Classifying German Questions According to Ontology-Based Answer Types

Adriana Davidescu, Andrea Heyl, Stefan Kazalski, Irene Cramer, and Dietrich Klakow

Spoken Language Systems, Saarland University, 66123 Saarbrücken, Germany
Dietrich.Klakow@lsv.uni-saarland.de

Summary. In this paper we describe the evaluation of three machine learning algorithms that assign ontology based answer types to questions in a question-answering task. We used shallow and syntactical features to classify about 1400 German questions with a Decision Tree, a k-nearest Neighbor, and a Naïve Bayes algorithm.

1 Introduction

Although information retrieval techniques have proven to be successful in locating relevant documents, users often prefer to get concise answers to their information need. Question answering systems therefore allow the user to ask natural language questions and provide answers instead of a list of documents. Today, most question answering systems do extensive analyses of the questions to deduce features of the answer. One core task of the question analysis consists of the classification according to an ontology-based answer type. This type represents the most important link between question and answer and normally helps the system to determine what type of answer the user is requesting. Many question analysis components rely on hand-coded rules. However, this strategy is often time-consuming and inflexible: little changes in data and classification may require to make parts of the work over. Our aim therefore was to implement and compare different machine learning methods to classify questions according to a given answer type ontology. As data material we mainly used about 500 German questions collected in the SmartWeb project [Wahlster 2004].[1] We trained and tested our classifiers on about 800 additional German questions that we collected with a Web-based experiment [Cramer et al. 2006]. Based on these approximately 1400 questions and on the SmartWeb ontology [Sonntag et al. 2006] we derived an answer type classification consisting of about 50 hierarchically organized classes. (The SmartWeb

[1] This work was partially funded by the BMBF project SmartWeb under contract 01IMD01M.

ontology is focused on multimodal dialog-based human-computer interaction, therefore several aspects had to be adapted.) Unlike the question answering tracks known form TREC [Voorhees 2001] we did not restrict ourselves to a small set of factoid answer types (cp. Table 2 and 3).

To the best of our knowledge, to date there exists no other attempt comparing various classification methods for German questions. Even for English (in spite of its data richness) there are only few attempts to systematically contrast several feature sets and classifiers. In 2002 [Li et al. 2002] studied a hierarchical classification algorithm on the TREC 10 questions. They were able to show that their approach achieves good results compared to heuristic rules. However, their answer type ontology only consisted of a few answer types. [Day et al. 2005] integrated a knowledge-based and a machine learning approach for a Chinese question set taking about 60 answer types into account. Most similar to our work are the studies by [Zhang et al. 2003] and [Li et al. 2002], which both compare several machine learning techniques (inter alia: Support Vector Machine, Decision Tree, k-nearest Neighbor, and AdaBoost) for an English question set. They both make exclusively use of shallow features like bag-of-words and bag-of-ngrams, which appear to be astonishingly appropriate compared to the well established use of rules and rule-like features in traditional question analysis components. Like [Li et al. 2002] they only considered a few answer types.

The rest of this paper is organized as follows: In Section 2 we describe our data, the features that we (automatically) annotated, and the answer types. Section 3 gives a very short introduction to the three classifiers and presents the design of our experiments. Section 4 discusses the evaluation and our results. Finally, we draw conclusions and summarize our work in Section 5.

2 Data, Answer Types, and Features

We used two question collections. About 2000 questions were complied in the SmartWeb project to foster the development and evaluation of the open-domain question-answering system. Some of the questions were elliptic or anaphoric and therefore did not fit in our open-domain approach. Table 1 gives an example of such inappropriate questions. We therefore excluded those questions. The remaining corpus consisted of about 500 questions.

We additionally collected about 1400 questions with a Wikipedia-based tool ([Cramer et al. 2006]). We merged both collections and manually classified all questions according to the given answer types. Table 2 shows a summary of the most frequent question words occurring in our corpus. Table 3 shows the distribution of the answer types.

The majority of the questions belong to the inquiry types *concept completion* and *quantification*–here: Location, Person, Date, and Number.Count – which is well reflected by the ontology. The *concept completion* and *quantification* are the most basic and simple types in question-answering. However,

Table 1. Elliptic and Anaphoric Questions

Q_{1a}:	Definiere die freie Enthalpie. Define the free enthalpy
Q_{1b}:	Wie wird sie noch genannt? How is it also called?
Q_{2a}:	Wann veröffentlichte Milan Füst seine ersten Gedichte? When did Milan Fust publish his first poems?
Q_{2b}:	Und worin? And where?

