Leipzig Linguistic Services
A 4 Years Summary of Providing Linguistic Web Services

Marco Büchler, Gerhard Heyer
Leipzig University
{mbuechler,heyer}@informatik.uni-leipzig.de

Leipzig University

25. March 2008

1. Motivation
2. Previous works
3. Applications
4. Technical experiences
    - Reducing network load
    - Load balancing
    - Access restrictions / Load reduction
    - Local caching strategies
5. Unsolved problems
6. Conclusion

Motivation for Establishing Leipzig Linguistic Services

- Reducing overhead in providing Text Mining data
  - Data needs to be dumped.
  - An user interface had to be build.
- Offering an easy access to Language Resources for everyone
- History:
  - April 2004: first version online
  - September 2006: Storing server access in daily files (A 2 years corpus of log files exists containing about 43 million entries.)
  - 2007: 26.9 million overall access in the year 2007
  - October 2007: Up to 1.7 million access per day
  - January 2008: Currently 18 services installed.
  - October 2008: 20.9 million access in October 2008
  - in 2008: 36.8 million overall access in the year 2008
  - until now: Client implementations in Java, .NET, Perl, Delphi, Python and PHP are known

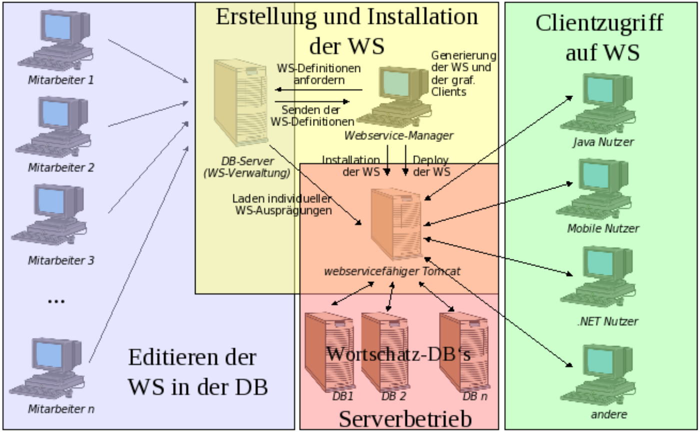
Motivation for Establishing Leipzig Linguistic Services

- Reducing overhead in providing Text Mining data
  - Data needs to be dumped.
  - An user interface had to be build.
- Offering an easy access to Language Resources for everyone
- History:
  - April 2004: first version online
  - September 2006: Storing server access in daily files (A 2 years corpus of log files exists containing about 43 million entries.)
  - 2007: 26.9 million overall access in the year 2007
  - October 2007: Up to 1.7 million access per day
  - January 2008: Currently 18 services installed.
  - October 2008: 20.9 million access in October 2008
  - in 2008: 36.8 million overall access in the year 2008
  - until now: Client implementations in Java, .NET, Perl, Delphi, Python and PHP are known

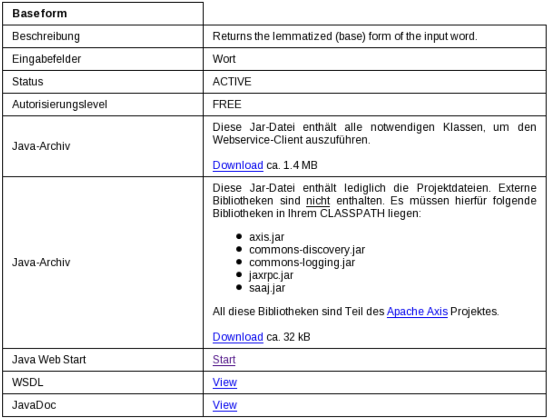
Agenda

1. Motivation
2. Previous works
3. Applications
4. Technical experiences
    - Reducing network load
    - Load balancing
    - Access restrictions / Load reduction
    - Local caching strategies
5. Unsolved problems
6. Conclusion

Leipzig Linguistic Services - Basics I (Roles)



Leipzig Linguistic Services - Basics II (Service Discovery)



