A Three-step Model of Language Detection in Multilingual Ancient Texts

Maria Sukhareva and Zahurul Islam

Text Technology Group, Goethe-Universität Frankfurt am Main

Monday, 10 October 2011

Outline

Example: Modern Sentences

1. Wenn der Driver beim Link zum Host trappt, muss er mal geupdated werden.
2. PHP peut également générer d'autres formats en rapport avec le Web, comme le WML, le SVG, le format PDF, ou encore des images bitmap telles que JPEG, GIF ou PNG.
3. French cuisine was codified in the 20th century by Escoffier to become the modern version of haute cuisine; Gastro-tourism and the Guide Michelin helped to acquaint people with the rich bourgeois and peasant cuisine of the French countryside starting in the 20th century.

courtesy: Armin Hoenen

Example: Modern Sentences

1. Wenn der Driver beim Link zum Host trappt, muss er mal geupdated werden.
2. PHP peut également générer d'autres formats en rapport avec le Web, comme le WML, le SVG, le format PDF, ou encore des images bitmap telles que JPEG, GIF ou PNG.
3. French cuisine was codified in the 20th century by Escoffier to become the modern version of haute cuisine; Gastro-tourism and the Guide Michelin helped to acquaint people with the rich bourgeois and peasant cuisine of the French countryside starting in the 20th century.

courtesy: Armin Hoenen

Multilingualism in Modern Texts

- The number of multilingual resources available on the web are rising drastically
- Imposing new challenges to NLP researchers
- It is also a challenge for many NLP applications
- There are many language detection toolkits available for modern languages

Example: Ancient Sentences

1. Er uuas miteuuare, also Esaias chat Gaude et letare, Hierusalem, quia rex tuus uenit tibi mansuetus.
2. et in anniuersario sancte thiedhilda to then neppenon ande to then almoson ande to themo inganga thero iungereno tue malt
3. Tiû grûba uólliu uuazzeres bézeichenet, dáz ér chât Saluum me fac, deus

courtesy: Timothy Price

Example: Ancient Sentences

1. Er uuas miteuuare, also Esaias chat Gaude et letare, Hierusalem, quia rex tuus uenit tibi mansuetus.
2. et in anniuersario sancte thiedhilda to then neppenon ande to then almoson ande to themo inganga thero iungereno tue malt
3. Tiû grûba uólliu uuazzeres bézeichenet, dáz ér chât Saluum me fac, deus

courtesy: Timothy Price

Multilingualism in Ancient Texts

Sources

- Loan words
- Comments in foreign languages

Patrologia Latina (Mehler et al. 2011)

- 700,000 foreign words (Sukhareva et al. 2011)
- Comments in French, English, German, etc.

Old High German (OHG)

- 9% of words are Latin (Sukhareva et al. 2011)

Multilingualism in Ancient Texts

Sources

- Loan words
- Comments in foreign languages

Patrologia Latina (Mehler et al. 2011)

- 700,000 foreign words (Sukhareva et al. 2011)
- Comments in French, English, German, etc.

Old High German (OHG)

- 9% of words are Latin (Sukhareva et al. 2011)