Table 2. Summary of the Question Types in Our Collection

question word	percentage	question word	percentage
wann (= when)	10.9%	wie/wie* (= how)	18.0%
warum (= why)	1.4%	wo (= where)	9.3%
was (= what)	20.0%	wo* (= whereby etc.)	3.8%
welch* (= which)	8.9%	not first word	9.1%
wer (= who)	11.8%	without	6.8%

Table 3. Summary of the Answer Types Occurring More Than 50 Times in Our Collection

question word	percentage	question word	percentage
Definition	15.4%	Explanation	14.1%
Location	10.9%	Date	10.6%
Person	9.3%	Number.Count	5.0%
Yes/No	4.3%	...	

there are a lot of questions that belong to the inquiry types *comparison*, *definition*, and *request*–here: Definition and Explanation. These types are much more complex and up to date research challenges. We tested several feature sets separately. Most of the features are used in English question-answering systems as well. Although, there are some that we added considering the rich German morphology and the fact that we aim at building an open-domain and Web-based question-answering system. As feature sets we used:

- **Collocations** are n-grams of lemmas that normally co-occur. In the first instance we selected them manually. Later we decided not to trust our intuition and calculated the standard deviation for all bigrams that either include a question word or occur at the beginning of a question. We chose those bigrams that had a high standard deviation.
- **Trigger-words** are mainly question words such as ”wann” (= when), ”wo” (=where). These words are the most obvious features in question-answering.
- **Punctuation marks** consider whether there is a question mark or an exclamation mark at the end of the question. As trigger-words the punc-

tuation mark is an obvious feature although it is normally of minor importance and little robustness.

- **Question length** is counted in word tokens. We found that there is sometimes a though weak relation between specificity of a question and the length of its answer.
- **Named entities** were annotated with LingPipe developed by the Alias-i, Inc., which assigns labels to proper nouns such as person names, locations, and organizations. Unfortunately, the named entity recognition in short sentences and questions shows a lack of robustness. Nevertheless, named entities play a part in the sentence patterns and are supposed to be an important hint for certain answer types.
- **Lemmas** are annotated using the TreeTagger developed at the Institute for Computational Linguistics of the University of Stuttgart. Lemmas were interesting features for a medium size German question corpus since they are able to condense several words to one.
- **POS-tags** are also annotated using the TreeTagger (see above), it uses the Stuttgart-Tuebingen Tagset. We hope to capture certain syntactic structures of the questions with this - kind of shallow - feature.
- **Bag-of-words** is a shallow method often used in information retrieval. It considers documents (here: questions) as a set of words disregarding any syntactic or semantic relation.
- **Sentence patterns** are constructed manually on the basis of about 500 questions annotated with POS-tags, lemmas, and named entities. We identified key words/verbs and encoded the verb arguments as a set of possible POS-tags and named entities with regular expressions.

3 Classifiers, Experimental Design, and Evaluation Method

We implemented three classifiers for our question analysis task: a Decision Tree, a Naïve Bayes, and a k-Nearest Neighbor algorithm. Decision Trees and k-Nearest Neighbor algorithms, respectively, are well known as very robust with only little training data available. We decided to contrast these two with an algorithm that is supposed to be both simple and scalable: the Naïve Bayes. We give a very rough overview (see [Duda et al. 2000] and [Mitchell 1997]) of all three in the following sections.

3.1 Decision Tree

A Decision Tree over the features F_1, F_2, ... , F_n with discrete values and the classes C is a tree where:

- every node is labeled with one of the features F_1, F_2, ..., F_n.
- every leaf is labeled with a possible class.

- every node with the label F_i has as many outgoing edges as there are possible values for the feature F_i.
- every outgoing edge for F_i is labeled with a possible value for F_i.

In every classification step the algorithm decides about the next feature according to its information gain. Splitting stops when either all examples are correctly classified or the information gain remains under a certain threshold. In addition, there are several more general design decisions: e. g. optimization according to the overall complexity of the tree or pruning after completion of the algorithm. We decided against pruning because of data sparseness and used the information gain as criterion for splitting.