Baseform	
Beschreibung	Returns the lemmatized (base) form of the input word.
Eingabefelder	Wort
Status	ACTIVE
Autorisierungslevel	FREE
Java-Archiv	Diese Jar-Datei enthält alle notwendigen Klassen, um den Webservice-Client auszuführen. Download ca. 1.4 MB
Java-Archiv	Diese Jar-Datei enthält lediglich die Projektdateien. Externe Bibliotheken sind nicht enthalten. Es müssen hierfür folgende Bibliotheken in Ihrem CLASSPATH liegen: • axis.jar • commons-discovery.jar • commons-logging.jar • jaxrpc.jar • saaj.jar All diese Bibliotheken sind Teil des Apache Axis Projektes. Download ca. 32 kB
Java Web Start	Start
WSDL	View
JavaDoc	View

Leipzig Linguistic Services - Basics III (Service Overview)

- 11 free of 18 services in LLS

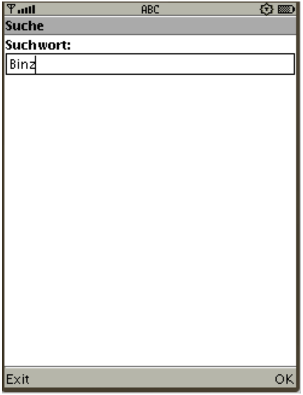
Service name	Access frequency	Percentage
Baseform	18.631, 956	43.036%
Frequencies	15, 927, 075	36.788%
Synonyms	3, 855, 662	8.906%
Sentences	1, 959, 490	4.526%
Thesaurus	1, 743, 172	4.026%
Sachgebiet	423, 788	0.979%
Wordforms	398, 532	0.921%
Cooccurrences	277, 777	0.642%
LeftNeighbours	25, 927	0.060%
Similarity	21, 725	0.050%
Kreuzwortraetsel	1, 564	0.004%

Agenda

1. Motivation
2. Previous works
3. Applications
4. Technical experiences
   - Reducing network load
   - Load balancing
   - Access restrictions / Load reduction
   - Local caching strategies
5. Unsolved problems
6. Conclusion

Mobile Access to Language Resources

Request



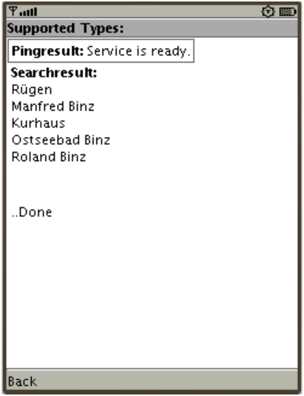
ABC

Suche

Suchwort:

Binz

Response



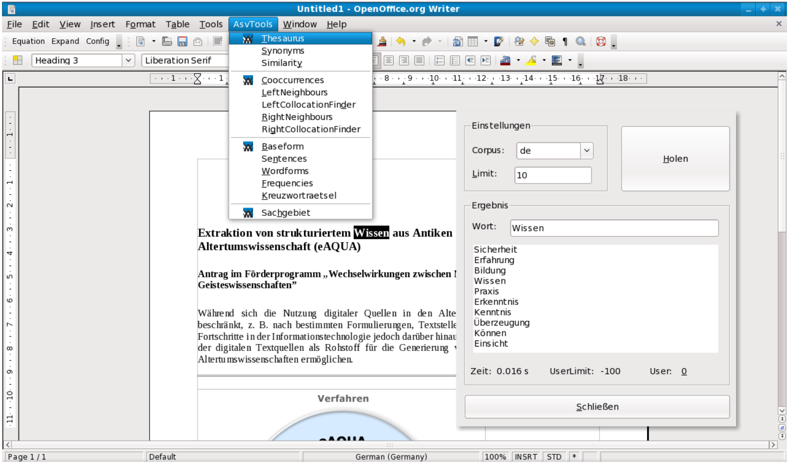
Supported Types:

Pingresult: Service is ready.

Searchresult:
Rügen
Manfred Binz
Kurhaus
Ostseebad Binz
Roland Binz

..Done

Office Integration of Language Resources



Extraktion von strukturiertem Wissen aus Antiken Altertumswissenschaft (eAQUA)

Antrag im Förderprogramm „Wechselwirkungen zwischen I Geisteswissenschaften"

Während sich die Nutzung digitaler Quellen in den Alte beschränkt, z. B. nach bestimmten Formulierungen, Textstelle Fortschritte in der Informationstechnologie jedoch darüber hinau der digitalen Textquellen als Rohstoff für die Generierung v Altertumswissenschaften ermöglichen.

