`gramophone` – A hybrid approach to grapheme-phoneme conversion

Kay-Michael Würzner, Bryan Jurish
{wuerzner,jurish}@bbaw.de

FSMNLP
Universität Düsseldorf
24th June 2015

Overview

The Task

- Finding the pronunciation of a word given its spelling

The Challenge: Ambiguity

- a phoneme may be realized by different characters
- a character may be represented by different phonemes

Our Approach: A combination of

- a hand-crafted rule set controlling segmentation and alignment,
- a conditional random field model for generating transcription candidates, and
- an N-gram language model for selecting the "best" grapheme-phoneme mapping

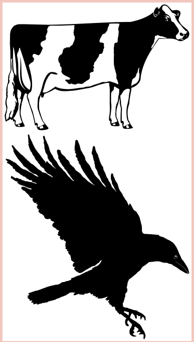
berlin-brandenburgische AKADEMIE DER WISSENSCHAFTEN

Outline

1. Grapheme-phoneme conversion and its applications
2. Existing approaches
3. The `gramophone` approach
   (a) Alignment/Encoding
   (b) Transcription
   (c) Rating
4. Comparative evaluation and error analysis
5. Discussion & Outlook

Grapheme-phoneme conversion: Problem description

- Symbolic representation of the pronunciation of words
- Orthography is ambiguous w.r.t. pronunciation, phonetic alphabets allow for an unambiguous representation



cow	/kaʊ̯/
crow	/kɹoʊ̯/

- Complex alignment: Single characters may be represented by multiple phonemes (and *vice versa*)

ph	oe	n	i	x
f	iː	n	ɪ	ks

Grapheme-phoneme conversion: Applications

Text-to-speech systems
(Black & Taylor 1997)

- Improvement of speech signal synthesis by disambiguation of the input text

Spelling correction / "canonicalization"
(Jurish 2010)

- Phonetic transcriptions as a normal form for identifying spelling variants

Speech recognition
(Galescu and Allen 2002)

- Inverse application of g2p models

Pronunciation dictionaries
(TC-Star project; DWDS)

- Generation of transcriptions or transcriptions candidates especially in compounding languages

Previous work: Rule-based approaches

- Inspired by *The Sound Pattern of English* *(Chomsky & Halle 1968)*
- Equivalent to regular grammars and rewriting systems *(Johnson 1972)*
- Successful model for g2p converters in many languages
- Used in various text-to-speech systems, e.g.
  - `MITalk` *(Allen et al. 1987)*
  - `TETOS` *(Wothke 1993)*
  - `festival` *(Taylor et al. 1998)*
- **Drawbacks:**
  - Expertise and effort required in their production and maintenance
  - Treatment of exceptional pronunciation e.g. in loan words (or even worse *compounds* of foreign and native words)

> *Versaillesdiktat*
>
> /vɛʁzaɪ̯dɪktaːt/
>
> engl. 'Versailles diktat'

Previous work: Statistical approaches

- Automatic inference of regularities in the correspondence of spellings and pronunciations from data (i.e. word+transcription pairs)
- Many large data sets exist
  - NETTalk
  - CELEX
  - wiktionary
- Many more existing approaches *(cf. Reichel et al. 2008)*
  - Neural networks *(Sejnowski & Rosenberg 1987)*
  - Joint-sequence N-gram models *(Bisani & Ney 2008)*
  - Conditional random fields *(Jiampojamarn & Kondrak 2009)*
- **Drawback:**
  - No direct control of results, linguistically implausible transcriptions may be inferred

> *Getue* ↦ */gətʃə/
>
> engl. 'fuss'

Alignment

Starting point

- Association of transcriptions with entire words

  $\rightsquigarrow$ *Alignment on the grapheme-substring level necessary*

- $n : m$ relation between grapheme-phoneme string pairs $\boxed{n, m \in \mathbb{N} \setminus \{0\}}$