3.2 k-Nearest Neighbor (kNN)

The key idea of this algorithm is to predict the class of an yet unlabeled example by computing the dominant class of its k nearest neighbors. It works as follows:

- select a suitable value for k;
- compute the distances between the feature vector a of an unlabled example and the vectors v_i of all training data by means of Euclidean distance

$$D_i(a, v_i) = \sqrt{\sum_j (a[j] - v_i[j])^2} \tag{1}$$

- find the k nearest neighbors among the training examples;
- predict the label based on the most frequent one among the k nearest neighbors.

To determine an appropriate k-value we conducted several experiments with features on the sentence level such as collocations and named entities. According to the best results in our experiments the value of k was established to 12.

3.3 Naïve Bayes

A Naïve Bayes classifier is a probabilistic classifier, which determines the most probable hypothesis h from a finite set H of hypotheses. The probability model for this classifier is a conditional one, it can be derived using Bayes' theorem. The joint model of Bayes' theorem and the strong independence assumption is expressed as

$$P(h|a_1, ..., a_n) = \frac{P(h) \prod P(a_i|h)}{P(a)}. \tag{2}$$

The classifier calculates the probabilities with the joint model for all classes and chooses the most probable one as result. Following the recommendations in [Duda et al. 2000], probabilities for unseen features were smoothed using Lidstone's law. We estimated the parameter on the basis of test runs with features on the sentence level and set it to 0.15.

4 Experiments and Results

To evaluate our classifiers and feature sets, we used holdout cross-validation: The questions were randomly split up into 10 sub-samples for training and test. We evaluated accuracy and also considered precision and recall.
We conducted various experiments with our three classifiers and also compared the results with the accuracy of the corresponding classifiers in the WEKA toolkit. Interestingly, Naïve Bayes slightly outperforms the Decision Tree and the kNN classifier – as Table 4 shows – in almost all cases. However, considering all combinations the algorithms do not differ very much. We found that there is no significant difference in performance between WEKA and our classifiers.

Table 4. Accuracy of Our Three Classifiers

	Decision Tree		Naïve Bayes		kNN	
Feature sets	500	1370	500	1370	500	1370
Baseline	0.37	0.30	0.512	0.48	0.47	0.50
Baseline, bag-of-words	0.45	0.56	0.59	0.61	0.48	0.53
Baseline, named entities	0.48	0.32	0.51	0.50	**0.62**	0.51
Baseline, statistical collocations	**0.54**	0.42	0.63	0.61	0.60	**0.63**
Baseline, statistical collocations, named entities	0.48	0.45	0.63	0.60	0.60	0.58
Baseline, statistical collocations, bag-of-words	0.52	0.59	**0.65**	**0.65**	0.57	0.58
Baseline, statistical collocations, bag-of-words, named entities	0.44	0.50	0.62	0.62	0.53	0.54
Baseline, intuitive collocations, bag-of-words	0.51	**0.60**	0.63	0.64	0.57	0.50
Baseline, intuitive collocations, bag-of-words, named entities	0.44	**0.60**	0.61	0.64	0.52	0.52

To evaluate the various features, we merged them to eleven sets (nine are shown in Table 4) which we examined separately. Our baseline consists of the handpicked trigger words, the question length, and punctuation mark. As Table 4 reveals, the baseline system already performs – with an accuracy of about 45 % – reasonably well. We then stepwise added one or more features. The results are shown in Table 4. The shallow features such as bag-of-words and statistically collected collocations (as opposed to the intuitive ones) make the most important contribution to enhance the performance. Contrary to the accepted opinion named entities and intuitive collocations do not help that much. They even seem to corrupt the performance, as Table 4 shows. Probably, this is due to the fact, that the statistical collocations and bag-of-words already cover information beyond named entities and intuitive collocations. That is to say, these shallow features cover the hand coded and additionally extend the information included in the features further. Comparing the accuracy calculated on the basis of the small test set (500 questions) with the bigger one (1370 questions) clearly shows that all classifiers achieve a generalization over the given data set.