Corpus: de

Limit: 10

Wort: Wissen

Sicherheit
Erfahrung
Bildung
Wissen
Praxis
Erkenntnis
Kenntnis
Überzeugung
Können
Einsicht

Zeit: 0.016 s UserLimit: -100 User: 0

Agenda

1. Motivation
2. Previous works
3. Applications
4. Technical experiences
   - Reducing network load
   - Load balancing
   - Access restrictions / Load reduction
   - Local caching strategies
5. Unsolved problems
6. Conclusion

General experiences

Number of total requests		43, 297, 467
Exceptions		233, 786
Number of total responses		43, 064, 741
Positive responses	22, 781, 529	
Negative responses	20, 283, 212	

Chances for CLARIN

- Building representative corpora (e. g. categories like *fashion*, *phonetics*, *gymnastics*, *linguistics* and *business language*)
- Sharing common data (e. g. *Baseform* or *Synonyms* services for several languages)

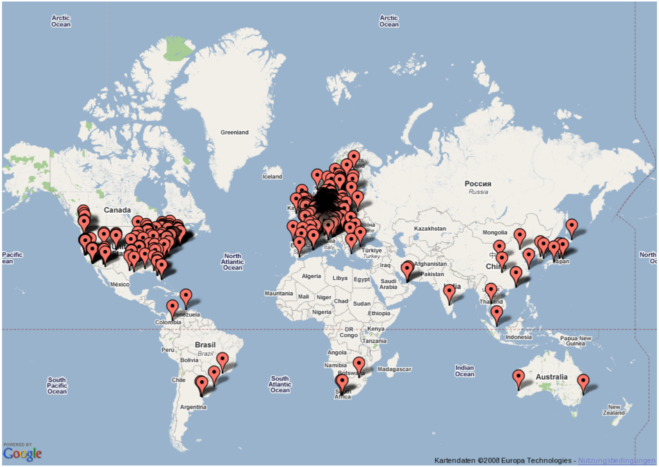
General experiences

Number of total requests		43, 297, 467
Exceptions		233, 786
Number of total responses		43, 064, 741
Positive responses	22, 781, 529	
Negative responses	20, 283, 212	

Chances for CLARIN

- Building representative corpora (e. g. categories like *fashion*, *phonetics*, *gymnastics*, *linguistics* and *business language*)
- Sharing common data (e. g. *Baseform* or *Synonyms* services for several languages)

Network load optimization - An example



Kartendaten ©2008 Europa Technologies - Nutzungsbedingungen

Network load optimization - Service Chaining

- Detecting possible composite services

Rank	Found service chains	Frequencies
1	Baseform Frequencies	3, 210, 956
2	Baseform Synonyms Sentences	1, 259, 308
4	Synonyms Sentences	143, 744
5	Baseform Synonyms	48, 171
6	Baseform Frequencies Synonyms	46, 336
7	Baseform Thesaurus	32, 488
12	Baseform Frequencies Sachgebiet	11, 629
13	Baseform Sachgebiet	11, 604
14	Frequencies Baseform Frequencies	10, 929
15	Thesaurus Similarity	9, 746

- Common data type as parameter and return value
- UDDI Mining (e. g. abstraction and composition of service descriptions, machine translation)

Network load optimization - Service Chaining

- Detecting possible composite services

Rank	Found service chains	Frequencies
1	Baseform Frequencies	3, 210, 956
2	Baseform Synonyms Sentences	1, 259, 308
4	Synonyms Sentences	143, 744
5	Baseform Synonyms	48, 171
6	Baseform Frequencies Synonyms	46, 336
7	Baseform Thesaurus	32, 488
12	Baseform Frequencies Sachgebiet	11, 629
13	Baseform Sachgebiet	11, 604
14	Frequencies Baseform Frequencies	10, 929
15	Thesaurus Similarity	9, 746

- Common data type as parameter and return value
- UDDI Mining (e. g. abstraction and composition of service descriptions, machine translation)

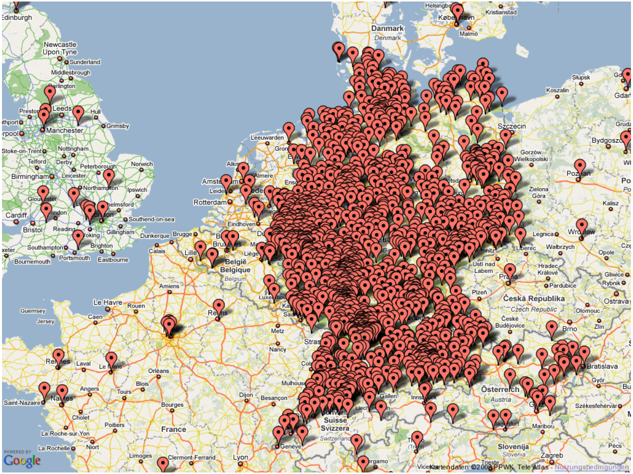
Network load optimization - Service Chaining

