J. Lin, and A. Ng. (2002). Web question answering: Is more always better? In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002), pp. 291–298, Tempere, Finland.

Sanda M. Harabagiu and Andrew Hickl. (2006). Methods for using textual entailment in open-domain question answering. In Proceedings of the ACL 2006.

S. Hartrumpf. (2006). Adapting a semantic question answering system to the web. In Proceedings of the EACL 2006 Workshop on Multilingual Question Answering (MLQA'06), pp. 61–68, Trento, Italy.

Vladimir I. Levenshtein. (1966). Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady, 10(8), pp. 707–710.

Dan I. Moldovan, Christine Clark, Sanda M. Harabagiu, and Daniel Hodges. (2007). Cogex: A semantically and contextually enriched logic prover for question answering. Journal of Applied Logic, 5(1), pp. 49–69.

Diego Mollá, Rolf Schwitter, Fabio Rinaldi, James Dowdall, and Michael Hess. (2003). NLP for answer extraction in technical domains. In EACL 03 Workshop: Natural Language Processing for Question Answering, pp. 5–11, Budapest.

Günter Neumann and Feiyu Xu. (2004). Mining natural language answers from the web. International Journal of Web Intelligence and Agent Systems, 2(2), pp. 123–135.

M. A. Pasca and S. M. Harabagiu. (2001). High performance question/answering. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'2001), pp. 366–374.

A. Schiller, S. Teufel, and C. Thielen. (1995). Guidelines fr das tagging deutscher textkorpora mit STTS. Technical report, University of Stuttgart and University of Tübingen.

H. Schmid, A. Fitschen, and U. Heid. (2004). SMOR: A german computational morphology covering derivation, composition and inflection. In Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC), pp. 1263–1266, Lisbon, Portugal.

Helmut Schmid. (1994). Probabilistic part-of-speech tagging using decision trees. In Proceedings of International Conference on New Methods in Languge Processing, pp. 44–49, Manchester, UK.

Z. Zheng. (2002). Answerbus question answering system. In Proceedings of the Human Language Technology Conference (HLT).