Multilingualism in Ancient Texts

Sources

- Loan words
- Comments in foreign languages

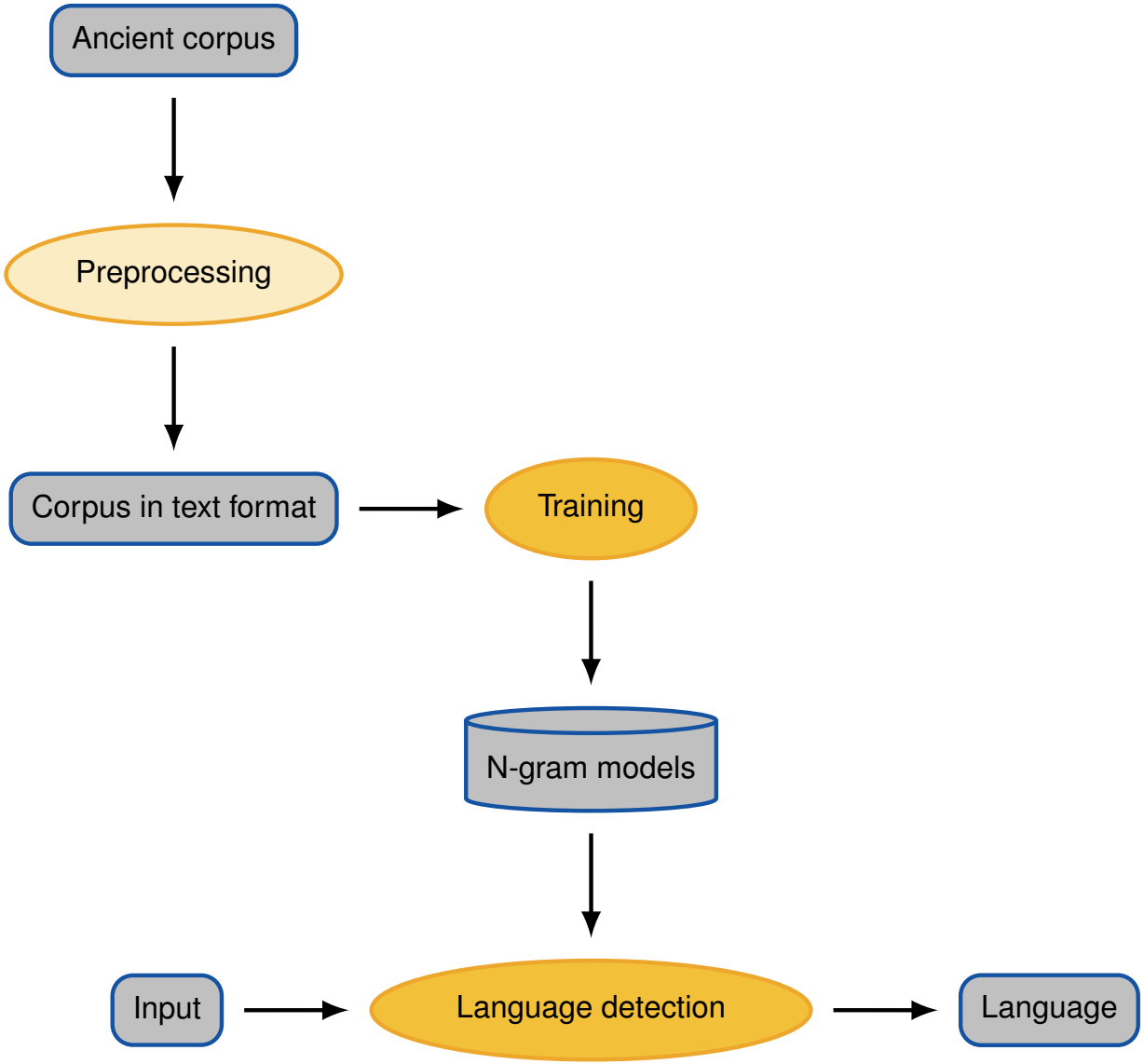
Patrologia Latina (Mehler et al. 2011)

- 700,000 foreign words (Sukhareva et al. 2011)
- Comments in French, English, German, etc.

