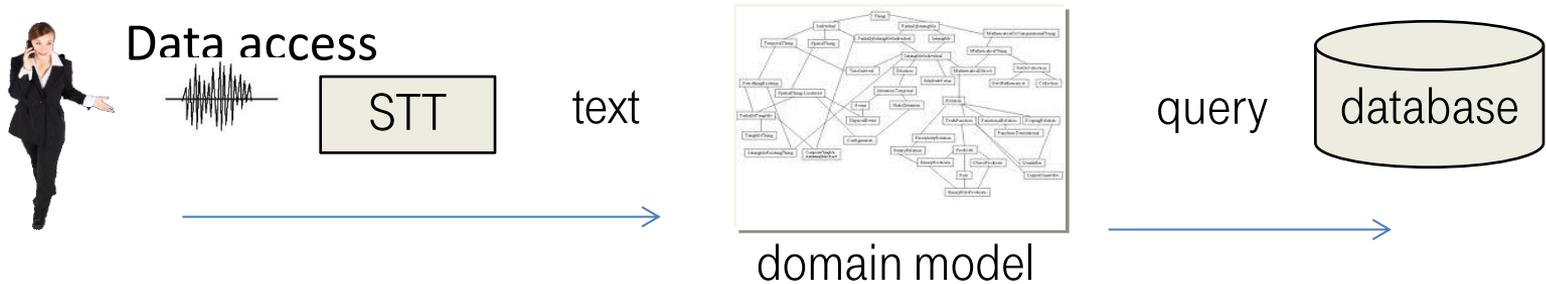


# Search by Voice



# Overview Search by Voice

- „Search by Voice “: content accessible by Voice
- Implies natural language processing
- Possibly includes acoustic content display (Text to Speech)
- Involves three steps:
  1. Speech to text
  2. Query expansion, semantic interpretation

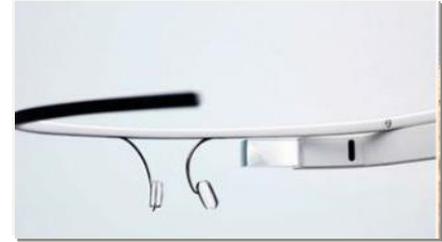


## Excursion: Natural language parsing

- Old approach
  - Grammar based ASR
  - Recognizes only what is defined in the grammar
  - Fills slots defined in the grammar
- New approach
  - Statistical based ASR
  - Recognizes arbitrary text
  - Needs later text parsing for semantic interpretation

# Voice interaction: the next big thing

- 1. Widespread use of mobile devices
  - Smartphones, Glass, Wearables, Implants...
- 2. TV becomes central entertainment center
- means the internet is 24/7 everywhere around us
- Keyboard is in many situations difficult to access,
  - device too small,
  - room too large
  - busy while driving
- Solution is voice control, speech recognition,
- beneath other multimodal inputs like silent voice, face recognition, body movements, biosignals, ...



# Evaluation measures: recall versus precision

- Recall: how many results do I get?
- Precision: how often did I get the right result?



Example: fishing in a lake for trouts.

- Recall: relation of caught trouts to all trouts
- Precision: relation of caught trouts to number of catches
- $F1 = 2 * \text{precision} * \text{recall} / (\text{precision} + \text{recall})$

# Tool: Gate

- We use GATE to annotate terms in user queries, based on gazeteers.
- Each string gets annotated with Part-of-Speech, lemma and NER
- Via JAPE grammars, annotations can be expanded
- Machine learning is also possible

The screenshot shows the GATE Developer interface. The main window displays a document editor with the text: "Zeig mir alle weißen Handies von Samsung zwischen 20 und 200 Euro". The text is annotated with various tags: "Zeig" (Action), "mir" (AnswerType), "alle" (Color), "weißen" (Color), "Handies" (Device), "von Samsung" (Device), "zwischen" (Color), "20" (Color), "und" (Color), and "200" (Color). The bottom of the window shows a table of annotations and a list of checked features.

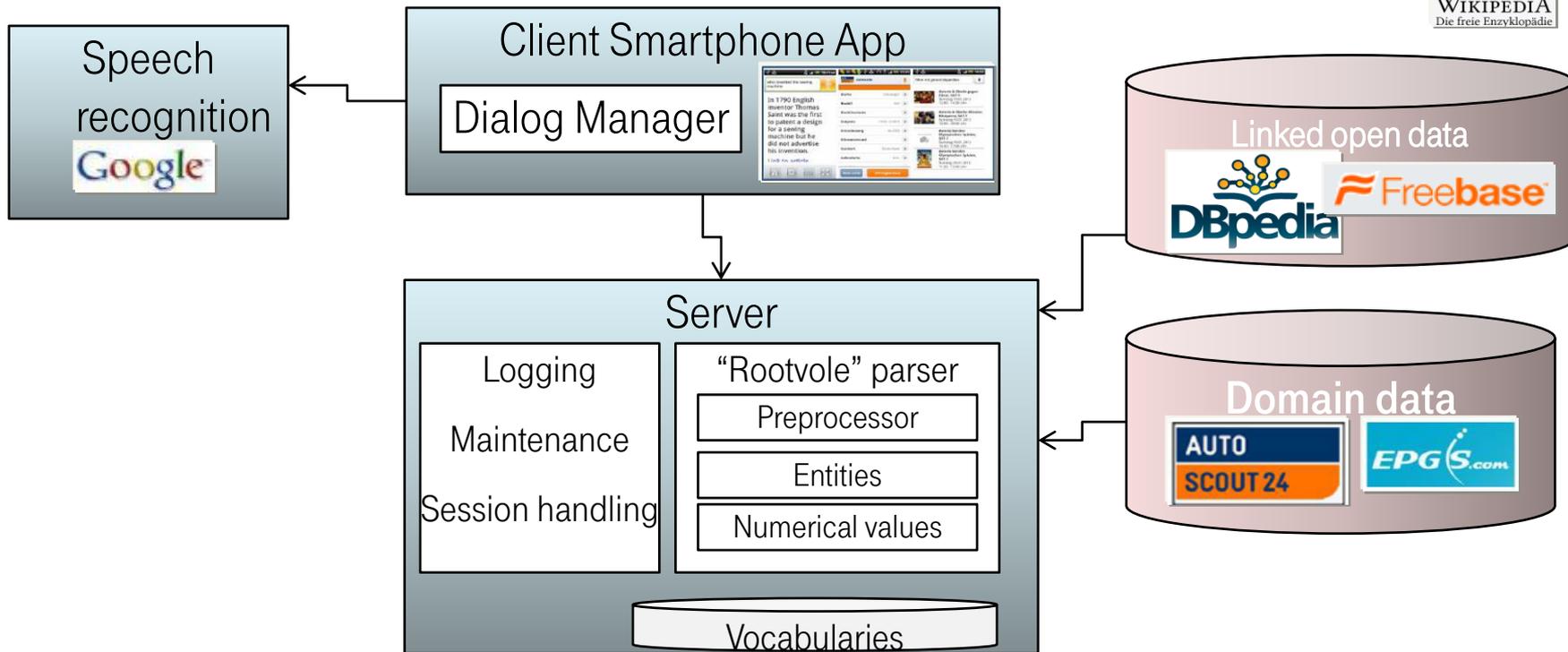
Type	Set	Start	End	Id	Features
Action		0	4	132	{rule=Show, type=show}
SPARQL		14	53	138	{sparql=FILTER (?price >= 20 && ?make = "samsung" && ?color = "weiß" && ?type = "smartph
AnswerType		14	53	139	{type=device}
Color		14	20	133	{color=weiß, rule=Colors}

