Unlocking the Secrets of the Past

Final Presentation: Mining the "Kabinettsprotokolle der Bundesregierung"

Andreas Schwarte Christopher Haccius
Sven Steudter Sebastian Steenbuck

Text Mining Seminar WS 2009/10 – 6.3.2010

Outline

- Motivation & Introduction
- Our Ideas
- Data Retrieval Techniques
- Methodology & Implementation
- Encountered Problems
- Evaluation and Findings
- Conclusion

Motivation

- Huge amounts of data are available
  - How can you find correlations?
  - How can you query over more dimensions, e.g. time, location, person?
  - What about efficiency?
- Solution: Mining and Processing Data
  - Indices, Semantic Tagging, Dictionaries, Ontologies, etc.

Introduction

- “Kabinettsprotokolle der Bundesregierung”
  - cover time from 1949 to 1964
  - protocols of cabinet meetings
  - about 10.000 articles, i.e. agenda items

Our Ideas

1. Geographical areas of interest over time

   →Finding geographic hot spots for certain time periods

   e.g. which countries are on the agenda during a certain period of time

2. Relevant political topics of interest over time

   → Extract information about topic correlations

   e.g. topics like foreign affairs, health, economic questions

3. Participation of politicans with respect to topic

   → Extract information about politicians and attendence

   e.g. which person attended which topic, was someone important missing

Data Retrieval Techniques

- Crawling Techniques on the Website
  - 10000 RTF-Documents (114MB) + Metadata
  - Conversion to plain text, omit style information
  - Crawling Process took about 4.5 hours
    - Slow Server (Tree Navigation was slow)
  - Half of the items were associated with a ministry
    - Can be used as training material for classification (details later)
- Java Interface provides Access to all data