5 Discussion

Even though we had relatively little training data the results were sufficiently accurate for the use in a German question-answering system. However, accuracy is still low for small classes, as they are for classes without specific question words. Nevertheless, we feel confident that small classes can either be handled by re-training with a larger amount of data or by integrating the concept of sub-classes/super-classes in our experiments (e.g. location is the super-class of location.mountain). In any case, as data collection continues during the course of the project (SmartWeb) we expect that even for under-represented classes accuracy will increase further. Although some easily predictable classes as e.g. "wer" (=who) almost always ask for a person, there are classes that do not have a specific question word: *Nenne die Mannschaft, in der Beckenbauer als letztes gespielt hat* (*Name Beckenbauer's last soccer team as active player*). We regard the abstract answer types (*Was ist der Unterschied zwischen Weizen- und Roggenpflanzen? What is the difference between wheat and rye plants?*) as another important challenge. Those two question categories are the most difficult ones for our algorithms. In our opinion, there are mainly two possibilities to solve this problem: These questions hopefully either match a sentence pattern or in case our corpus sufficiently grows in the near future may be catched by means of the bag-of-words strategy. We also plan to further explore these classes to distinguish between the fundamental types that need to be correctly handled by our system and noise. During the course of our experiments we continued to manually construct more sentence patterns matching the questions in our corpus. We did not consider them, yet, to avoid overfitting. However, the patterns may improve the performance – especially for the small classes. Sentence patterns are part of many question-answering systems for English. Although they still have to prove usefulness for German data, what we had to leave for future investigation. We also intend to further improve our answer type ontology. We found that the questions (and answers) in our corpus often do not entirely meet its concepts. In addition, there are questions that the answer type ontology does not cover at all. Especially the abstract types are under-represented. We plan to integrate the results of these experiments into the question analysis component of a German question-answering system. We hope to thus improve the performance of the hole system. While there is still room for improvement, we think – considering the complexity of the task – the achieved performance is surprisingly good.

References

[Wahlster 2004] WAHLSTER, Wolfgang (2004): SmartWeb: Mobile applications of the Semantic Web. In: Adam, Peter / Reichert, Manfred, editors. *INFORMATIK 2004 - Informatik verbindet, Band 1. Beiträge der 24. Jahrestagung der Gesellschaft fr Informatik e.V. (GI), Ulm.* (Web: www.smartweb-project.de)

[Cramer et al. 2006] CRAMER, Irene / LEIDNER, Jochen L. / KLAKOW, Dietrich (2006, to appear): Building an Evaluation Corpus for German Question Answering by Harvesting Wikipedia. *Proceedings of The 5th International Conference on Language Resources and Evaluation, Genoa, Italy.*

[Li et al. 2002] LI, Xin / ROTH, Dan (2002): Learning Question Classifiers. *COLING'02.*

[Li et al. 2002] LI, Xin / HUANG, Xuan-Jing / WU, Li-de (2005):Question Classification using Multiple Classifiers. *Proceedings of the 5th Workshop on Asian Language Resources and First Symposium on Asian Language Resources Network.*

[Zhang et al. 2003] ZHANG, Dell / LEE, Wee Sun (2003): Question classification using support vector machines. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, Toronto, Cananda.*

[Day et al. 2005] DAY, Min-Yuh / LEE, Cheng-Wei / WU, Shih-Hung / ONG, Chorng-Shyong / HSU, Wen-Lian (2005): An Integrated Knowledge-based and Machine Learning Approach for Chinese Question Classification. *IEEE International Conference on Natural Language Processing and Knowledge Engineering.*

[Sonntag et al. 2006] SONNTAG, Daniel /ROMANELLI, Massimo (2006, to appear): A Multimodal Result Ontology for Integrated Semantic Web Dialogue Applications. *Proceedings of the 5th international conference on Language Resources and Evaluation, Genoa, Italy.*

[Voorhees 2001] VOORHEES, Ellen (2001): Overview of the TREC 2001 Question Answering Track. *Proceedings of the 10th Text Retrieval Conference, NIST, Gaithersburg, USA.*

[Witten et al. 2000] WITTEN, Ian H. / FRANK, Eibe (2000): *Data Mining - Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann Publishers. (Web: http://www.cs.waikato.ac.nz/ ml/weka)

[Duda et al. 2000] DUDA, Richard O. / HART, Peter E. / STORK, G. (2000): *Pattern Classification.* New York: John Wiley and Sons.

[Mitchell 1997] MITCHELL, Tom M. (1997): *Machine Learning.* Boston, MA: McGraw Hill.