7 Annotations (0 selected) Select: [ ]

Document Editor Initialisation Parameters Relation Viewer

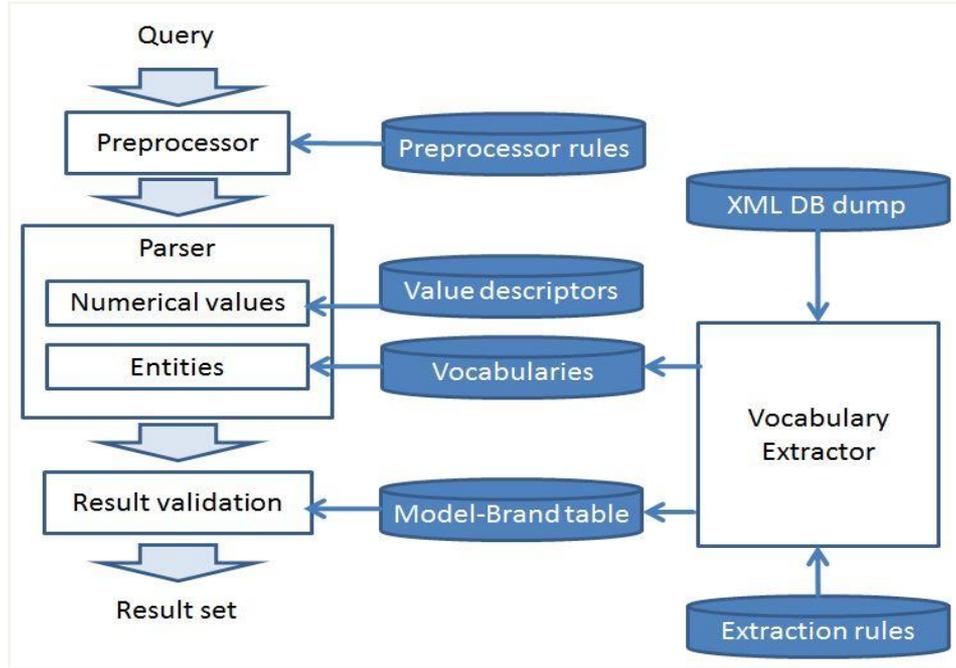
QAAnnotation-DE run in 0.059 seconds

# General architecture Mobile Voice Search



# Text Parser library: “Rootvole”

- Java library for an island parser
- Parses input for
  - Entities (String)
  - Values (Values)
  - rest string



## Rootvole 2

Numerical values are described by

Unit string, e.g. “euro”

Pre- or postfix unit, “Baujahr 2008” vs “300 euro”

Min- max string, e.g. “wenigstens, höchstens”

Can be a region, e.g. “between X and Y”

# Rootvole 3

Entities are stored in vocabularies

Format

<id> <synonym\_1>...<synonym\_N>

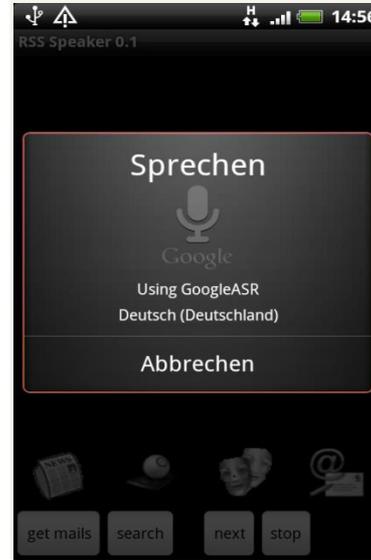
Example parsing result:

```
parsing input krimi im zdf (krimi im zdf)
found: got 6 entities and 0 values, (rest: im ) : [genres: krimi (Krimi)] [genres: krimi (Krimi-Serie)] [genres: krimi (Krimiserie)] [genres: krimi (Polizeifilm)] [senders: zdf (ZDF)] [actors: zdf (Eine pornographische Beziehung)]
```

# RSSReader

## General Work mode

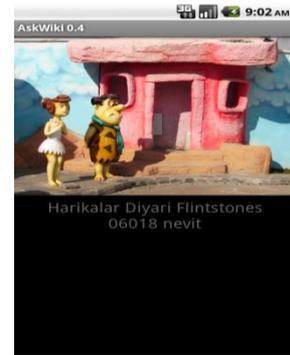
- The news reader (RSS-Reader) is an Android Application that reads RSS-Feeds (short news items) from the Internet.
- Features:
  - News items can be search for with (spoken) keyword spotting.
  - Keyword recognition by Google ASR
  - Text to Speech technology (provided by the phone) reads the articles.
  - E-mail reading integrated (with K9 interface)
  - Three categories: News, Sports, Culture
  - No server connection needed (new items can be read off line)



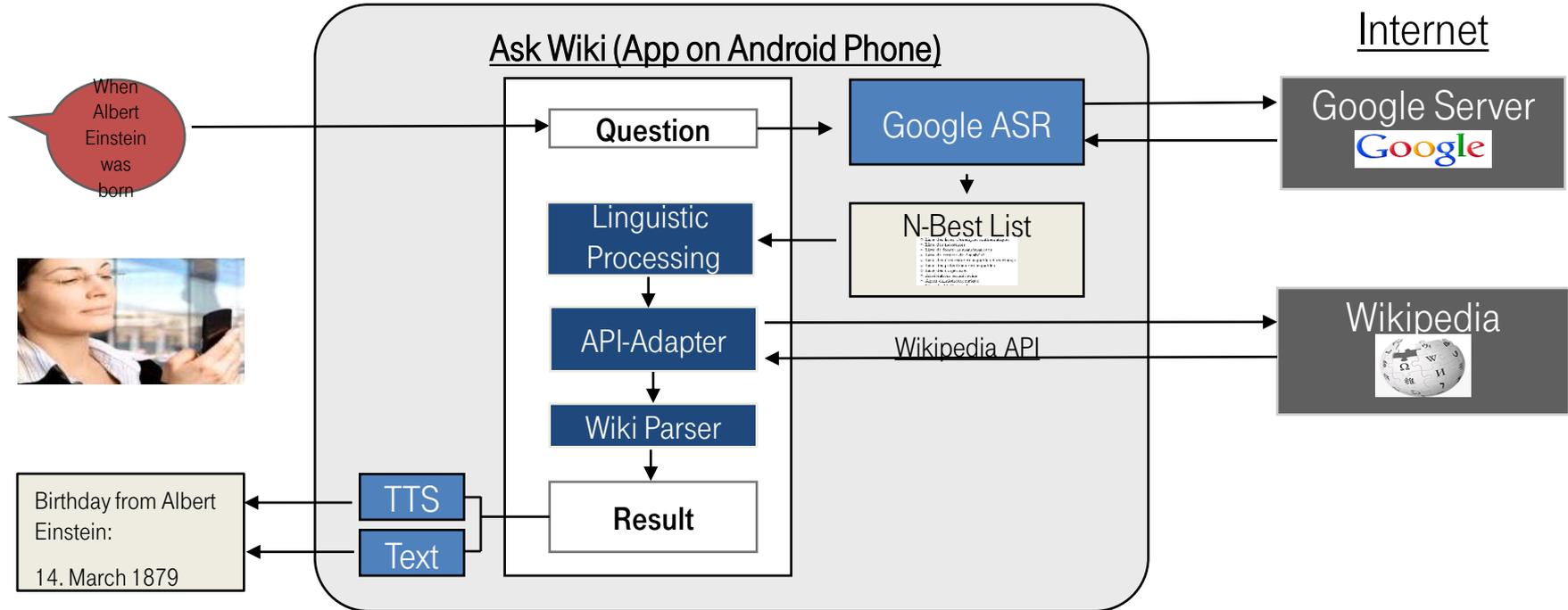
# AskWiki: overview

## General Work mode

- AskWiki allows users to ask short factual questions.
- The Google Speech Recognizer integrated in the Android operating system allows for speech input.
- Sample queries would comprise
  - “Leiter von Siemens” (*Head of Siemens*)
  - “Vorwahl von Rumänien” (*area code of Romania*)
- It is possible to query for features (e.g. „*Bürgermeister* von Berlin“) or sub parts (e.g. „*Nordflügel* Schloss Gottorf“)
- Disambiguation can be resolved by naming the category, e.g. „*Nashville Film*“
- The application runs completely on the smartphone without the need to host an additional server