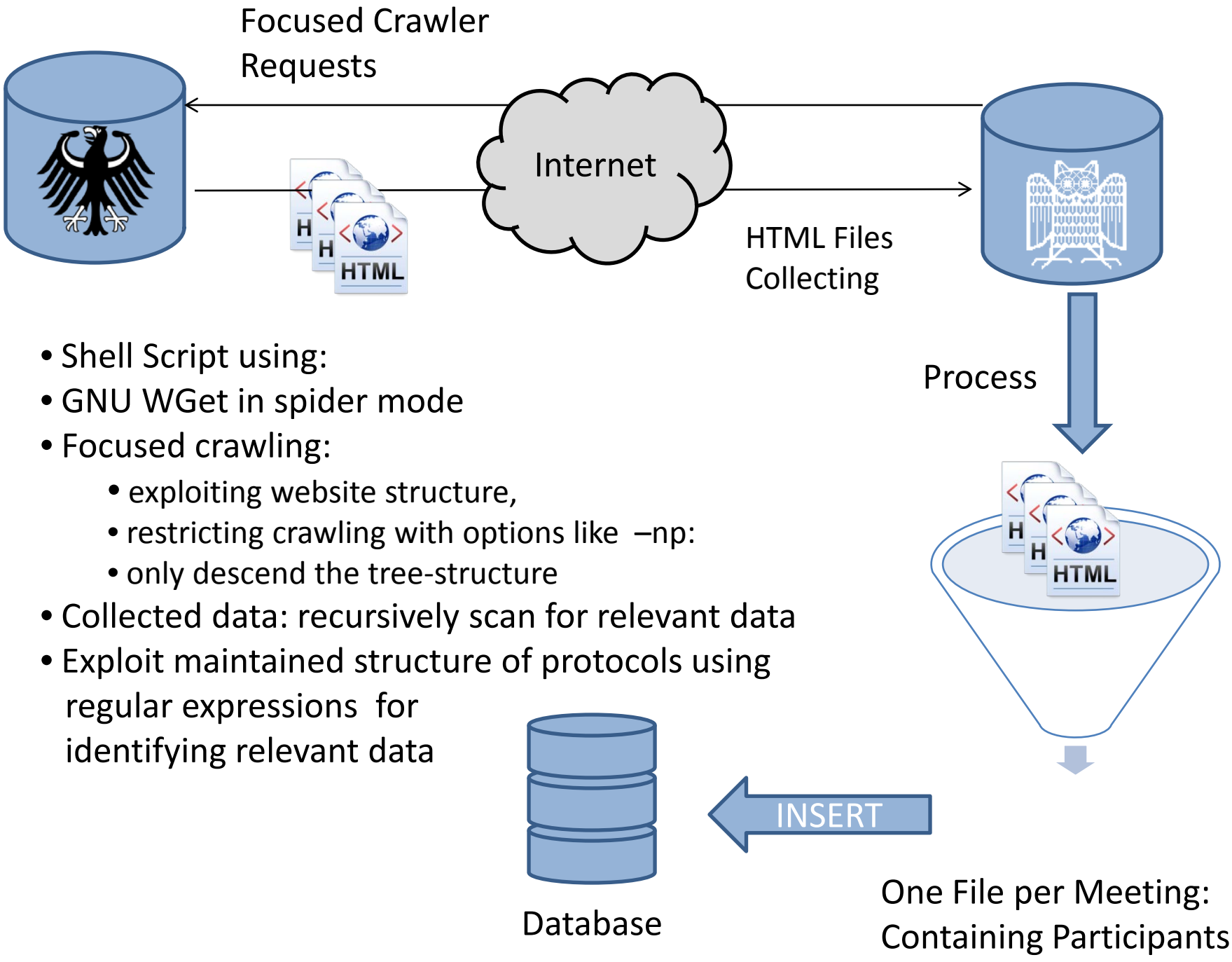
Data Retrieval Techniques

- Retrieval of Countries
  - Needed for Mapping of (*agenda item, countries*)
  - Key Idea
    - Get a complete list of countries
    - Adapt list to time span, i.e. add *countries* like UdSSR
    - Scan input documents for occurrences
  - Possible Improvements
    - Use some form of stemming to recognize variations
    - Include adjective forms as well, e.g. recognize "*der französische Außenminister*"

Data Retrieval Techniques

- Retrieval of Persons – very basic approach
  - Website provides lists of participants
  - At the moment: no entity disambiguation

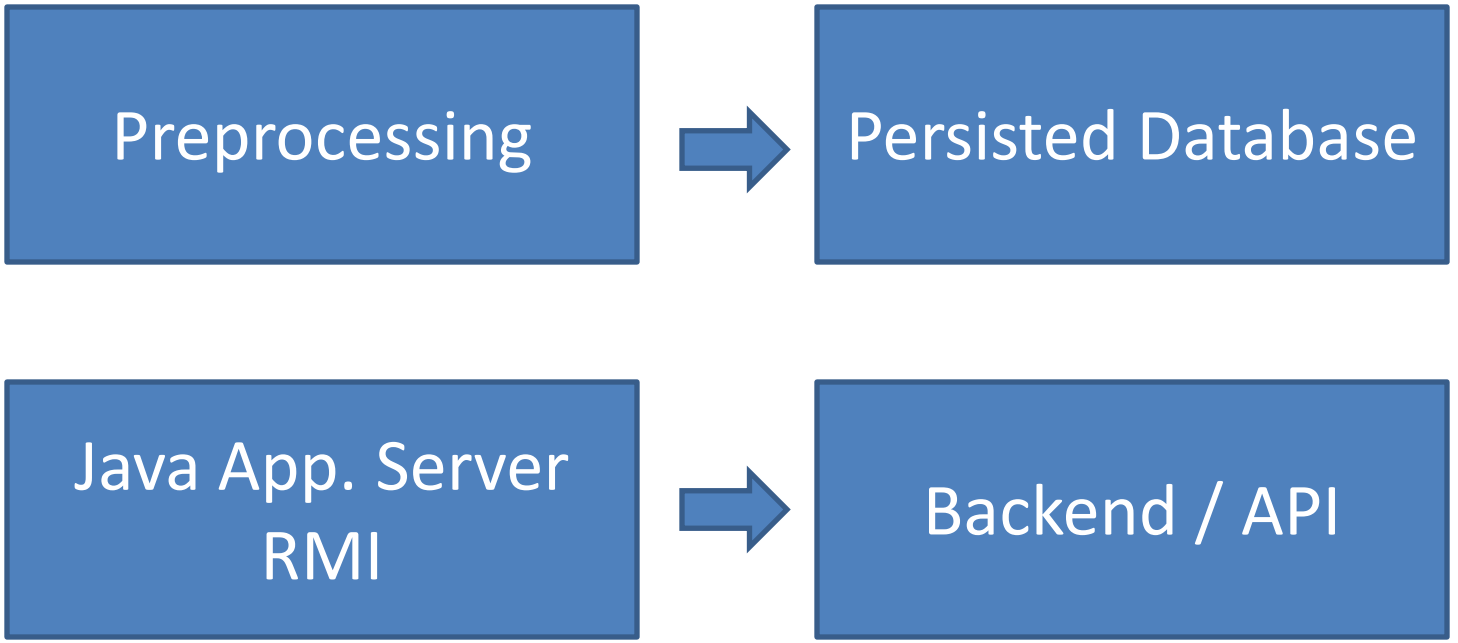
Illustration



- Shell Script using:
- GNU WGet in spider mode
- Focused crawling:
  - exploiting website structure,
  - restricting crawling with options like –np:
  - only descend the tree-structure
- Collected data: recursively scan for relevant data
- Exploit maintained structure of protocols using regular expressions for identifying relevant data

The Implementation Model

A two layer approach



→ **Client can access the backend/API through a Java interface**

Preprocessing

- Huge dataset requires preprocessing
  1. Analysis of Data, Document Set Construction
  2. Construction of the Inverted Index
  3. Topic Classification (details on next slide)
  4. Further Analysis (Idf & Scores, participants, countries)
  5. Persisting the constructed data structures
- Duration: about 45 minutes