Alignment

Starting point

- Association of transcriptions with entire words

  $\rightsquigarrow$ *Alignment on the grapheme-substring level necessary*
- $n : m$ relation between grapheme-phoneme string pairs $\boxed{n, m \in \mathbb{N} \setminus \{0\}}$

ph	oe	n	i	x
$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$
f	iː	n	ɪ	ks

Alignment

Starting point

- Association of transcriptions with entire words

  $\rightsquigarrow$ *Alignment on the grapheme-substring level necessary*

- $n : m$ relation between grapheme-phoneme string pairs $\boxed{n, m \in \mathbb{N} \setminus \{0\}}$

Approaches

- Numerous existing alignment methods *(cf. Reichel 2012)*
- Simplify the $n : m$ relation to a more tractable case $\boxed{n, m \in \{0, 1\}}$

ph	oe	n	i	x
$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$
f	iː	n	ɪ	ks

Alignment

Starting point

- Association of transcriptions with entire words

  $\rightsquigarrow$ *Alignment on the grapheme-substring level necessary*

- $n : m$ relation between grapheme-phoneme string pairs $\boxed{n, m \in \mathbb{N} \setminus \{0\}}$

Approaches

- Numerous existing alignment methods *(cf. Reichel 2012)*
- Simplify the $n : m$ relation to a more tractable case $\boxed{n, m \in \{0, 1\}}$

p	h	o	e	n	i	x	ε
$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$
f	ɛ	iː	ɛ	n	ɪ	k	s

Alignment

Starting point

- Association of transcriptions with entire words

  $\rightsquigarrow$ *Alignment on the grapheme-substring level necessary*
- $n : m$ relation between grapheme-phoneme string pairs $\boxed{n, m \in \mathbb{N} \setminus \{0\}}$

Approaches

- Numerous existing alignment methods *(cf. Reichel 2012)*
- Simplify the $n : m$ relation to a more tractable case $\boxed{n, m \in \{0, 1\}}$
- Application of some Levenshtein-like mechanism *(Levenshtein, 1966)*

p	h	o	e	n	i	x	ε
$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$
f	ɛ	iː	ɛ	n	ɪ	k	s

Alignment

Alternatives?

- *Deletion* doubtful in the context of grapheme-phoneme correspondence
- Inference of many-to-many alignments error-prone *(Jiampojamarn et al. 2007)*
- Linguistically motivated alignment desirable

Constraint-based alignment

- **Manual** definition of possible mappings between grapheme sequences and phonemic realizations $\boxed{M \subset (\Sigma_{\mathrm{G}}^{+} \times \Sigma_{\mathrm{P}}^{+})}$
- Compiled as FST $\boxed{E = \langle Q, \Sigma_{\mathrm{G}} \cup \{|\}, \Sigma_{\mathrm{P}} \cup \{\_\}, q_0, q_0, \delta \rangle}$
  - Add a path $(q_0, q_0, g \cdot |, p \cdot \_)$ for each mapping $(g, p) \in M$
  - '$|$' and '_' are reserved delimiter symbols
- Generate **all admissible** segmentations of a word and its transcription
  - FST I_{G} with a path $(q_0, q_0, g, g \cdot |)$ for every g in the domain of M
  - FST I_{P} with a path $(q_0, q_0, p, p \cdot \_)$ for every p in the codomain of M

Alignment

- Construct letter FSTs W and T for a word w and its transcription t
- Alignment of w and t is generated by a series of compositions which filters out all non-matching pairings $\boxed{A_{W,T} = \pi_2(W \circ I_{\mathrm{G}}) \circ E \circ \pi_2(T \circ I_{\mathrm{P}})}$