# AskWiki: architecture



# AskWiki: domain model

## Query model

- A query consists of
  - Stop words that may be ignored
  - A feature, e.g. “date of birth”
  - A specifier, e.g. “physicist”
  - The target article’s title
  - Ambiguous words that can be stop words as well as part of the target

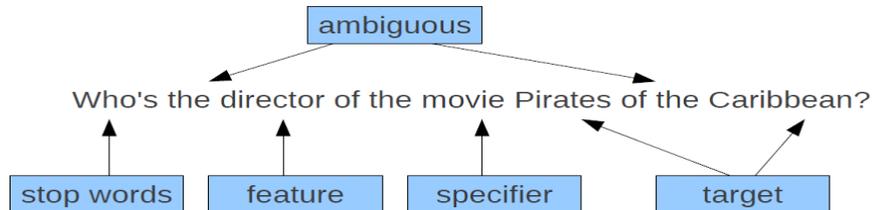


Fig 1: Tokenization of an input phrase

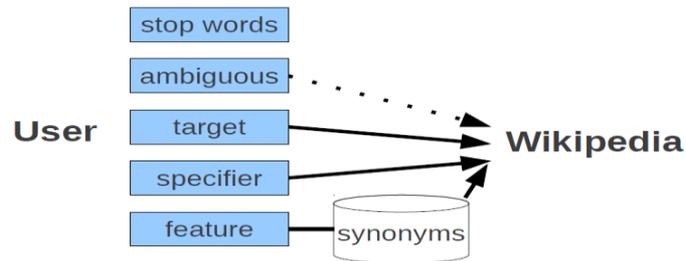
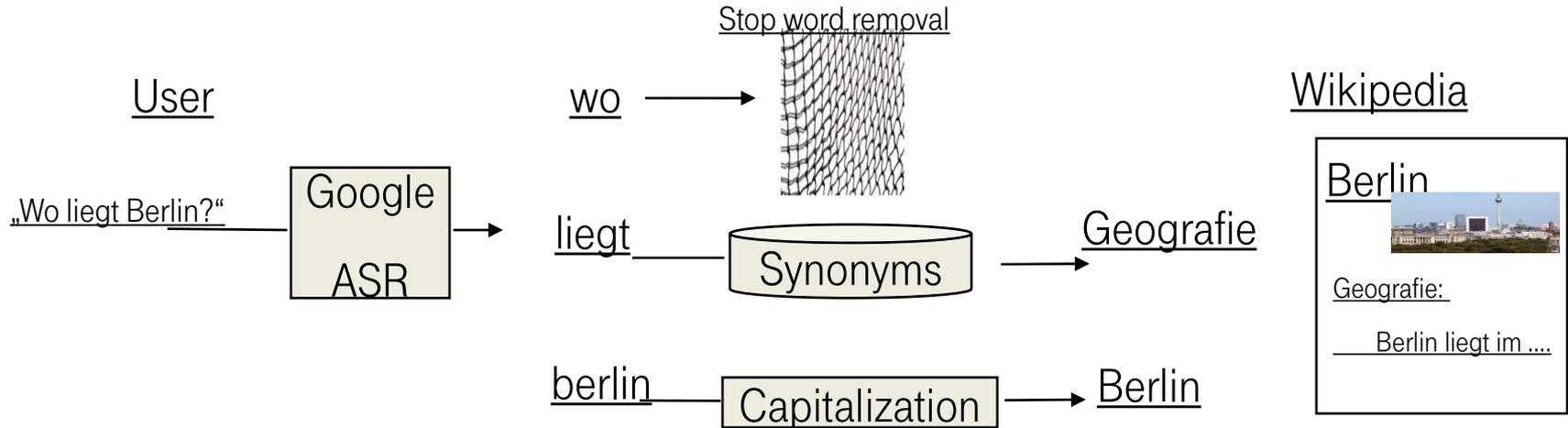


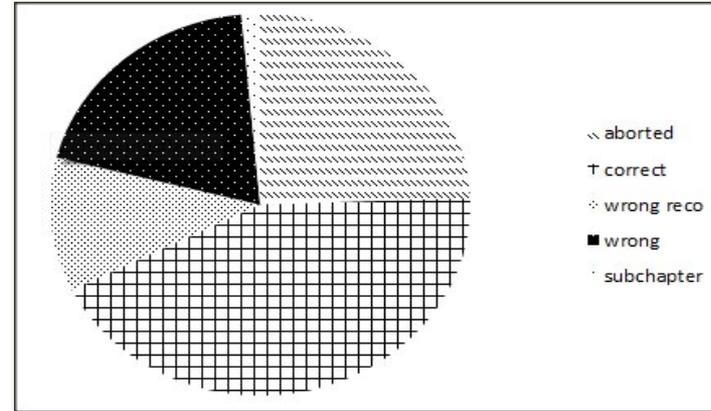
Fig 2: Some words get filtered, others get transformed

# AskWiki: example

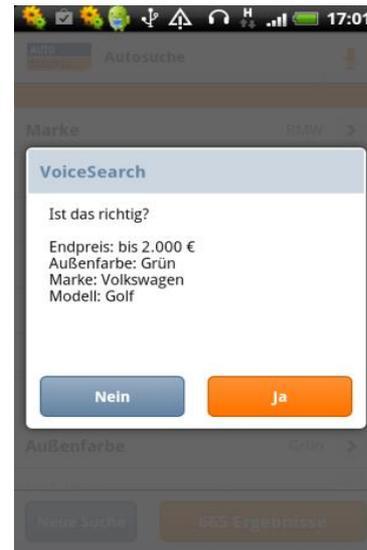


# AskWiki: evaluation

- collected 1052 queries, labeled manually with the following categories.
- The search was aborted (24,4 %) by the user, which means probably that the ASR result was wrong.
- In most of the cases (43,16 %), the answer is correct when the App gets tuned, i.e. sometimes an acronym or synonym had to be added to the vocabulary.
- In 11.4 % of the cases, the ASR result was obviously wrong, i.e. no recognizable question could be detected.
- In 19.58 %, the answer was wrong or could not be detected.
- Only 1.43 % were queries for sub chapters.