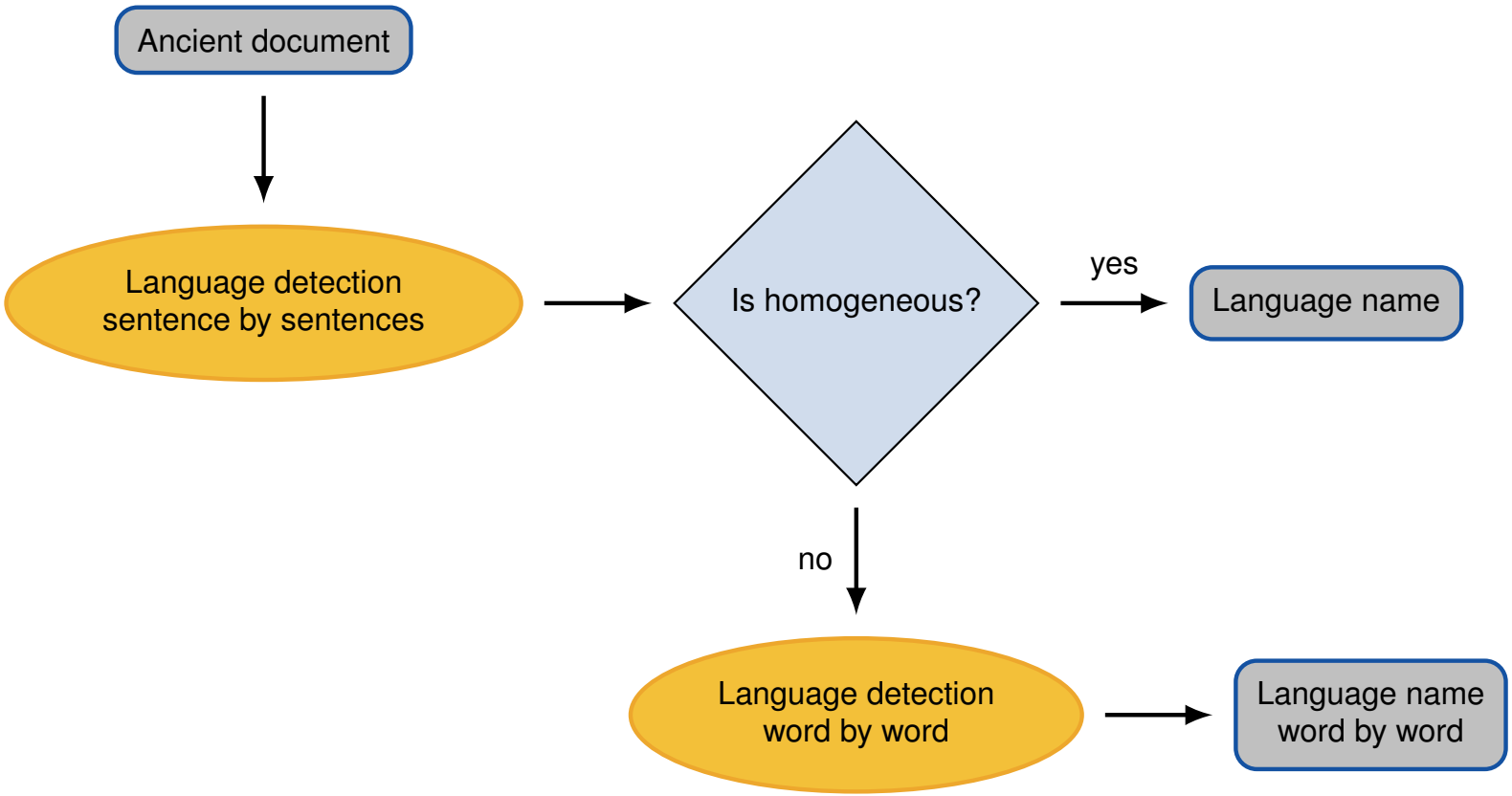
Old High German (OHG)

- 9% of words are Latin (Sukhareva et al. 2011)

Language Detection (LD) toolkit (Waltinger and Mehler 2009)



LD toolkit (Islam et al. 2011)



Evaluation: Test set

Language	Tokens	Sentences	Unknown
German–French	5893	315	460
English–Turkish	14022	724	438
OHG–Latin	1397	217	499

- English–Turkish test corpus is comprised of English Wikipedia articles (e.g. Atatürk, Istanbul etc.), which contain numerous Turkish words.
- German–French test corpus is comprised of German Wikipedia articles, which contain numerous French words.
- OHG–Latin corpus is comprised of OHG sentences, which contain Latin words.