Example

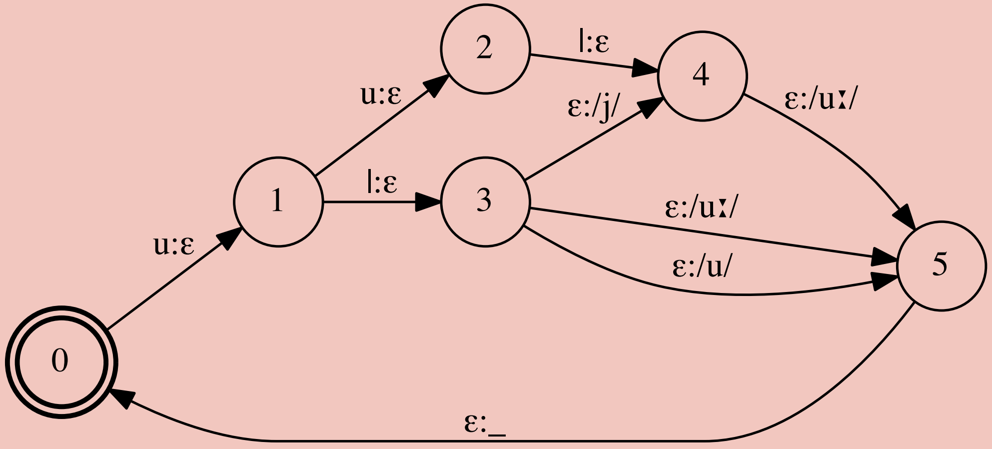
$$M = \{\mathrm{u}{:}/\mathrm{u}/, \mathrm{u}{:}/\mathrm{uː}/, \mathrm{u}{:}/\mathrm{juː}/, \mathrm{uu}{:}/\mathrm{uː}/\}$$

Alignment

- Construct letter FSTs W and T for a word w and its transcription t
- Alignment of w and t is generated by a series of compositions which filters out all non-matching pairings $\boxed{A_{W,T} = \pi_2(W \circ I_{\mathrm{G}}) \circ E \circ \pi_2(T \circ I_{\mathrm{P}})}$

Example

$$M = \{\mathrm{u}{:}/\mathrm{u}/, \mathrm{u}{:}/\mathrm{uː}/, \mathrm{u}{:}/\mathrm{juː}/, \mathrm{uu}{:}/\mathrm{uː}/\}$$



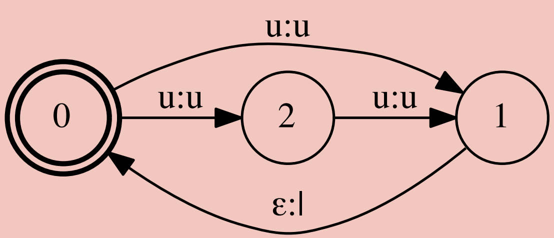
E

Alignment

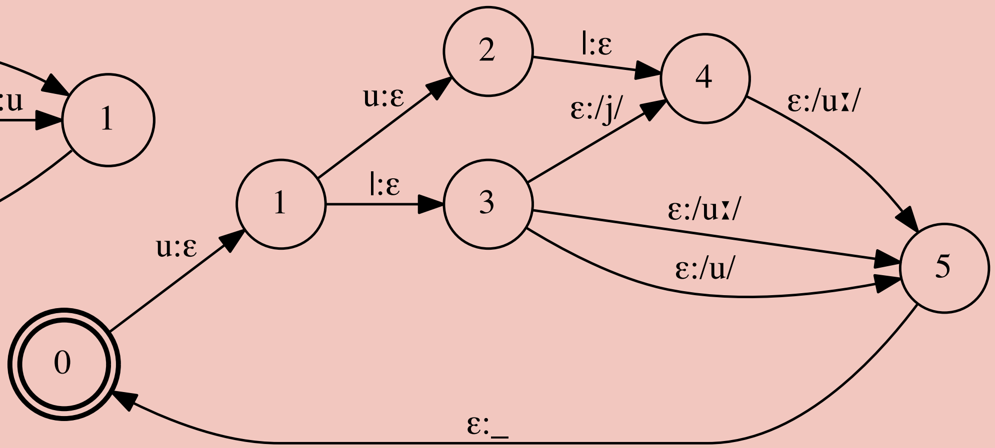
- Construct letter FSTs W and T for a word w and its transcription t
- Alignment of w and t is generated by a series of compositions which filters out all non-matching pairings $\boxed{A_{W,T} = \pi_2(W \circ I_{\mathrm{G}}) \circ E \circ \pi_2(T \circ I_{\mathrm{P}})}$

Example

$$M = \{\mathrm{u}{:}/\mathrm{u}/, \mathrm{u}{:}/\mathrm{uː}/, \mathrm{u}{:}/\mathrm{juː}/, \mathrm{uu}{:}/\mathrm{uː}/\}$$