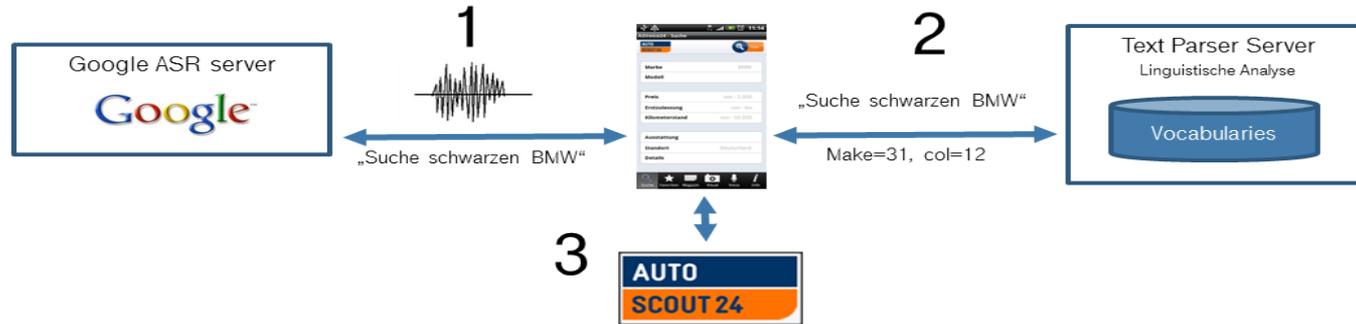


# AutoScout24



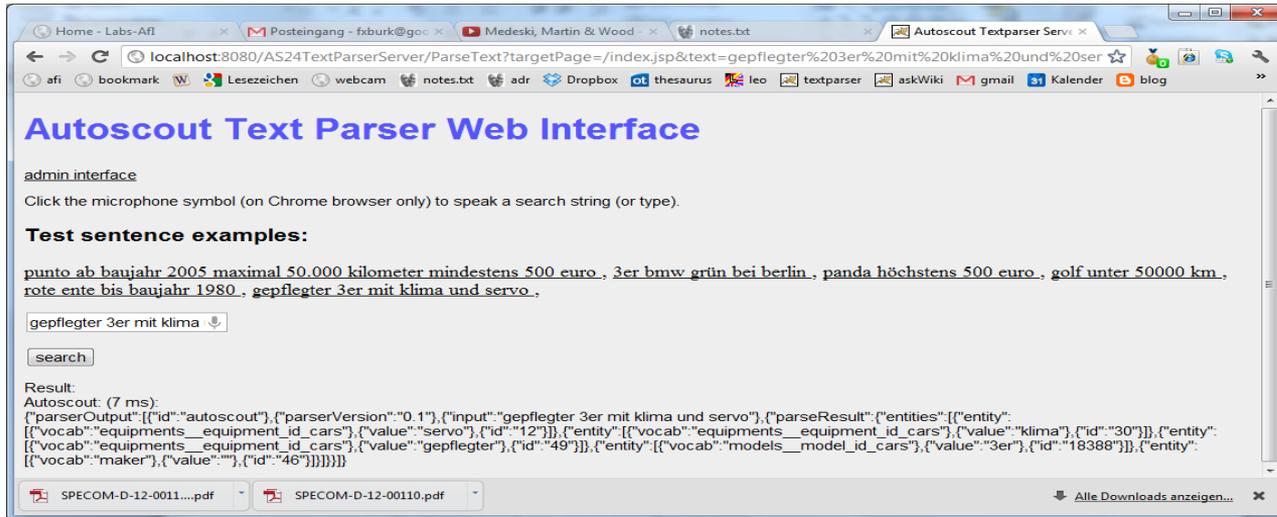
# AutoScout24: overview

- The AS24 Text parser parses natural language queries for car-related words to be used for a voice frontend to the AS24 car search App.
- The admin interface is used primarily to tune the vocabularies of the parser.



The user speech input is recognized by Google ASR (on Android Phone), gets interpreted by the Text Parser Server and the interpretation is then sent to the AS24 database to execute the search.

# AutoScout24: overview test interface



- The test interface is used to enter queries and see the JSON response.
- Input can be speech, using the Google speech recognizer, when running in Chrome Browser
- From here, link at the top leads to the admin interface

# AutoScout24: overview admin interface

## Autoscout Text Parser Administration Web Interface

test interface

### Commands

Re-Initialize parser

Extract vocabularies from AS24 XML file

Refresh vocabs from disk

### Files

- [config file](#)
- [AS24 XML file](#)
- [transformation rules](#)
- [transformation vocab](#)
- [bodies\\_body\\_id\\_cars](#)
- [body\\_paintings\\_body\\_painting\\_id\\_cars](#)
- [body\\_colorgroups\\_body\\_colorgroup\\_id\\_cars](#)

- The administration interface is used to maintain the vocabularies
- There are two parts:
  - **Commands:** execute commands on the server
  - **Files:** edit / view configuration files

# AutoScout24: files: AS24 XML file

The screenshot displays the AutoScout24 web interface for editing the AS24 XML file. On the left, there is a sidebar with a 'Commands' section containing three buttons: 'Re-Initialize parser', 'Extract vocabularies from AS24 XML file', and 'Refresh vocabs from disk'. Below this is a 'Files' section with a list of files: 'config file', 'AS24 XML file', 'transformation rules', 'transformation vocab', 'stopwords', and 'min words'. The main area shows the 'suchparameter.xml' file with a 'write changes to disk' button. The XML content is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <code>0</code>
  <msg>OK</msg>
  <notice>Notice: You are using a permaToken as access key. This
  is valid for development only! Access will be denied with this key
  after end of development. If you are not in development please use a
  regular access key.</notice>
  <searchparameter>
    <bikes>
      <accident_free>
        <label>Hoffallfahrzeuge</label>
```

- AS24 XML file is the basis for the model, maker and other specific vocabularies. They get extracted from this file.

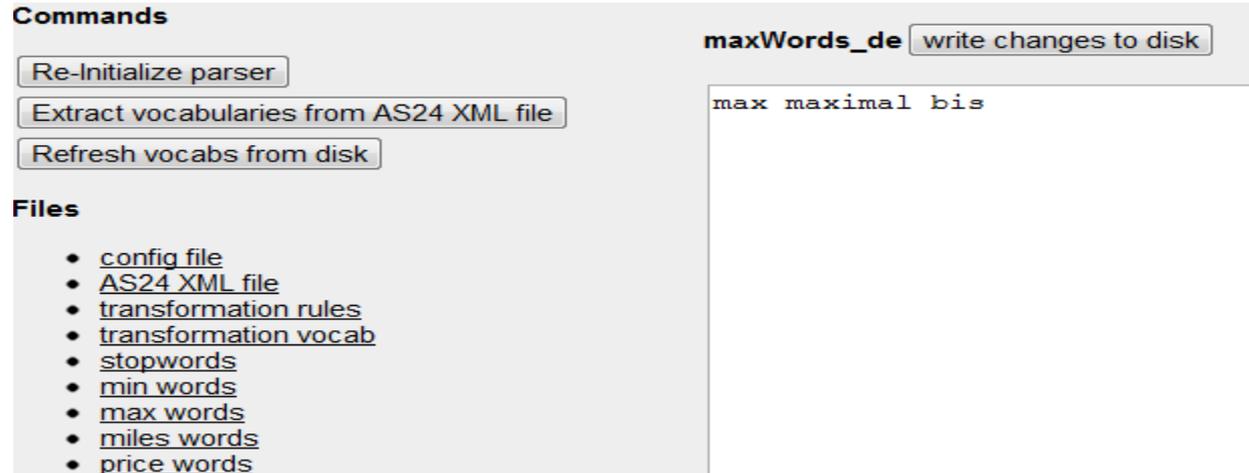
# AutoScout24: files: transformation rules and vocab

The screenshot shows a web interface for testing. On the left, under 'test interface', there are sections for 'Commands' and 'Files'. The 'Commands' section has three buttons: 'Re-Initialize parser', 'Extract vocabularies from AS24 XML file', and 'Refresh vocabs from disk'. The 'Files' section lists several files: 'config file', 'AS24 XML file', 'transformation rules', 'transformation vocab', 'stopwords', and 'min words'. On the right, there is a text area labeled 'vocabLoaderRules.txt' with a 'write changes to disk' button. The text area contains the following transformation rules in Java Pattern Matching syntax:

```
# (bla) -> bla
\ (. * ? \) = $1
# r 11 -> r11
( . + ) ( \ d + ) = $1 $2 , $1 $2
# x - y = x y
( \ s ) * - ( \ s ) * = $1
# SUV / Geländewagen
( \ s + ) \ / ( \ s + ) = $1 , $2
# x / y = x y
( \ s ) * \ / ( \ s ) * = $1
^1 $ = 1 , 1er
^2 $ = 2 . 2er
```

- Transformation rules and vocab are in Java Pattern Matching syntax and are used to convert AS24 XML labels to vocabulary usable for a speech interface.
- Example
  - Input: SUV/Geländewagen
  - Output: suv,geländewagen

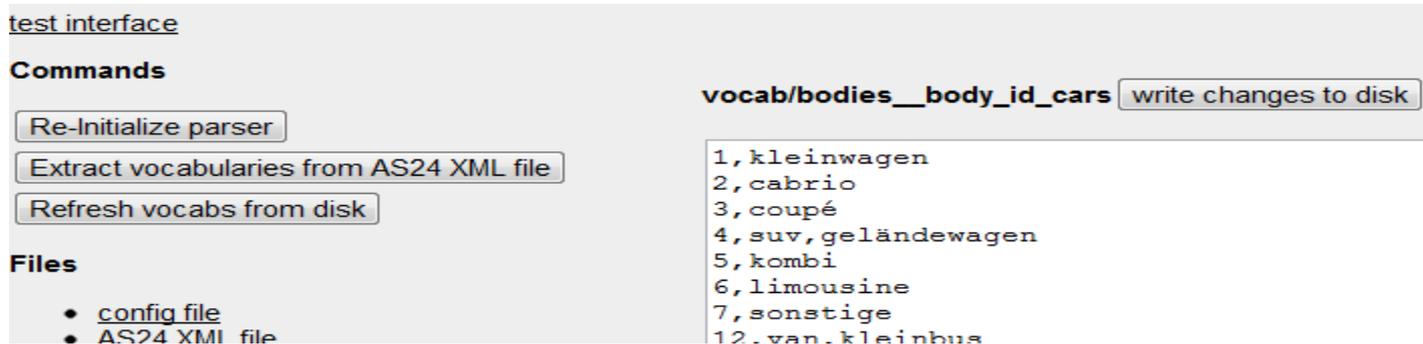
# AutoScout24: files: min, max, miles, price



The screenshot shows a web interface for AutoScout24. On the left, under the heading "Commands", there are three buttons: "Re-Initialize parser", "Extract vocabularies from AS24 XML file", and "Refresh vocabs from disk". Below this is a section titled "Files" with a list of links: [config file](#), [AS24 XML file](#), [transformation rules](#), [transformation vocab](#), [stopwords](#), [min words](#), [max words](#), [miles words](#), and [price words](#). On the right, there is a text input field containing "maxWords\_de" and a "write changes to disk" button. Below the input field, the text "max maximal bis" is displayed.

- These words are used as keywords for boundaries and units.
- Postfix or Prefix notation for units is controlled in the config file:
  - E.g. „miles\_postfix=true“ means
    - „3000 km“ gets recognized, but „km 3000“ not

# AutoScout24: files: entity vocabularies

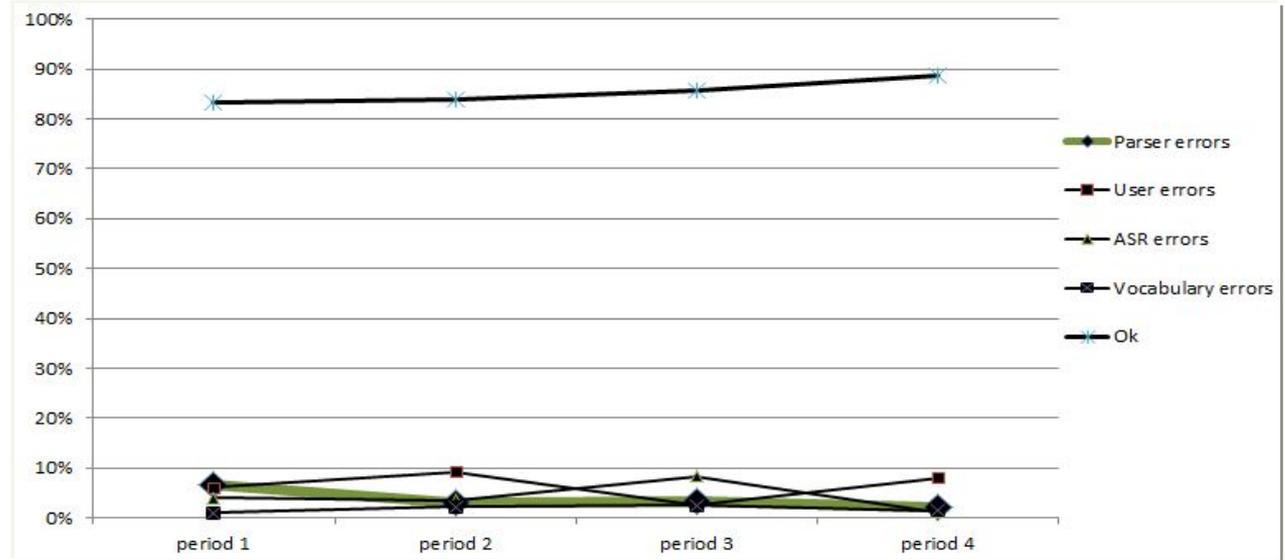


The screenshot shows a web interface for managing entity vocabularies. On the left, under the heading "test interface", there is a "Commands" section with three buttons: "Re-Initialize parser", "Extract vocabularies from AS24 XML file", and "Refresh vocabs from disk". Below this is a "Files" section with two bullet points: "config file" and "AS24 XML file". On the right, there is a text input field containing "vocab/bodies\_\_body\_id\_cars" and a "write changes to disk" button. Below the input field is a list of car body types, each with an ID and a name: "1, kleinwagen", "2, cabrio", "3, coupé", "4, suv, geländewagen", "5, kombi", "6, limousine", "7, sonstige", and "12. van, kleinbus".

- All other files were extracted from the AS24 XML file and can be viewed here for control.
- The vocabulary syntax is: <id>,<synonym 1>,...,<synonym N>
- If the vocabulary has to be changed, it should NOT be done here but by the pattern matching rules.
- Otherwise synchronization problems occur when the vocabulary (e.g. make or color names and/or IDs) from the AS24 database changes.

# AutoScout24: tuning

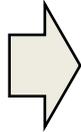
- Measured errors manually in four 1-week periods
- Tuned after each measurement
- Distinguished
  - Parser error
  - ASR error
  - User error
  - Vocabulary error