Preprocessing - Details

- OpenNLP library used for tokenization
- Stemming is used during index construction
  - Based on Snowball algorithm, e.g. Katze -> Katz
- Classification of agenda items into topics
  - Based on LingPipe API, language model of n-Grams
  - Manually picked Categories, e.g. Wirtschaft, Außenpolitik
  - Training Data generated from our data set
    - Part of the protocols were done by ministries, e.g. BMWi

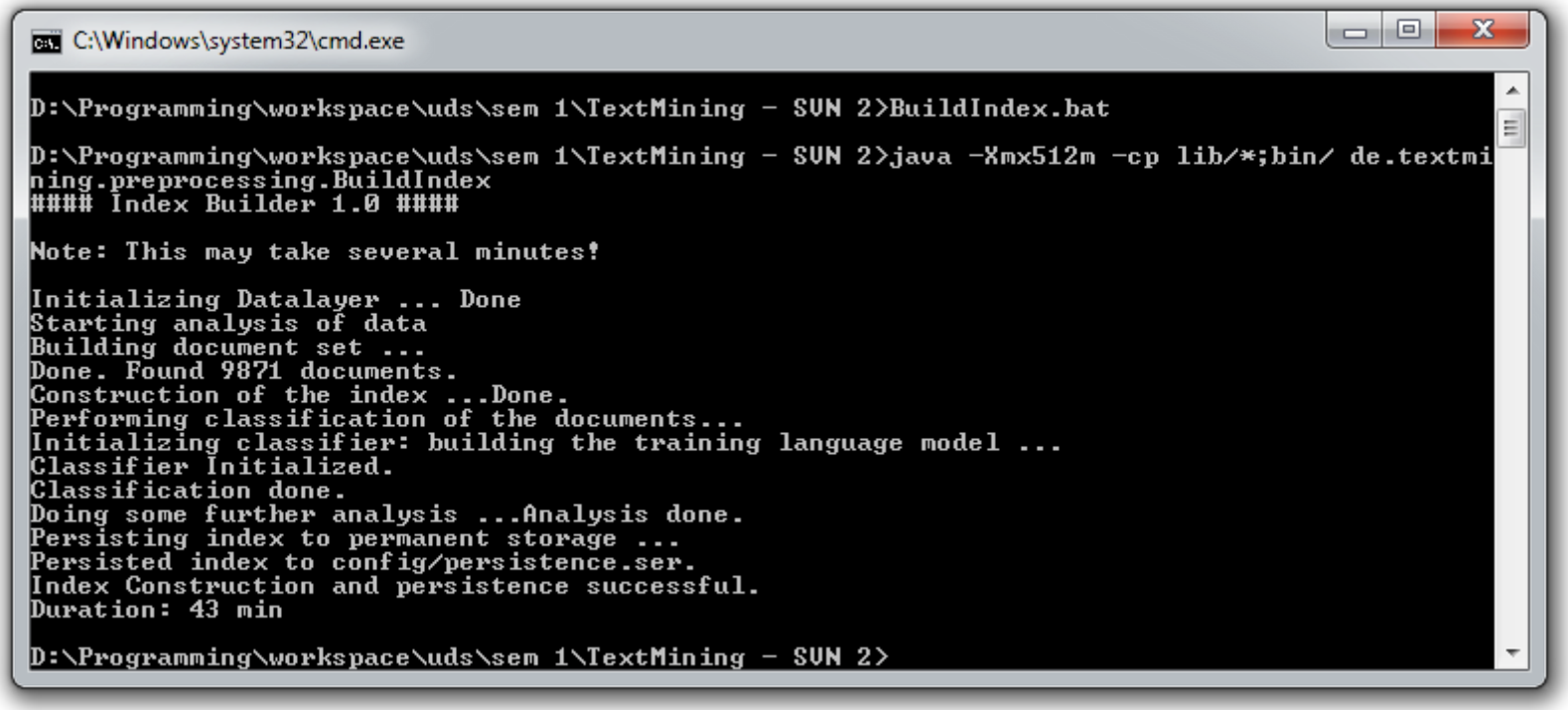
→ Very Promising results

Preprocessing – Illustration (1)