- Detecting possible composite services

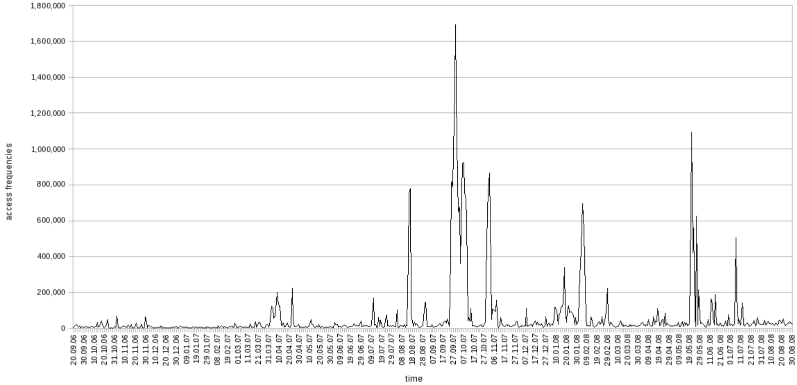
Rank	Found service chains	Frequencies
1	Baseform Frequencies	3, 210, 956
2	Baseform Synonyms Sentences	1, 259, 308
4	Synonyms Sentences	143, 744
5	Baseform Synonyms	48, 171
6	Baseform Frequencies Synonyms	46, 336
7	Baseform Thesaurus	32, 488
12	Baseform Frequencies Sachgebiet	11, 629
13	Baseform Sachgebiet	11, 604
14	Frequencies Baseform Frequencies	10, 929
15	Thesaurus Similarity	9, 746

- Common data type as parameter and return value
- UDDI Mining (e. g. abstraction and composition of service descriptions, machine translation)

DSPIN - Load balancing



Access restrictions / load reduction - An example



Access restrictions / load reduction - Requirements

- **User level brake**: Based on different authorization levels requests are delayed
- **IP address brake**: Reduce load if multi threaded requests will be sent to the server
- **Number of meaningful requests**: Filtering of senseless requests e. g. if 80% of all requests from an IP can't be answered
- **Regeneration**: Reducing load by delaying all requests as system regeneration

Local caching strategies I

Assumption

- Words are in corpus' word frequency order (Zipfian law).
- Y-axes will be replaced by the access frequency of a word.
- What is the distribution?

*Working hypothesis 1: TF*IDF similar access frequencies*

- The plot equals a term weighting plot like TF*IDF.
- Stop words will be removed to speed up the application by reducing the number of requests (Top 100 frequent words have a text coverage of about 50%).

Working hypothesis 2: Zipfian law similar access frequencies

- The plot equals a Zipfian law distribution.
- Stop words will be requested most frequently.
- pragmatic vs. ignorant users

Local caching strategies I

Assumption

- Words are in corpus' word frequency order (Zipfian law).
- Y-axes will be replaced by the access frequency of a word.
- What is the distribution?

*Working hypothesis 1: TF*IDF similar access frequencies*

- The plot equals a term weighting plot like TF*IDF.
- Stop words will be removed to speed up the application by reducing the number of requests (Top 100 frequent words have a text coverage of about 50%).

Working hypothesis 2: Zipfian law similar access frequencies

- The plot equals a Zipfian law distribution.
- Stop words will be requested most frequently.
- pragmatic vs. ignorant users

Local caching strategies I

Assumption

- Words are in corpus' word frequency order (Zipfian law).
- Y-axes will be replaced by the access frequency of a word.
- What is the distribution?