# TV program guide: actors



Voice Search for  
Entertain



+++ Was kommt heute um 20:00 Uhr?  
+++



**Asterix & Obelix gegen  
Cäsar, SAT.1**

Samstag 19.01.2013  
12:00 - 14:30 Uhr



**Asterix & Obelix: Mission  
Kleopatra, SAT.1**

Samstag 19.01.2013  
18:00 - 20:00 Uhr

Photo Not Available



**Asterix bei den  
Olympischen Spielen,  
SAT.1**

Samstag 19.01.2013  
14:30 - 17:00 Uhr



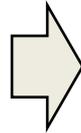
**Asterix bei den  
Olympischen Spielen,  
SAT.1**

Sonntag 20.01.2013  
11:20 - 13:50 Uhr

# TV program guide: stations



Voice Search for  
Entertain



am wochenende in den privaten



**Matrix Revolutions, Pro7**  
Samstag 26.01.2013  
00:40 - 03:00 Uhr

Photo Not Available



**BLOCKBUSTER TV -  
MAKING OF: Flight, Pro7**  
Samstag 26.01.2013  
03:00 - 03:10 Uhr

Photo Not Available



**ProSieben  
Spätnachrichten, Pro7**  
Samstag 26.01.2013  
03:10 - 03:15 Uhr



**Killzone S.P.L., Pro7**  
Samstag 26.01.2013  
03:15 - 04:40 Uhr



**Zenon III - Das Rennen  
zum Mond, Pro7**

+++ Was kommt heute um 20:00 Uhr?  
+++

# TV program guide: times



Voice Search for  
Entertain



+++ Was kommt heute um 20:00 Uhr?  
+++



freitag mittag



Photo Not Available



**Unsere erste  
gemeinsame Wohnung,**  
RTL

Freitag 25.01.2013  
11:30 - 12:00 Uhr



**How I Met Your Mother,**  
Pro7

Freitag 25.01.2013  
11:50 - 12:15 Uhr

Photo Not Available



**Punkt 12 - Das RTL-  
Mittagsjournal,** RTL

Freitag 25.01.2013  
12:00 - 14:00 Uhr

Photo Not Available



**heute, ZDF**

Freitag 25.01.2013  
12:00 - 12:10 Uhr

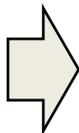
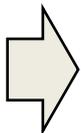
**RICHTER**

**Richter Alexander Hold,**  
SAT.1

# TV program guide: characters



Voice Search for  
Entertain



+++ Was kommt heute um 20:00 Uhr?  
+++



gräfin schönberg



**SOKO Kitzbühel, ZDF**  
Dienstag 29.01.2013  
16:10 - 17:00 Uhr



**SOKO Kitzbühel, ZDF**  
Mittwoch 30.01.2013  
16:10 - 17:00 Uhr



**SOKO Kitzbühel, ZDF**  
Donnerstag 31.01.2013  
16:10 - 17:00 Uhr



**SOKO Kitzbühel, ZDF**  
Freitag 01.02.2013  
16:10 - 17:00 Uhr

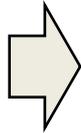


**SOKO Kitzbühel, ZDF**

# TV program guide: keywords



Voice Search for  
Entertain



+++ Was kommt heute um 20:00 Uhr?  
+++



obduktion



**Aufschneider, 3sat**  
Mittwoch 30.01.2013  
20:15 - 21:40 Uhr



**Aufschneider, 3sat**  
Donnerstag 31.01.2013  
02:10 - 03:40 Uhr



**Der letzte Zeuge, Arte**  
Donnerstag 31.01.2013  
20:15 - 21:00 Uhr



**SOKO Kitzbühel, ZDF**  
Montag 04.02.2013  
16:10 - 17:00 Uhr

# TV program guide: detail view



Voice Search for  
Entertain



+++ Was kommt heute um 20:00 Uhr?  
+++

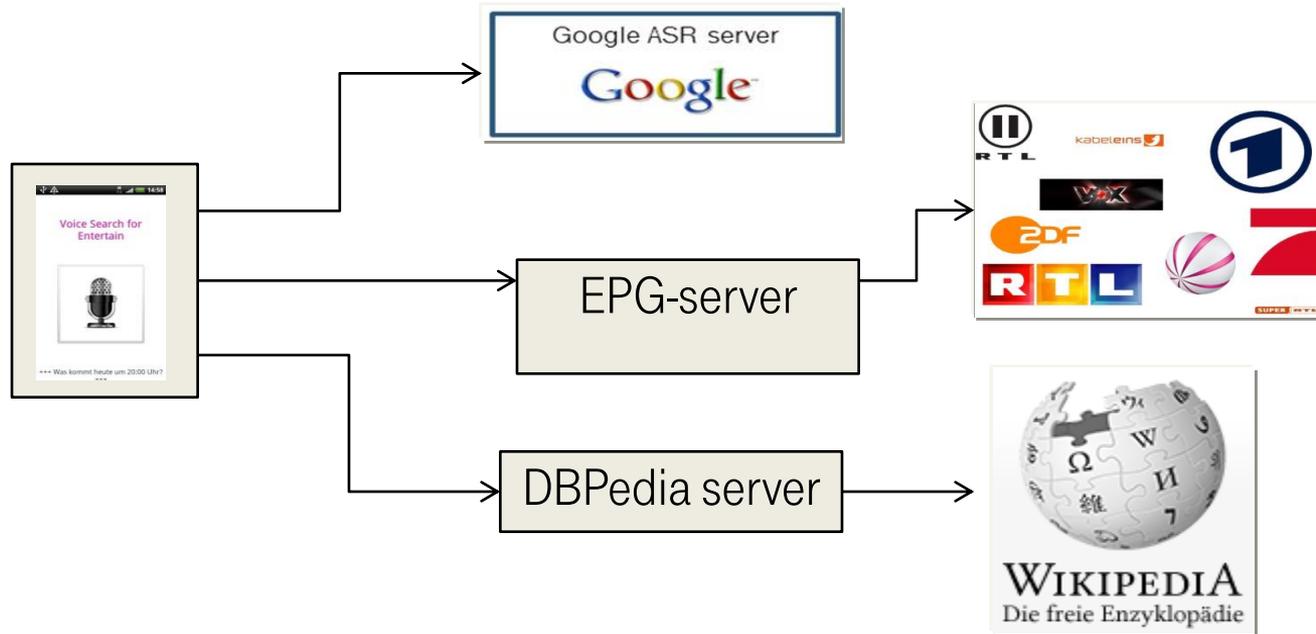


**How I Met Your Mother,  
Pro7**  
Freitag 25.01.2013  
11:50 - 12:15 Uhr

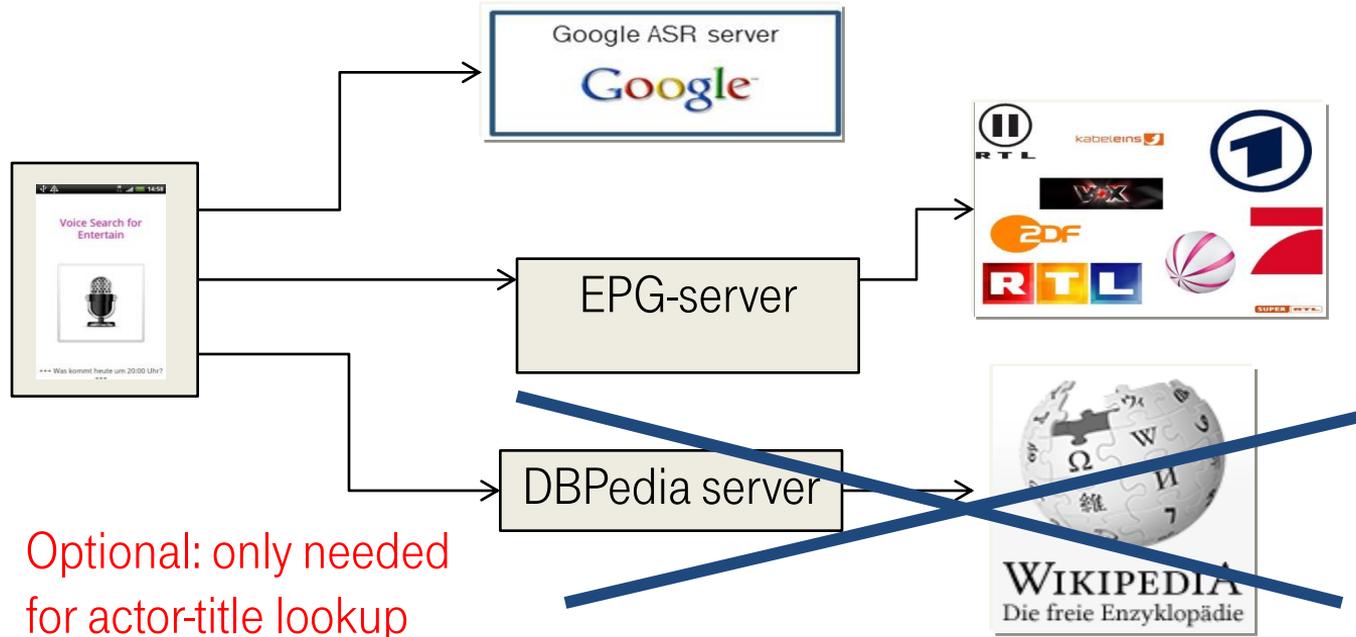
Sitcom