I_{G}



E

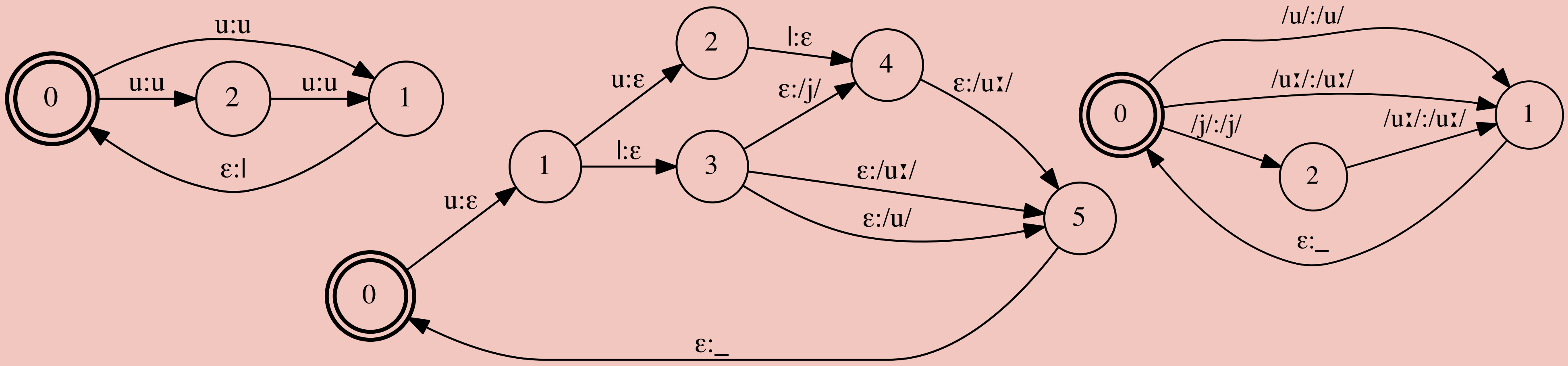
Alignment

- Construct letter FSTs W and T for a word w and its transcription t
- Alignment of w and t is generated by a series of compositions which filters out all non-matching pairings

$$\boxed{A_{W,T} = \pi_2(W \circ I_{\mathrm{G}}) \circ E \circ \pi_2(T \circ I_{\mathrm{P}})}$$

Example

$$M = \{\text{u:/u/}, \text{u:/uː/}, \text{u:/juː/}, \text{uu:/uː/}\}$$



I_{G} E I_{P}

Alignment

Extended mappings

- Procedure allows for more complex mappings, i.e. context restriction
- Treatment of multiple alignments:

matinee : $\mathrm{matine\!:}$

m	a	t	i	ne	e
$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$
m	a	t	i	n	$\mathrm{e\!:}$

$\rightsquigarrow$ Conflicting rules may be disambiguated using lookahead conditions

Segmentation

- I_{G} is used to generate possible grapheme level segmentations for subsequent transcription at runtime

Alignment

Extended mappings

- Procedure allows for more complex mappings, i.e. context restriction
- Treatment of multiple alignments:

matinee : matineː

m	a	t	i	n	ee
$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$	$\updownarrow$
m	a	t	i	n	eː

$\rightsquigarrow$ Conflicting rules may be disambiguated using lookahead conditions

Segmentation

- I_{G} is used to generate possible grapheme level segmentations for subsequent transcription at runtime

Transcription

Idea

- Given aligned word-transcription pairs, transcription may be considered as *sequence labelling* problem
- Grapheme sequences are *observations*, phoneme sequences are *labels*
- Many existing methods, e.g. *Hidden Markov Models*, *Support Vector Machines*, *Conditional Random Fields* *(cf. Erdogan 2010)*

CRFs *(Lafferty et al. 2001)*

- Graph-based model: labels and observations are represented by nodes
- Labelling is based on a set of random variables expressing characteristics of the observation $\rightsquigarrow$ *features*
- Training process computes
  - Transition probabilities
  - Influence (weight) of the pre-defined features
- Runtime: Find the most likely state sequence