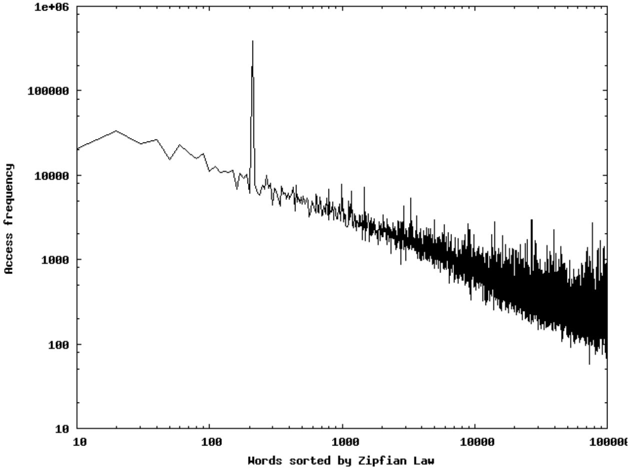
*Working hypothesis 1: TF*IDF similar access frequencies*

- The plot equals a term weighting plot like TF*IDF.
- Stop words will be removed to speed up the application by reducing the number of requests (Top 100 frequent words have a text coverage of about 50%).

Working hypothesis 2: Zipfian law similar access frequencies

- The plot equals a Zipfian law distribution.
- Stop words will be requested most frequently.
- pragmatic vs. ignorant users

Local caching strategies II



User’s pragmatism

Rank	Found service chains	Frequencies
3	Baseform Synonyms Sentences Baseform Synonyms Sentences	143, 744
8	Baseform Baseform	27, 693
10	Frequencies Frequencies	22, 344
11	Thesaurus Thesaurus	12, 619

- Due to high load artefacts of retries if the server sends timeouts
- User’s pragmatism on text samples like *had had*

Agenda

1. Motivation
2. Previous works
3. Applications
4. Technical experiences
    - Reducing network load
    - Load balancing
    - Access restrictions / Load reduction
    - Local caching strategies
5. Unsolved problems
6. Conclusion

Unsolved problems

- SOAP vs. REST:
  - SOAP Pro: Standardized protocol
  - SOAP Con: A lot of overhead in supporting user's client connection (authentication, complex data types)
  - REST: Rest usage for simple database lookups like LLS (easier sharing of Language Resources)
- Synchronous vs. asynchronous communication support
  - SOAP and REST are application protocols and require a network protocol (typically HTTP)
  - Problem: Algorithm Services typically need more time than an HTTP timeout.
  - First trials in switching to an asynchronous communication have failed in January 2007.
- Scalability of a algorithm's memory usage
  - Different implementations of algorithms are on a proof of concept level.
  - Most implementations don't scale and don't allow multiple instance at the same time.
- Incompatibility of data types
  - Integer data types: unsigned able vs. unsigned disabled number representation causes overflow
  - Floating point data types: 0.00314 vs. 3.14E-3 vs. 3.14e-3
  - date and time

Unsolved problems

- SOAP vs. REST:
  - SOAP Pro: Standardized protocol
  - SOAP Con: A lot of overhead in supporting user's client connection (authentication, complex data types)
  - REST: Rest usage for simple database lookups like LLS (easier sharing of Language Resources)
- Synchronous vs. asynchronous communication support
  - SOAP and REST are application protocols and require a network protocol (typically HTTP)
  - Problem: Algorithm Services typically need more time than an HTTP timeout.
  - First trials in switching to an asynchronous communication have failed in January 2007.
- Scalability of a algorithm's memory usage
  - Different implementations of algorithms are on a proof of concept level.
  - Most implementations don't scale and don't allow multiple instance at the same time.
- Incompatibility of data types
  - Integer data types: unsigned able vs. unsigned disabled number representation causes overflow
  - Floating point data types: 0.00314 vs. 3.14E-3 vs. 3.14e-3
  - date and time

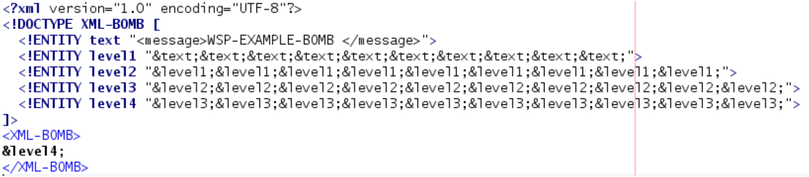
Unsolved problems

- SOAP vs. REST:
  - SOAP Pro: Standardized protocol
  - SOAP Con: A lot of overhead in supporting user's client connection (authentication, complex data types)
  - REST: Rest usage for simple database lookups like LLS (easier sharing of Language Resources)
- Synchronous vs. asynchronous communication support
  - SOAP and REST are application protocols and require a network protocol (typically HTTP)
  - Problem: Algorithm Services typically need more time than an HTTP timeout.
  - First trials in switching to an asynchronous communication have failed in January 2007.
- Scalability of a algorithm's memory usage
  - Different implementations of algorithms are on a proof of concept level.
  - Most implementations don't scale and don't allow multiple instance at the same time.
- Incompatibility of data types
  - Integer data types: unsigned able vs. unsigned disabled number representation causes overflow
  - Floating point data types: 0.00314 vs. 3.14E-3 vs. 3.14e-3
  - date and time