1: Änderung der Zeitkartentarife des Berufs- und Schülerverkehrs der Deutschen Bundesbahn --- [Verkehr, IP-Volk]
2: Entwurf einer Verordnung zur Änderung der Verordnung zur Durchführung des Gesetzes zur Erhebung einer Abgabe „Notopfer Berlin, BMF --- [Verkehr, Verteidigung]
3: Entwurf eines Gesetzes über den Niederlassungsbereich von Kreditinstituten, BMF --- [Wirtschaft, Innenpolitik]
4: Drittes Gesetz zur Aufhebung des Besatzungsrechts] --- [Justiz, Verkehr]
5: Handelsabkommen mit Uruguay --- [**Außenpolitik, Wirtschaft**]
6: Tarifverhandlungen im öffentlichen Dienst --- [Verkehr, IP-Volk]
7: Untersuchungen des Preisrates über die Notwendigkeit einer Erhöhung des Zuckerrübenpreises, BMF --- [Gesundheit, Landwirtschaft]
8: Erhöhung der Straßenbenutzungsgebühren in der Sowjetzone --- [IP_STAAT, Familie]
9: Bericht über die Verhandlungen in Paris --- [Verteidigung, Außenpolitik]
10: Anordnung des britischen Hohen Kommissars betreffend Vermögen, das einer Abrüstungs- oder Entmilitarisierungsmaßnahme unterliegt, BMF --- [Wirtschaft, Verkehr]
11: Reise des Bundeskanzlers nach Frankreich --- [Außenpolitik, Wirtschaft]
12: Zollsituation] --- [IP-Volk, Landwirtschaft]
13: a) Vorzeitige Rückzahlung von Tilgungsraten des deutsch-amerikanischen Nachkriegswirtschaftshilfe-Abkommens vom 27.2.1953 --- [Außenpolitik, Wirtschaft]
14: Wirtschaftspolitischer Koordinierungsausschuß, BK --- [Landwirtschaft, Gesundheit]
15: Entwurf einer Verordnung über Zolländerungen, BMF --- [Verkehr**, Gesundheit**]

15 Randomly Selected Agenda Items and their classification

Preprocessing – Illustration (2)



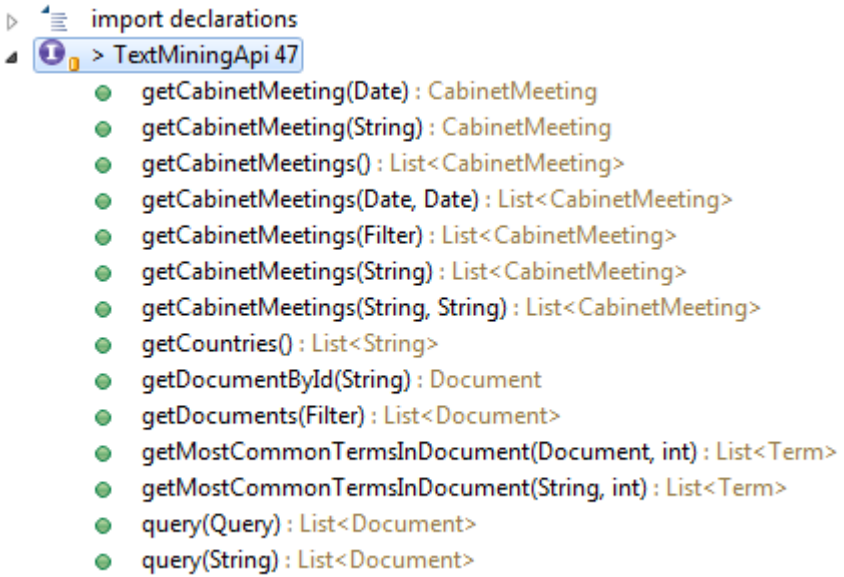
Build Index Tool: Duration 43 min – Persisted Index is 42MB

Backend / API Access

- Startup of App. Server: load persisted DB
  - Use preprocessed data and save time
- Functionality available through Java interface
  - Query Engine & Filter Engine
  - Make use of index structures
  - Various kind of queries possible

Get Agenda Items WHERE:

1: Date_in_Range(01-1951, 06-1955)
2: Topic(„Wirtschaft“)
3: Country(„Kuba“)



Example Usage

```
 1:  TextMiningApi api = (TextMiningApi) Naming.lookup("rmi://localhost:60501/backend");
 2:
 3:  System.out.println("Number of Cabinet Meetings yearwise and grouped by category.\n");
 4:  System.out.println("Total number of cabinet meetings: " + api.getCabinetMeetings().size() );
 5:
 6:  for (int i=1949; i<=1964; i++) {
 7:      String year = Integer.toString(i);
 8:      System.out.println("### YEAR " + year + " ###");
 9:      System.out.println("Number of Meetings: " + api.getCabinetMeetings(year).size() );
10:
11:   for (String cat : Config.CATEGORIES) {
12:        Filter filter = new AndFilter(new YearExactFilter(year), new CategoryFilter(
                              Utils.getCategoryFromString(cat)));
13:
14:       List<CabinetMeeting> cms = api.getCabinetMeetings(filter);
15:       System.out.println(cat + ": " + cms.size());
16:   }
17:
18:   System.out.println("\n");
```

Source Code

Part of the output:

```
Total number of cabinet meetings: 808
### YEAR 1949 ###
Number of Cabinet Meetings: 30
Außenpolitik: 26
Familie: 2
Gesundheit: 12
Innenpolitik: 14
IP-Staat: 5
IP-Volk: 16
Justiz: 24
Landwirtschaft: 18
Verkehr: 26
Verteidigung: 25
Wirtschaft: 27


### YEAR 1950 ###
Number of Cabinet Meetings: 85
Außenpolitik: 79
[…]
```