Transcription

Features

- Selection of features is a non-trivial task (i.e. no "inference" method)
- Given an input string $o = o_1 \ldots o_n$, gramophone relys only on the (observable) grapheme context
  - Each position i is assigned a feature function f_j^k for each substring of o of length $m = (k - j + 1) \leq N$ within a context window of $N - 1$ characters relative to position i
  - N is the *context size window* or "order" of a gramophone model

$$f_j^k(o, i) = o_{i+j} \cdots o_{i+k} \text{ for } -N < j \leq k < N$$

$N = 1$	o_{i-3}	o_{i-2}	o_{i-1}	o_i	o_{i+1}	o_{i+2}	o_{i+3}
	m	a	t	**i**	n	e	e

Transcription

Features

- Selection of features is a non-trivial task (i.e. no "inference" method)
- Given an input string $o = o_1 \dots o_n$, `gramophone` relys only on the (observable) grapheme context
  - Each position i is assigned a feature function f_j^k for each substring of o of length $m = (k - j + 1) \leq N$ within a context window of $N - 1$ characters relative to position i
  - N is the *context size window* or "order" of a `gramophone` model

$$f_j^k(o, i) = o_{i+j} \cdots o_{i+k} \text{ for } -N < j \leq k < N$$

$N = 2$	o_{i-3}	o_{i-2}	o_{i-1}	o_i	o_{i+1}	o_{i+2}	o_{i+3}
	m	a	**t**	**i**	**n**	e	e

Transcription

Features

- Selection of features is a non-trivial task (i.e. no “inference” method)
- Given an input string $o = o_1 \ldots o_n$, `gramophone` relys only on the (observable) grapheme context
  - Each position i is assigned a feature function f_j^k for each substring of o of length $m = (k - j + 1) \leq N$ within a context window of $N - 1$ characters relative to position i
  - N is the *context size window* or “order” of a `gramophone` model

$$\boxed{f_j^k(o,i) = o_{i+j} \cdots o_{i+k} \text{ for } -N < j \leq k < N}$$

$N = 3$	o_{i-3}	o_{i-2}	o_{i-1}	o_i	o_{i+1}	o_{i+2}	o_{i+3}
	m	**a**	**t**	**i**	**n**	**e**	e

Rating

Idea

- Select the "best" transcription from the segmented and labeled candidates
- Statistical model defined over strings of grapheme-phoneme segment pairs *("graphones")*
- N-gram model: joint probability as product of conditional probabilities under Markov assumptions $\boxed{P(gp_0 \ldots gp_n) \approx \prod_{i=0}^{n} P(gp_i | gp_{i-N} \ldots gp_{i-1})}$

Implementation

- Interpolate all k-gram distributions with $1 \leq k \leq N$ *(Jelinek & Mercer 1980)*
- Combined with Kneser-Ney discounting for treatment of out-of-vocabulary items *(Kneser & Ney 1995)*
- Model parameters are estimated from (aligned) word-transcription pairs
- Implementable within the finite-state calculus *(Pereira & Riley 1997)*

Experiments

Corpora & Mappings

- **de-LexDB** : 71,481 words, 277 graphone types *(Gibbon & Lüngen 2000)*
- **de-Wiki** : 147,359 words, 589 graphone types (*http://de.wiktionary.org*)
- **en-CELEX**: 73,736 words, 463 graphone types *(Baayen et al. 1995)*

