Retrieval in Text Collections with Historic Spelling Using Linguistic and Spelling Variants

Andrea Ernst-Gerlach and Norbert Fuhr

University of Duisburg-Essen, Germany
Contents

1 Motivation
2 Architecture
3 Evaluation
4 Summary & Future work
Motivation

Samson. "Gregory, on my word, we’ll not carry coals"
Greg. "No. For then we should be colliers"
Sam "I strike quickly, being mov’d"
...

- Long absence of spelling standards

⇒ Problem searching for historic documents
Architecture

User Interface

Generating Word Forms

Generating Historic Word Forms

Search Engine

DL

Layer 4

Layer 3

Layer 2

Layer 1

Layer 0
Search example

Motivation
Andrea Ernst-Gerlach
Retrieval in Text Collections with Historic Spelling Using Linguistic and Spelling Variants

Architecture

Evaluation

Summary & Future work
User Interface

![Historic Search Interface](image)

- **Search terms:** Tür
- **Year:** from 1800 to 1850
- **Location:** Germany
- **Term precision as %:** 0.8

Andrea Ernst-Gerlach

Retrieval in Text Collections with Historic Spelling Using Linguistic and Spelling Variants

UNIDUE
Generating Word Forms

- First mapping:
  Search term $\rightarrow$ contemporary inflections

- Problem: German is highly inflected
e.g. haben $\rightarrow$ hab, hat, hätten, ...
$\rightarrow$ Cannot use standard stemming approach

- Use dictionary (German Vocabulary Database)
Generating Historic Word Forms

- Second mapping: Contemporary inflections $\rightarrow$ spelling variants
- Apply rules on every generated word form
- Variants get weights reflecting the precision
- Rules can be applied until minimum precision is not reached
### Example rules for English and German

<table>
<thead>
<tr>
<th>Contemp. spelling</th>
<th>19th century</th>
<th>rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>always</td>
<td>alwaies</td>
<td>y → ie</td>
</tr>
<tr>
<td>sudden</td>
<td>suddain</td>
<td>e → ai</td>
</tr>
<tr>
<td>publicly</td>
<td>publikely</td>
<td>c → ke</td>
</tr>
<tr>
<td>wiedergaben</td>
<td>wiedergaben</td>
<td>wieder → wider</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ie → i</td>
</tr>
<tr>
<td>akzeptieren</td>
<td>acceptieren</td>
<td>kz → cc</td>
</tr>
<tr>
<td></td>
<td></td>
<td>k → c ∧ z → c</td>
</tr>
<tr>
<td>seht</td>
<td>sehet</td>
<td>t → et</td>
</tr>
</tbody>
</table>
Generation of transformation rules

1. Training set of triplets
   - Contemporary word form
   - Historic variant
   - Collection frequency
Generation of transformation rules

1. Training set of triplets
   - Contemporary word form
   - Historic variant
   - Collection frequency

2. Generate rule core
   - e.g. unnütz - unnuts
     rule cores: \( \hat{\text{unn(ü,u)t}} \text{ t(z,s)} \)
Generation of transformation rules

1. Training set of triplets
   - Contemporary word form
   - Historic variant
   - Collection frequency

2. Generate rule core
   - e.g. unnütz - unnuts
     rule cores: ˆunn(ü,u)t t(z,s)§

3. Generate rule candidates
   - e.g. ü → u  nü → nu  üt → ut  nüt → nut
Generation of transformation rules

1. Training set of triplets
   - Contemporary word form
   - Historic variant
   - Collection frequency

2. Generate rule core
   - e.g. unnütz - unnuts
     rule cores: \( \hat{\text{unn}}(\ddot{u},u)t \) \( t(z,s) $

3. Generate rule candidates
   - e.g. \( \ddot{u} \rightarrow u \) \( \ddot{n}u \rightarrow nu \) \( \ddot{u}t \rightarrow ut \) \( \ddot{n}u t \rightarrow nut \)

4. Pruning of rule candidates
   - Based on precision and frequency of candidates
Search engine

- Merge all variants
- Build query expansion
- Probabilistic weighting of search terms
- Using the probabilistic search engine PIRE
- Can also be combined with other search engines.
Evaluation

- Evaluate combination of dictionary and rule application
- Baseline Stemming and Levenshtein distance
- 2 different evaluations:
  - Words from the whole collection
  - Restricted to words in historic forms
- One term queries
- Words the same meaning relevant.
Evaluation for words from the whole collection

![Graph showing precision and recall for different methods.]

- **Stemming**
- **Vocabulary database**
- **Stemming + Levenshtein 1**
- **Stemming + Levenshtein 2**
- **Stemming + Rule application**
- **Vocabulary database + Levenshtein 1**
- **Vocabulary database + Levenshtein 2**
- **Vocabulary database + Rule application**
- **Conventional search engine**
Evaluation restricted to historic forms

![Graph](image)

- Stemming
- Vocabulary database
- Stemming + Levenshtein 1
- Stemming + Levenshtein 2
- Stemming + Rule application
- Vocabulary database + Levenshtein 1
- Vocabulary database + Levenshtein 2
- Vocabulary database + Rule application
Summary & Future work

I presented

- Architecture for a historic search engine
- Developed approach outer performs traditional approach
- Levenshtein does not work
- We should also use stemming

Future work:

- Enhancement of the rule generation
- User driven generation of rule sets.