Implementation Problems

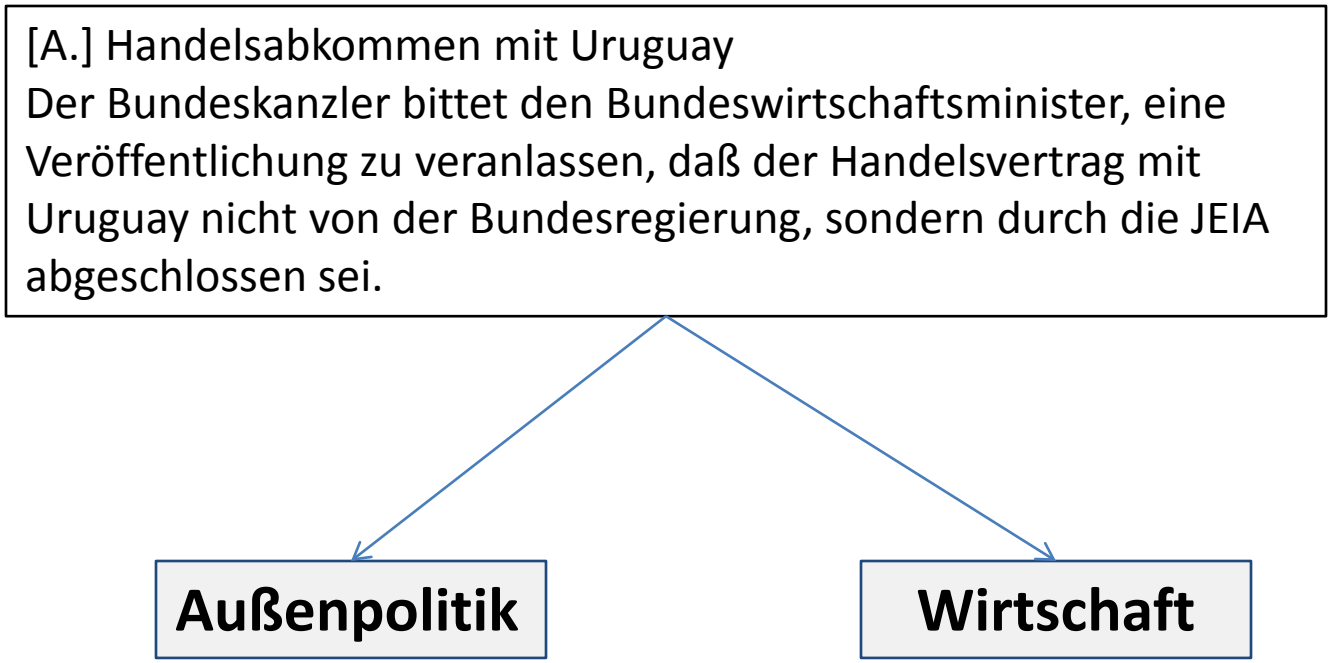
- Problems we encountered during development
  - Certain Encoding Problems (*Umlaute*, different encoding schemes in different parts)
  - Retrieval of parts of the data
    - Problems with different styles in data aggregation
  - First approach to classification did not work
    - Term Frequency analysis -> no expressive results
    - Extra Weighting of terms in the title did not help much
    - Documents were probably to short
  - Testing of data retrieval and construction was time consuming
    - Preprocessing takes long, retrieval of data took quite some time

Evaluation & Findings

- Very powerful Query and Filtering engine
  - High variation of queries are possible
  - Multidimensional Search (Topic, Time, Location)
- However:
  - Difficult to find interesting correlations
  - Problem: usually you do not know what to look for
- Some results are presented on the next few slides