Barneys Mutter Loretta plant, ihr Haus zu verkaufen. Barney und seine Freunde helfen dabei, das Haus auszuräumen. Beim Verpacken diverser Erinnerungstücke entdecken sie einen Brief an einen gewissen Sam Gibbs, der aber nie verschickt wurde. Im Umschlag befindet sich ein Foto von Barney und seinem Bruder James, auf dessen Rückseite Loretta die Worte "dein Sohn" geschrieben hatte. Sofort macht sich Barney mitsamt James und den Freunden auf den Weg zu seinem vermeintlichen Vater.

# TV program guide: architecture



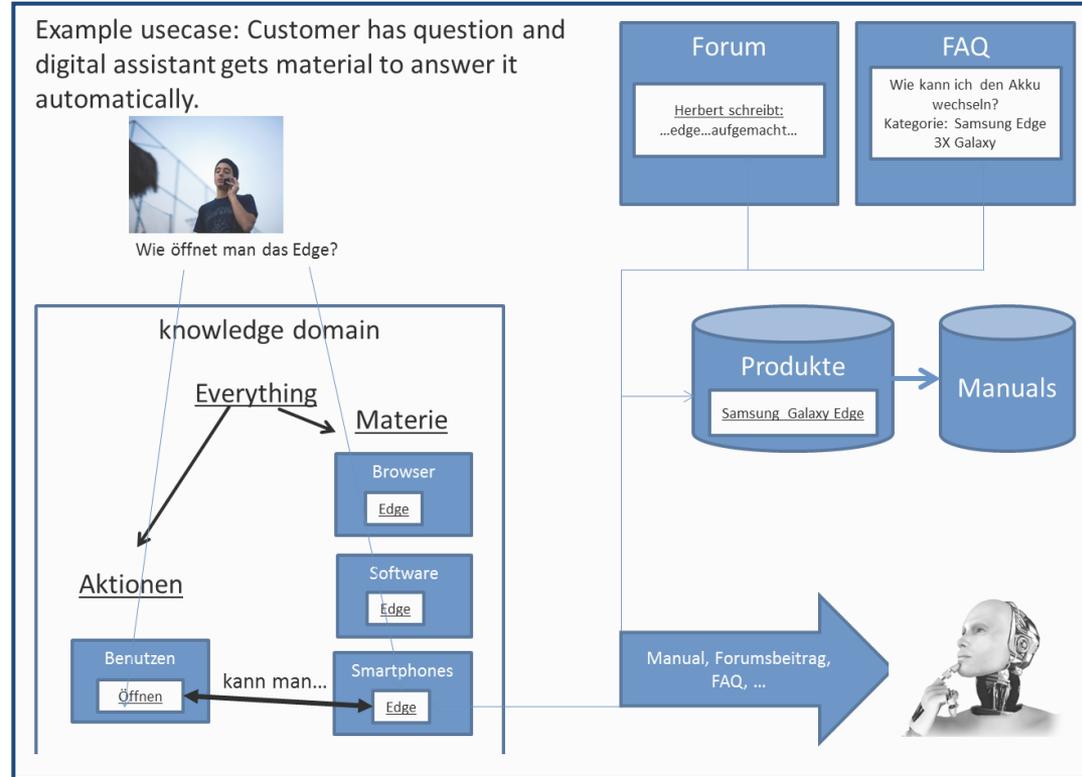
# TV program guide: architecture



# Quark: Architecture for a QA Machine: MOTIVATION

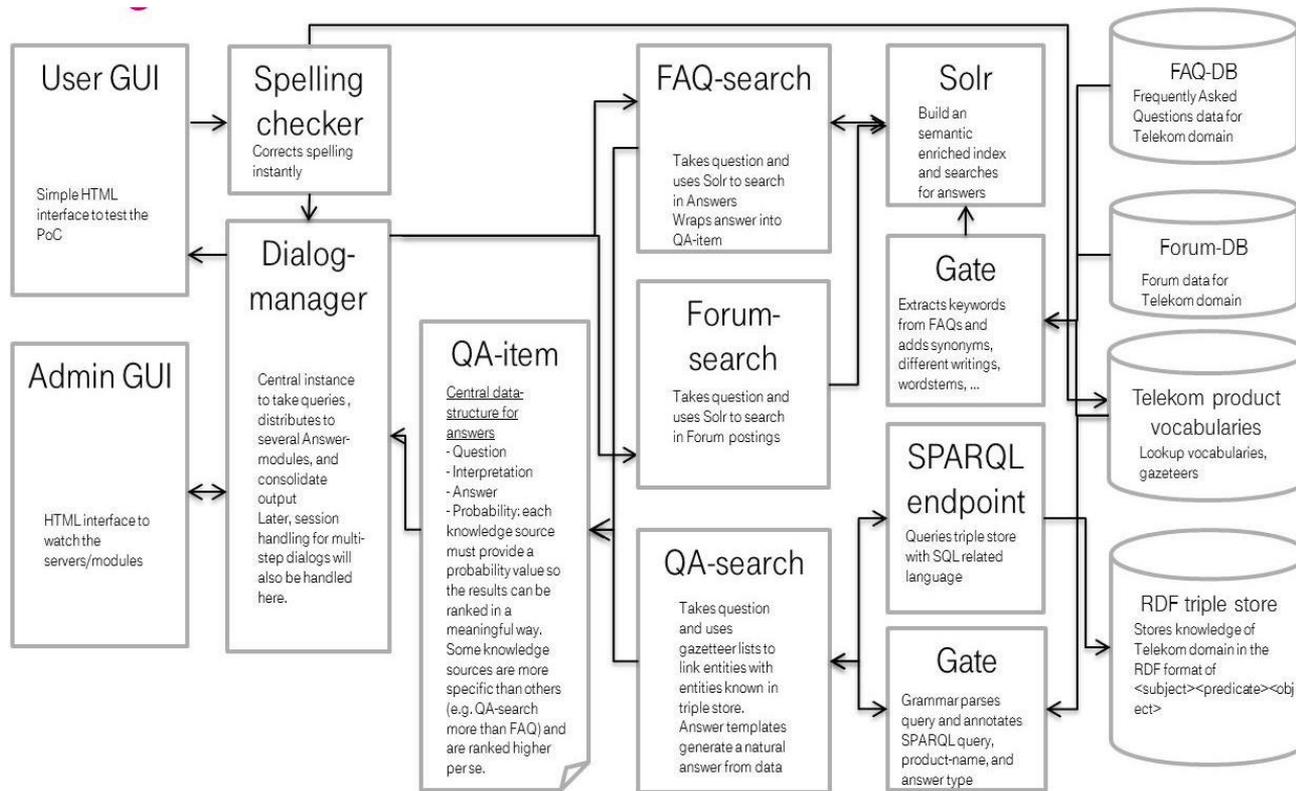
To ease the work of human agents and save costs automatic question answering systems are valuable

One example are so-called „chatbots“, i.e. automatic dialog systems



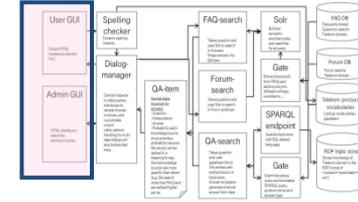
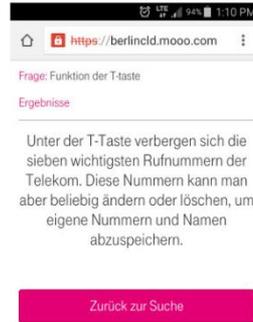
# Quark: Architecture for a QA Machine: ARCHITECTURE

- We developed an architecture to develop, test and compare several components of such a question answering system.
- It is also used to build demonstrators for management



# Quark: Architecture for a QA Machine: FRONTENDS

- There are several interfaces, e.g. mobile apps for demonstration
- Web interfaces for testing and maintenance



## QA\_DM Server Interface

Click the microphone symbol (on Chrome browser only) to speak a search string (or type).

handies vjn samsung ab 250

search

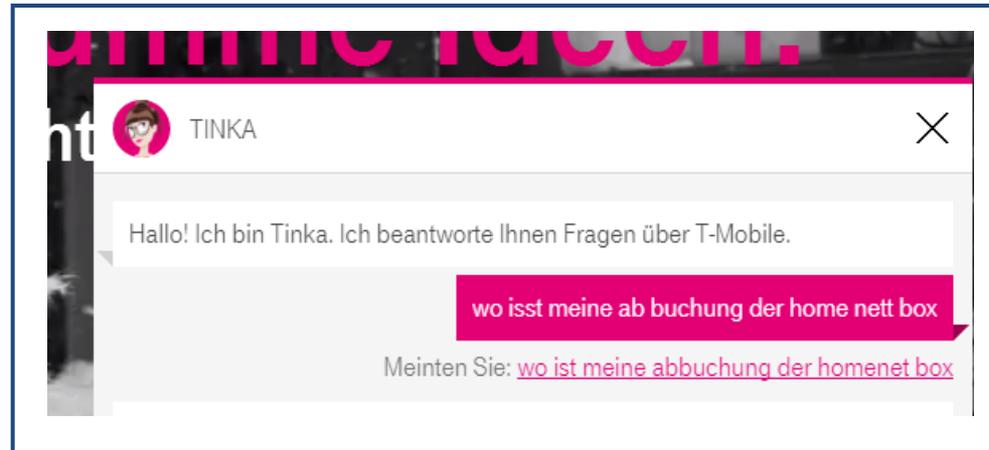


Result: Es wurden 5 Produkte gefunden: Das Samsung Galaxy Note 4 schwarz Samsung Galaxy S6 32 GB schwarz, Samsung Galaxy S6 64 GB schwarz, Samsung Galaxy S6 128 GB schwarz, und das Samsung Galaxy S6 edge 128 GB gold

Used time: 2514

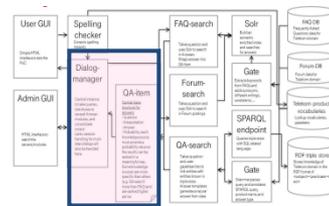
# Quark: Architecture for a QA Machine: SPELLCHECKER

- In the current Tinka implementation we use the open source Hunspell spellchecker
- The API was algorithmically enhanced by supporting
  - Weighted user lexicon
  - Enabling recognition of bi-grams and tri-grams



# Quark: Architecture for a QA Machine: DIALOGMANAGER

- Central instance
- Missing dialog model for slot-filling and ellipse handling
- Main challenge is currently the result list order, i.e. a quality measure for the answers from several moduls



**Dialog-manager**

Central instance to take queries , distributes to several Answer-modules, and consolidate output

Later, session handling for multi-step dialogs will also be handled here.

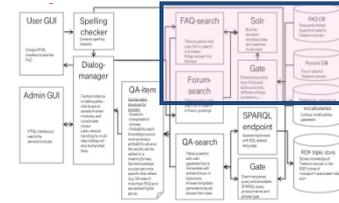
**QA-item**

Central data-structure for answers

- Question
- Interpretation
- Answer
- Probability: each knowledge source must provide a probability value so the results can be ranked in a meaningful way. Some knowledge sources are more specific than others (e.g. QA-search more than FAQ) and are ranked higher per se.