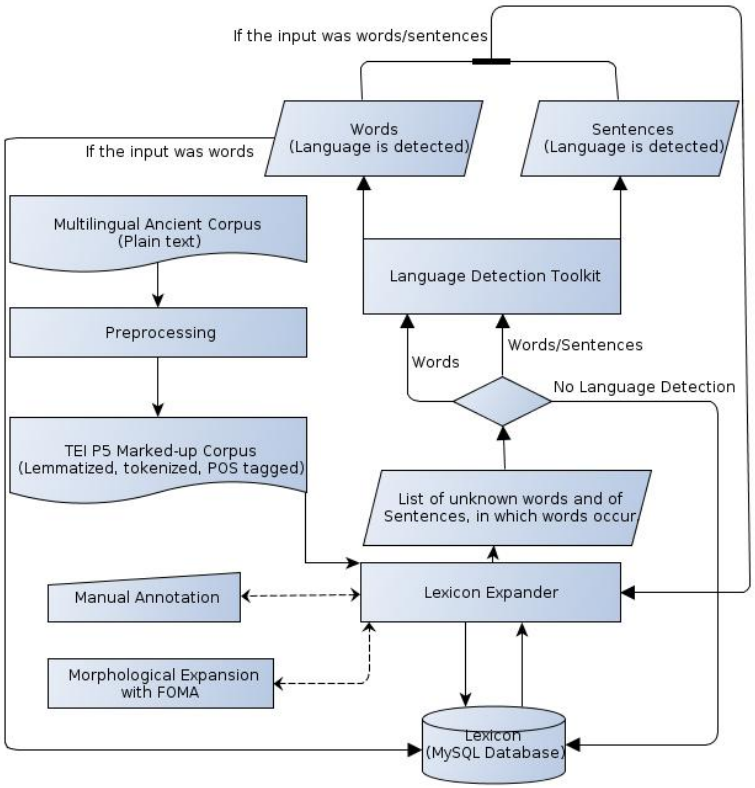
LD toolkit: Evaluation

Language	**F-score**	**Accuracy**
German–French	0.40	35.43%
English–Turkish	0.36	38.13%
OHG–Latin	0.79	70.34%

Lexicon Expander

1. An application module for the eHumanities Desktop
2. Used to build and annotate lexica
3. The LD Toolkit is integrated into it

System Architecture



1. A multilingual text is preprocessed
2. The Lexicon Expander extracts unknown words
3. One of three options of language detection is applied
4. The results are saved in a MySQL DB
5. The user can manually annotate the lexicon or apply morphological expansion