Evaluation & Findings

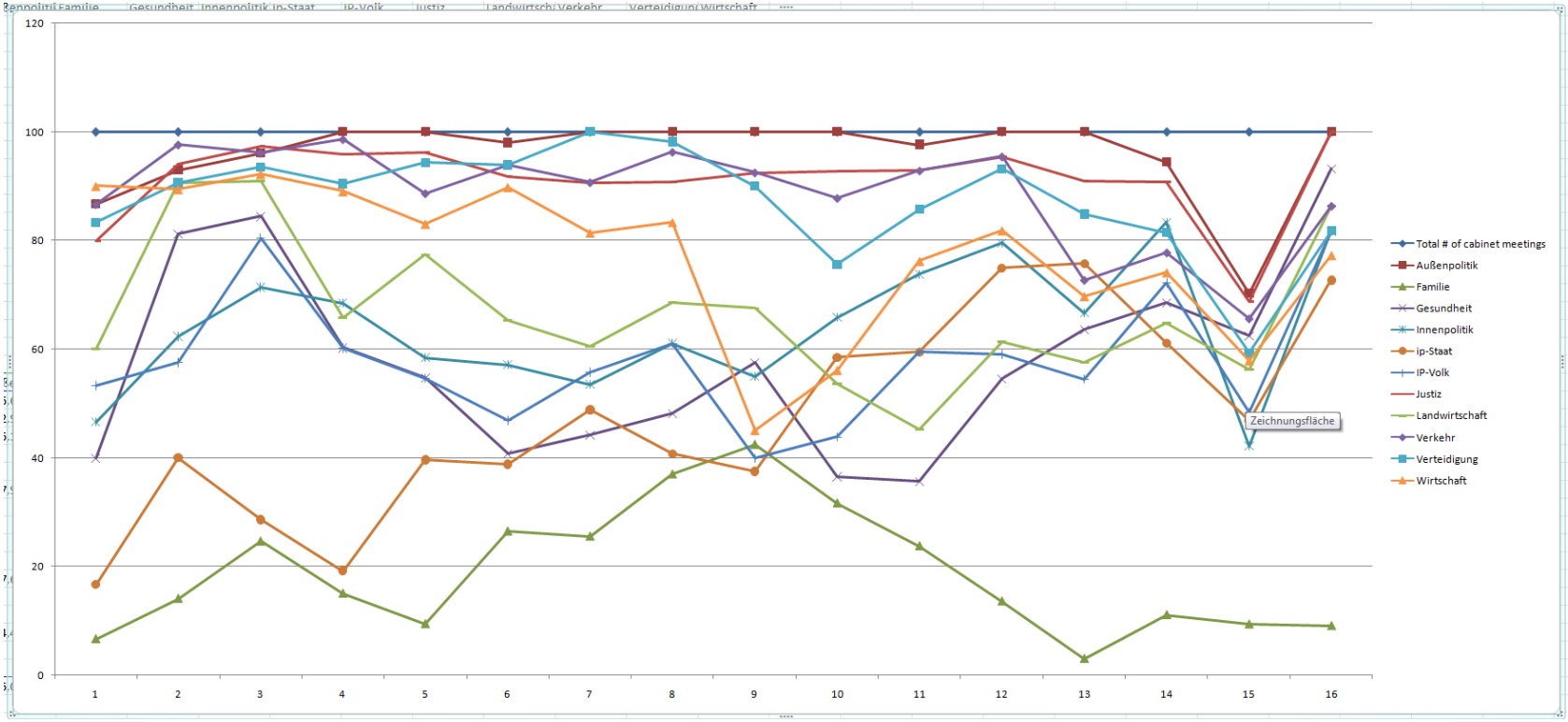
- Automatic Classification Result:



→ Good, isn‘t it?

Evaluation & Findings

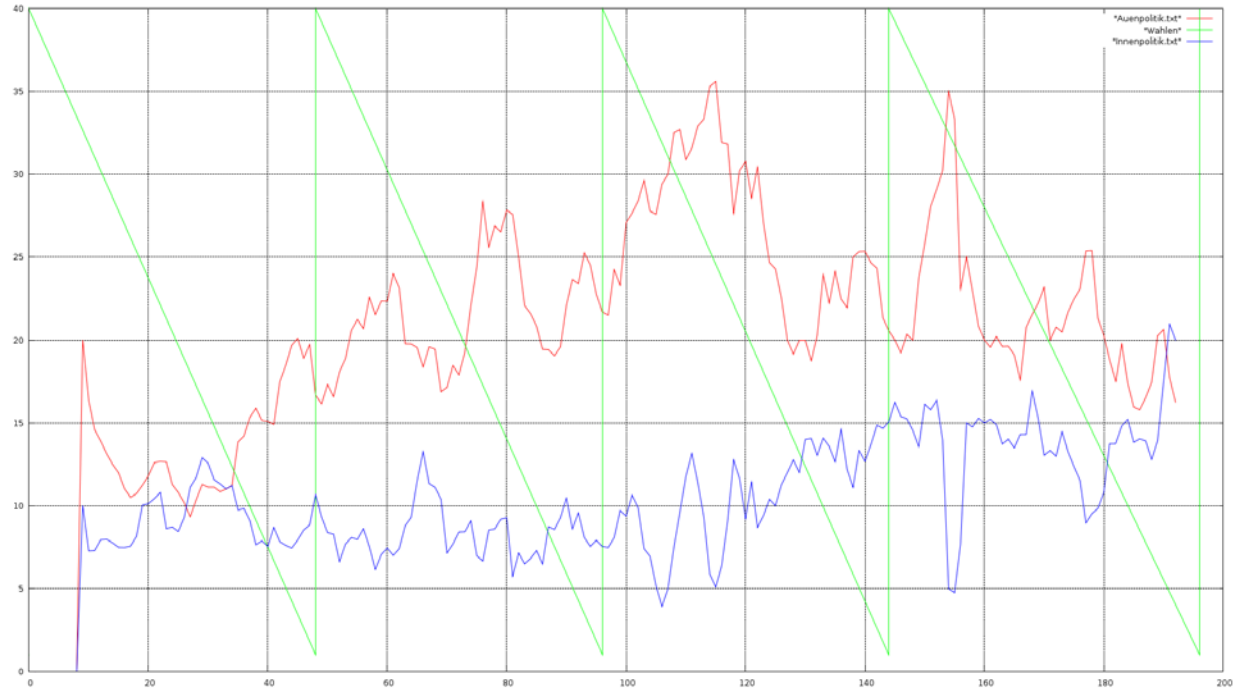
- Topics w.r.t. time:



Do you see some trend or correlation?