Unsolved problems

- SOAP vs. REST:
  - SOAP Pro: Standardized protocol
  - SOAP Con: A lot of overhead in supporting user's client connection (authentication, complex data types)
  - REST: Rest usage for simple database lookups like LLS (easier sharing of Language Resources)
- Synchronous vs. asynchronous communication support
  - SOAP and REST are application protocols and require a network protocol (typically HTTP)
  - Problem: Algorithm Services typically need more time than an HTTP timeout.
  - First trials in switching to an asynchronous communication have failed in January 2007.
- Scalability of a algorithm's memory usage
  - Different implementations of algorithms are on a proof of concept level.
  - Most implementations don't scale and don't allow multiple instance at the same time.
- Incompatibility of data types
  - Integer data types: unsigned able vs. unsigned disabled number representation causes overflow
  - Floating point data types: 0.00314 vs. 3.14E-3 vs. 3.14e-3
  - date and time

Security problems: An inoffensive XML bomb



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE XML-BOMB [
  <!ENTITY text "<message>WSP-EXAMPLE-BOMB </message>">
  <!ENTITY level1 "&text;&text;&text;&text;&text;&text;&text;&text;&text;&text;">
  <!ENTITY level2 "&level1;&level1;&level1;&level1;&level1;&level1;&level1;&level1;&level1;">
  <!ENTITY level3 "&level2;&level2;&level2;&level2;&level2;&level2;&level2;&level2;&level2;&level2;">
  <!ENTITY level4 "&level3;&level3;&level3;&level3;&level3;&level3;&level3;&level3;&level3;&level3;">
]>
<XML-BOMB>
&level4;
</XML-BOMB>
```

- Protection against attacks
- Handling of large messages (e. g. a complete word list of million words will be sent)

Agenda

1. Motivation
2. Previous works
3. Applications
4. Technical experiences
    - Reducing network load
    - Load balancing
    - Access restrictions / Load reduction
    - Local caching strategies
5. Unsolved problems
6. Conclusion

Conclusion

- Technical experiences
  - Reducing network load
  - Load balancing
  - Access restrictions / Load reduction
  - Local caching strategies
- Unsolved problems
  - Synchronous vs. asynchronous communication support
  - Easier access to language resources (customer view)
  - Scalability
  - Incompatibility of data types
- Applications
  - Mobile applications
  - Office integration

Conclusion

- Technical experiences
  - Reducing network load
  - Load balancing
  - Access restrictions / Load reduction
  - Local caching strategies
- Unsolved problems
  - Synchronous vs. asynchronous communication support
  - Easier access to language resources (customer view)
  - Scalability
  - Incompatibility of data types
- Applications
  - Mobile applications
  - Office integration

Conclusion

- Technical experiences
  - Reducing network load
  - Load balancing
  - Access restrictions / Load reduction
  - Local caching strategies
- Unsolved problems
  - Synchronous vs. asynchronous communication support
  - Easier access to language resources (customer view)
  - Scalability
  - Incompatibility of data types
- Applications
  - Mobile applications
  - Office integration

Conference on Text Mining Services – TMS, Leipzig 2009

- General Chair
  - Gerhard Heyer (Computer Science, University of Leipzig),
  - Charlotte Schubert (Historical Sciences, University of Leipzig),
  - Peter Wittenburg (Computer Science, MPI Nimwegen),
  - Manfred Kirchgeorg (Marketing, Leipzig Graduate School of Manag.)
- Dates
  - **Workshop**: 24-25.3.2009
  - **Student Day**: 23.3.2009
  - **Submission of workshop papers**: 20.11.08
  - **Publication ready version**: 2.2.2009
- Topics
  - Design and engineering of text mining services
  - Basic text mining services technologies and architectures
  - Use of text mining services in the E-Humanities
  - Industrialization and standardization of text mining services