Lexicon Expander

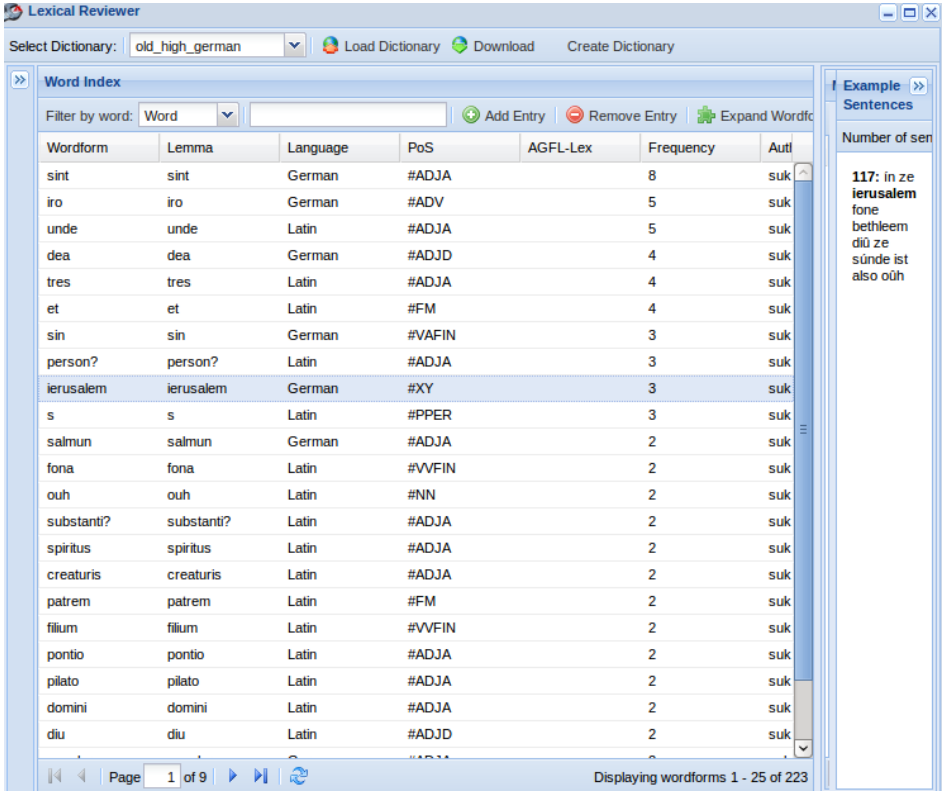


Figure: The GUI of the Lexicon Expander

- The F-score and accuracy are low if the LD Toolkit input is single words
- The Lexicon Expander post-processes the LD Toolkit output, improving the f-score and accuracy
- The Lexicon Expander relies on the language sentences, in which the target word occurs and on the co-occurring unknown words

Lexicon Expander

- $S(w) = \{s \in S \mid w \in f(s)\}$

1. $L: W \rightarrow \{l_1, \ldots, l_m\} = \mathbb{L}$
2. $L_1(w) = \arg\max_{l \in \mathbb{L}} \{|\{w' \in W' \mid \exists s \in S(w): w' \in f(s) \wedge L_1(w') = l\}|\}$
3. $L_2(w) = \arg\max_{l \in \mathbb{L}} \{|\{s \in S(w) \mid L_1(s) = l\}|\}$

Data: The set of unknown words $W' = \{w_1, \ldots, w_n\}$
Result: The language $\mathcal{L}(w)$ of any word $w_i \in W'$
for $i = 1..n$ **do**
  $\mathrm{L_s}(w_i) \leftarrow \{l \in \mathrm{L} \mid \exists s \in S(w_i) : l = L_1(s)\}$;
  if $|\mathrm{L_s}(w_i)| = 1$ **then**
    $\mathcal{L}(w) \leftarrow L_1(w_i)$;
  end
  else
    $\mathcal{L}(w) \leftarrow L_2(w_i)$;
  end
end

Figure: Lexicon Expander Language Assignment Algorithm

Lexicon Expander

- $S(w) = \{s \in S \mid w \in f(s)\}$

1. $L\colon W \rightarrow \{l_1, \ldots, l_m\} = \mathbb{L}$
2. $L_1(w) = \arg\max_{l \in \mathbb{L}} \{|\{w' \in W' \mid \exists s \in S(w)\colon w' \in f(s) \wedge L_1(w') = l\}|\}$
3. $L_2(w) = \arg\max_{l \in \mathbb{L}} \{|\{s \in S(w) \mid L_1(s) = l\}|\}$

Data: The set of unknown words $W' = \{w_1, \ldots, w_n\}$
Result: The language $\mathcal{L}(w)$ of any word $w_i \in W'$
for $i = 1..n$ **do**
  $\mathrm{L_s}(w_i) \leftarrow \{l \in \mathrm{L} \mid \exists s \in S(w_i) : l = L_1(s)\}$;
  if $|\mathrm{L_s}(w_i)| = 1$ **then**
    $\mathcal{L}(w) \leftarrow L_1(w_i)$;
  end
  else
    $\mathcal{L}(w) \leftarrow L_2(w_i)$;
  end
end

Figure: Lexicon Expander Language Assignment Algorithm

Lexicon Expander

- $S(w) = \{s \in S \mid w \in f(s)\}$

1. $L\colon W \rightarrow \{l_1, \ldots, l_m\} = \mathbb{L}$
2. $L_1(w) = \arg\max_{l \in \mathbb{L}} \{|\{w' \in W' \mid \exists s \in S(w)\colon w' \in f(s) \wedge L_1(w') = l\}|\}$
3. $L_2(w) = \arg\max_{l \in \mathbb{L}} \{|\{s \in S(w) \mid L_1(s) = l\}|\}$