Evaluation & Findings

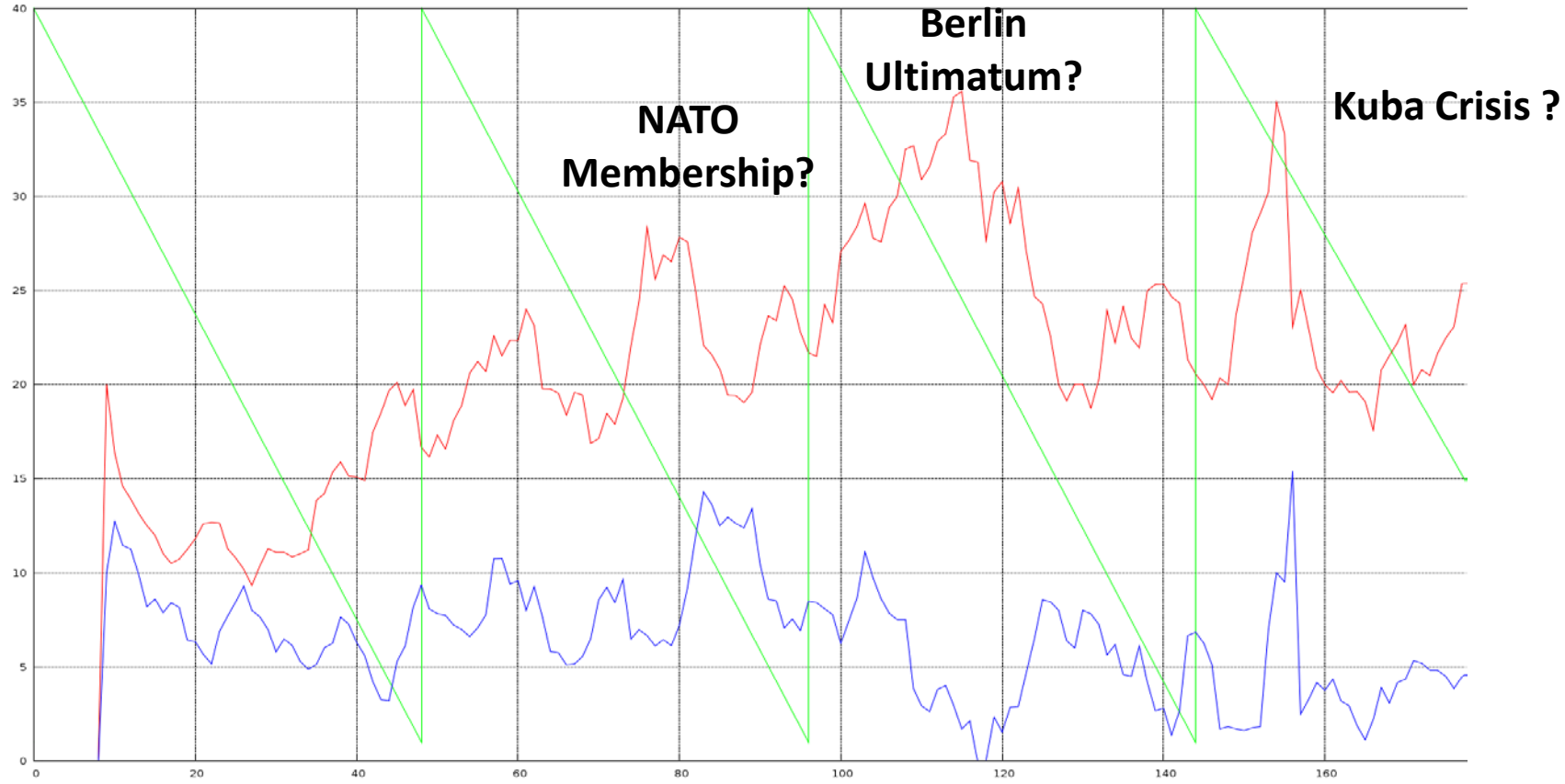
- „Außenpolitik“ (Red) vs. „Innenpolitik“ (Blue)



Slight trend: Inverse correlation

Evaluation & Findings

- „Außenpolitik“ (Red) vs. „Verteidigung“ (Blue)