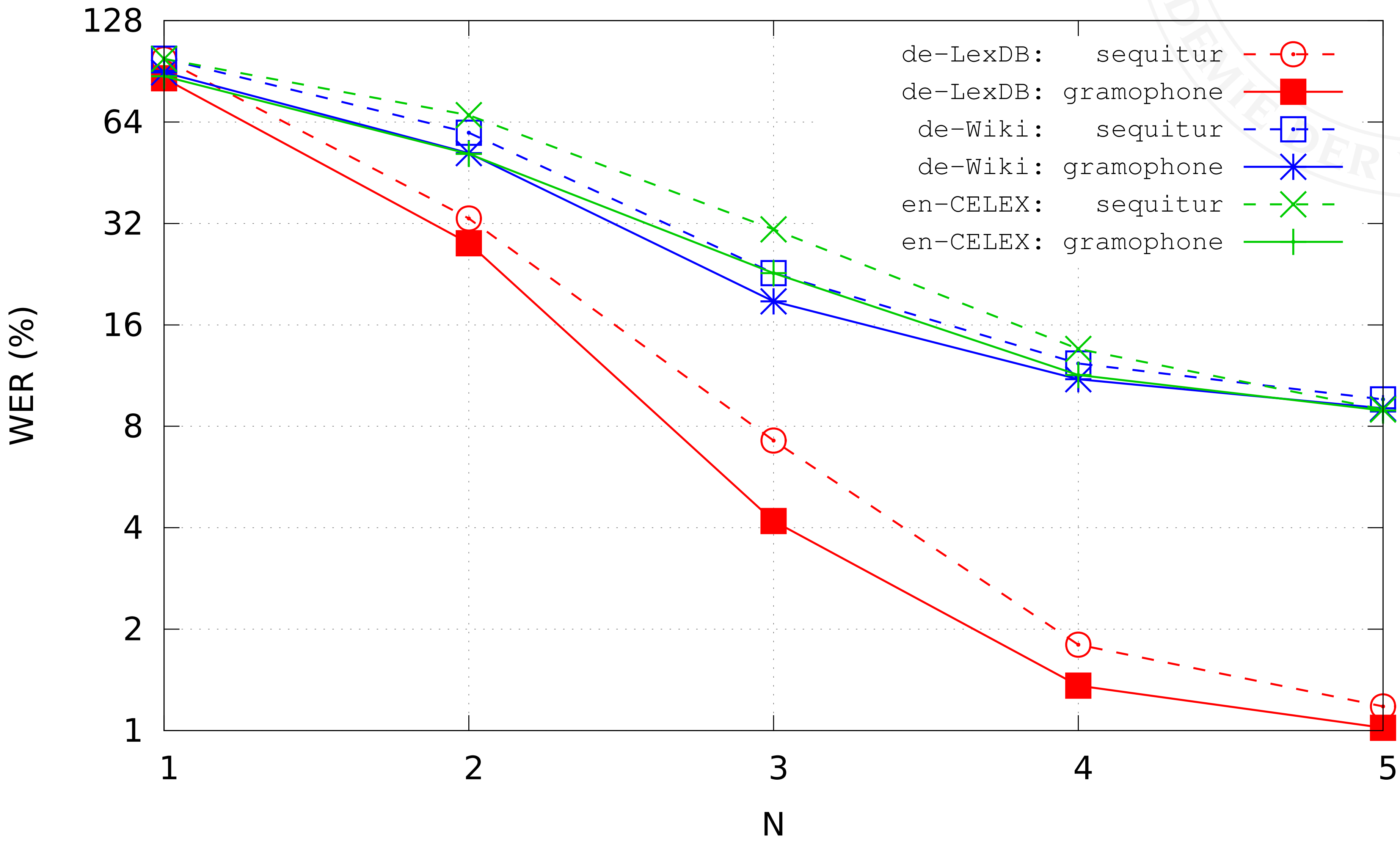
Method

- Compare `gramophone` ***versus*** `sequitur` *(Bisani & Ney 2008)*
- Test model orders $N \in \{1, 2, 3, 4, 5\}$ using 10-fold cross validation
- Investigate both **word** and **phoneme** error rates (WER, PER)