Data: The set of unknown words $W' = \{w_1, \ldots, w_n\}$
Result: The language $\mathcal{L}(w)$ of any word $w_i \in W'$
for $i = 1..n$ **do**
  $\mathrm{L_s}(w_i) \leftarrow \{l \in \mathrm{L} \mid \exists s \in S(w_i) : l = L_1(s)\}$;
  if $|\mathrm{L_s}(w_i)| = 1$ **then**
    $\mathcal{L}(w) \leftarrow L_1(w_i)$;
  end
  else
    $\mathcal{L}(w) \leftarrow L_2(w_i)$;
  end
end

Figure: Lexicon Expander Language Assignment Algorithm

Lexicon Expander

- $S(w) = \{s \in S \mid w \in f(s)\}$

1. $L\colon W \rightarrow \{l_1, \ldots, l_m\} = \mathbb{L}$
2. $L_1(w) = \arg\max_{l \in \mathbb{L}} \{|\{w' \in W' \mid \exists s \in S(w)\colon w' \in f(s) \wedge L_1(w') = l\}|\}$
3. $L_2(w) = \arg\max_{l \in \mathbb{L}} \{|\{s \in S(w) \mid L_1(s) = l\}|\}$

Data: The set of unknown words $W' = \{w_1, \ldots, w_n\}$
Result: The language $\mathcal{L}(w)$ of any word $w_i \in W'$
for $i = 1..n$ **do**
  $\mathrm{L_s}(w_i) \leftarrow \{l \in \mathrm{L} \mid \exists s \in S(w_i) : l = L_1(s)\}$;
  if $|\mathrm{L_s}(w_i)| = 1$ **then**
    $\mathcal{L}(w) \leftarrow L_1(w_i)$;
  end
  else
    $\mathcal{L}(w) \leftarrow L_2(w_i)$;
  end
end

Figure: Lexicon Expander Language Assignment Algorithm

Evaluation: Results

Language	F-Score	Accuracy
German - French	0.40	35.43%
English - Turkish	0.36	38.13%
OHG - Latin	0.79	70.34%

Table: Performance of the LD Toolkit: word level

Language	F-Score	Accuracy
German - French	0.58	53.5%
English - Turkish	0.52	51%
OHG - Latin	0.95	91.78%

Table: Performance of the Lexicon Expander

Conclusion

1. Multilingualism in ancient corpora causes problems for lexicon building
2. The Lexicon Expander post-processes the LD Toolkit output and improves f-score and accuracy scores
3. This safes annotators efforts and simplifies automatic processing

Reference

Sukhareva et al. 2011: Sukhareva, Maria; Islam, Zahurul; Hoenen, Armin; Mehler, Alexander; *A Three-step Model of Language Detection in Multilingual Ancient Texts*, In preparation, 2011;

Mehler et al. 2011: Alexander Mehler, Nils Diewald, Ulli Waltinger, Rüdiger Gleim, Dietmar Esch, Barbara Job, Thomasu Küchelmann, Olga Pustylnikov, and Philippe Blan-u chard. *Evolution of Romance language in written communication: Network analysis of late Latin and early Romance corpora*. Leonardo, 44(3), 2011.

Islam et al. 2011: Islam, Md. Zahurul; Mittmann, Roland und Mehler, Alexander ; *Multilingualism in Ancient Texts: Language Detection by Example of Old High German and Old Saxon*, In GSCL conference on Multilingual Resources and Multilingual Applications (GSCL 2011), 28-30 September, Hamburg, Germany, 2011.

Waltinger and Mehler 2009: Ulli Waltinger and Alexander Mehler, *The feature difference coefficient: Classification by means of feature distributions*, In Proceedings of the Conference on Text Mining Services , Leipziger Beiträge zur Informatik: Band XIV, pages 159–168. Leipzig University, Leipzig.

Acknowledgement

This work is funded by LOEWE Digital- Humanities project in the Goethe-Universität, Frankfurt.

digital humanities

Goethe-Universität Frankfurt | Technische Universität Darmstadt | Freies Deutsches Hochstift

Thank you!