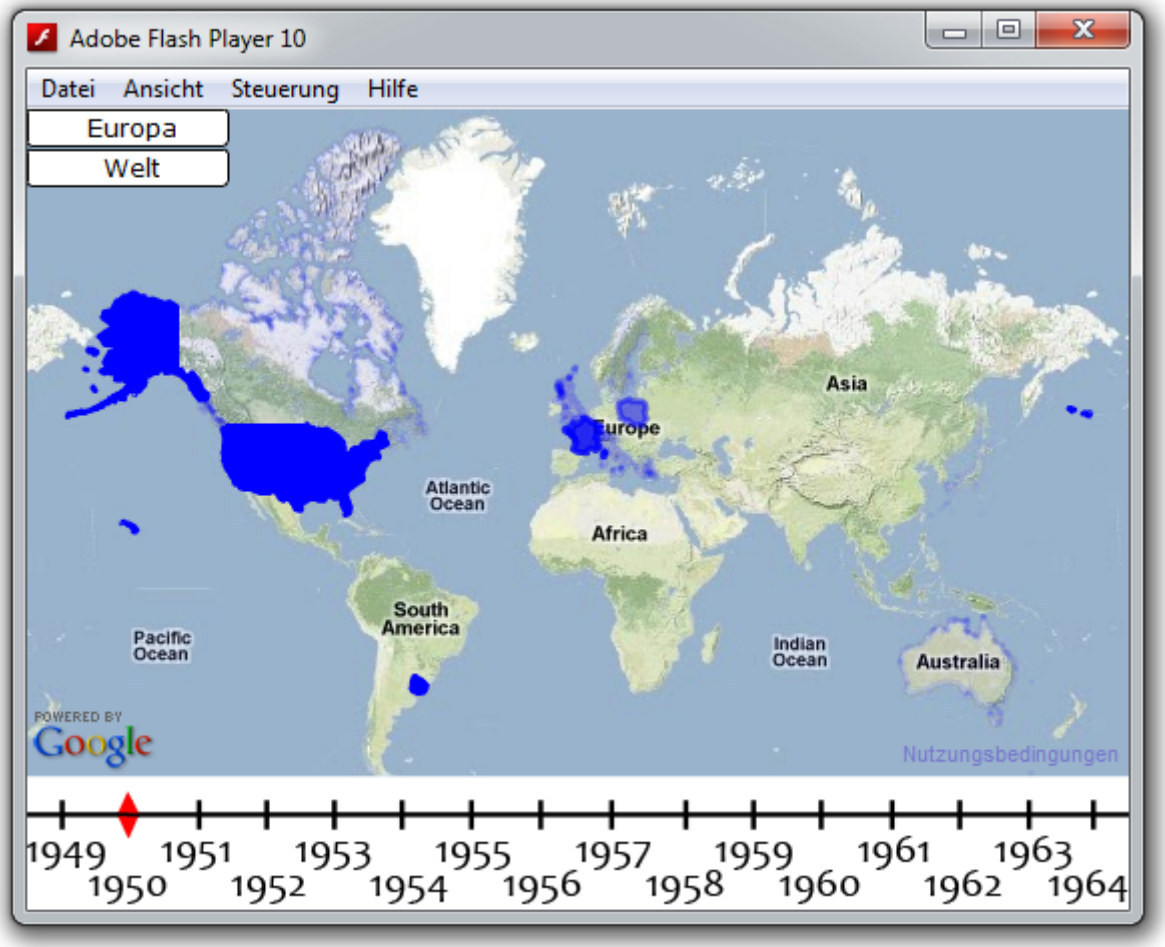
Evaluation & Findings

- Based on Dataset retrieved from API
  - Query:
    - for each year and country
      - Print out the number of occurrences in the agenda items
  - Result (Excerpt):

Occurrences per year:
YEAR 1949
USA : 2
Amerika: 11
DDR: 3
Deutschland: 19
England: 2
Frankreich: 9
Polen: 2
Schweden: 1
Schweiz: 3
Spanien: 1
Uruguay: 1

Occurrences per year:
YEAR 1950
USA : 2
Amerika: 25
Argentinien: 1
Australien: 2
Belgien: 2
Brasilien: 6
Deutschland: 70
England: 6
Frankreich: 22
Griechenland: 2
[…]

Evaluation & Findings



DEMO

Conclusion

- Really hard to find correlations
  - Need to have knowledge about domain?
- Very powerful Query and Filter interface
  - Can be used to evaluate various kinds of queries

- Gathered Experiences in Textmining and Data Retrieval
  - Not all approaches worked directly, try & error

Future Work

- Persons must be integrated into the system (DB + Query engine)
- Enhance Search for countries, integration of countries into flash application
- Simple Web interface
- Scalability? Performance?

References

- **Bundesarchiv:** *http://www.bundesarchiv.de/cocoon/barch/0000/index.html*
- **OpenNLP:** *http://opennlp.sourceforge.net/*
- **SnowBall:** *http://snowball.tartarus.org/*
- **LingPipe:** *http://alias-i.com/lingpipe/index.html*
- **GoogleMaps**: *http://code.google.com/intl/de-DE/apis/maps/documentation/flash/*
- **WorldBorders**: *http://thematicmapping.org/downloads/world_borders.php*

Thank you for your attention.

Any Questions ???