# Quark: Architecture for a QA Machine: SEMANTIC SEARCH

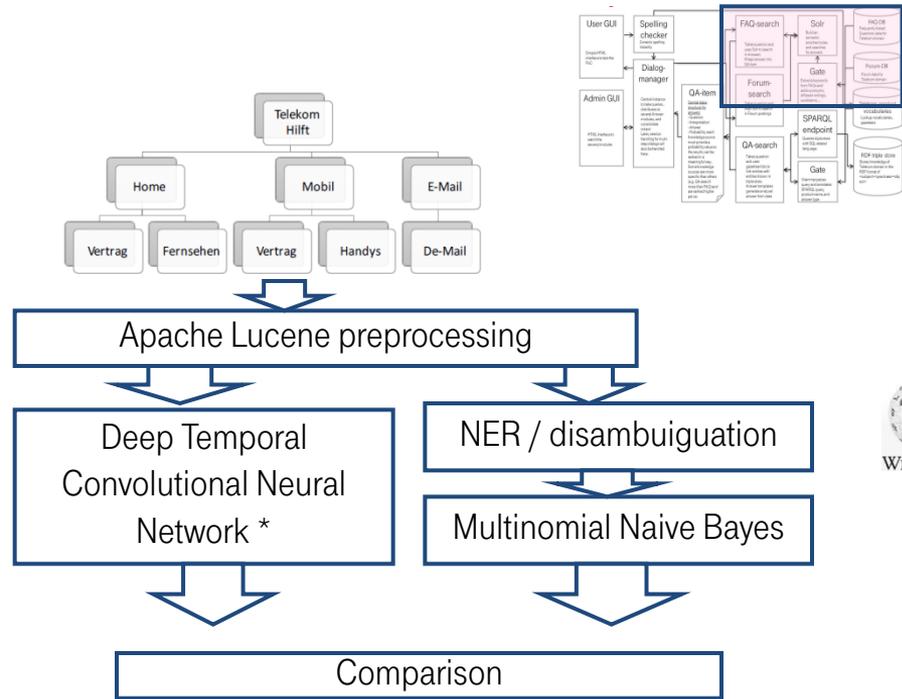
- With content that already contains answers, FAQ and extracted from forum, the user query must only be matched with the question.
- Finding important words and synonyms for „query-expansion“ can be done with GATE’s term extractor, had to be adapted for German.
- The search index of a SOLR search engine can than be enhanced by these terms.
- The number of matching terms would be part of the quality criterion



[9] MAYNARD, D. und W. LI, Y. AND PETERS: *NLP techniques for term extraction and ontology population*. In: BUITELAAR, P. und P. CIMIANO (Hrsg.): *Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, S. 171–199. IOS Press, Amsterdam, 2008.

# Quark: Architecture for a QA Machine: TOPIC CLASSIFICATION OF FORUM DATA

- Together with the DAI (Distributed Artificial Intelligence) Labor of TU-Berlin we investigated the topic classification of „Telekom Hilft“ user forum
- Compared „classical machine learning“ with Deep neural nets.
- Both resulted in 55% accuracy rsp. 83% „one in three“
- Also investigated subclustering with DNN (4 subclusters per category)



\* Zhang, Zhao, LeCun, 2015



# Quark: Architecture for a QA Machine: ANSWER TEMPLATES

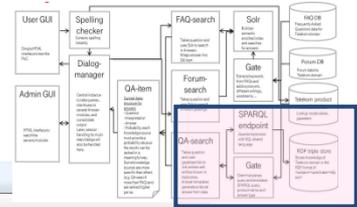
- We also use GATE to annotate terms in user queries, based on gazeteers.
- Each string gets annotated with Part-of-Speech, lemma and NER
- Disambiguation is not done yet
- Via JAPE grammars a pre-defined answer (and question) template is determined
- With this template, a SPARQL database is queried and the answer filled.
- SPARQL is the query language for RDF, a W3C suggestion for semantic annotation

The screenshot shows the GATE Developer interface. The main window displays a text document with the sentence: "Zeig mir alle weißen Handies von Samsung zwischen 20 und 200 Euro". The text is annotated with various tags: "Zeig" (Action), "mir" (AnswerType), "alle" (Color), "weißen" (Device), "Handies" (Money), "von Samsung" (SPARQL), and "zwischen 20 und 200 Euro" (SPARQL). Below the text, a table lists the annotations:

Type	Set	Start	End	Id	Features
Action		0	4	132	{rule=Show, type=show}
SPARQL		14	53	138	{sparql=FILTER (?price >= 20 && ?make = "samsung" && ?color = "weiß" && ?type = "smartph
AnswerType		14	53	139	{type=device}
Color		14	20	133	{color=weiß, rule=Colors}

At the bottom, a table lists the features for each annotation:

Feature	Value
Action	
AnswerType	
Color	weiß
Device	Handies
Device maker	Samsung
Money	20 und 200 Euro
SPARQL	



[6] CUNNINGHAM, H., D. MAYNARD, K. BONTCHEVA und V. TABLAN: *GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications*. In: *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, 2002.

Thanks

[fxburk@gmail.com](mailto:fxburk@gmail.com)