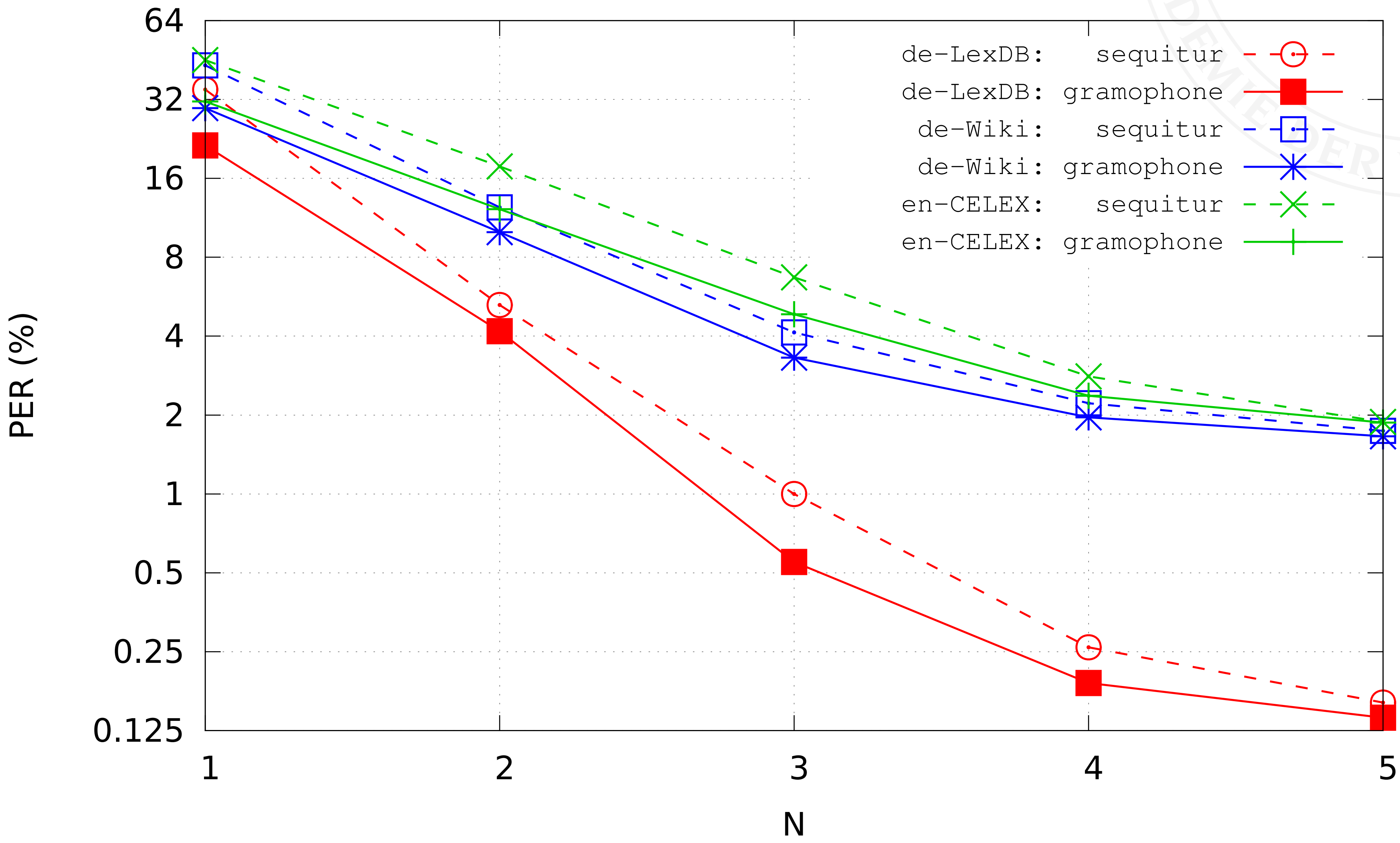
Implementation

- `OpenFST` for alignment and segmentation *(Allauzen et al. 2007)*
- `wapiti` for CRF training and application *(Lavergne et al. 2010)*
- `OpenGRM` for candidate rating *(Roark et al. 2012)*

Results: Word Error Rate



Results: Phoneme Error Rate



Results: Discussion

General Trends

- `gramophone` outperformed `sequitur` for all conditions tested
- performance gain drops as model order increases, negligible for $N = 5$
- upper bound imposed by mapping heuristics beyond $N = 5$?
- LexDB performance looks suspiciously good
  - LexDB data were to a large extent automatically generated *(Lüngen p.c.)*

Interesting Phenomena

- de-Wiki: 25% of the phoneme errors concern schwa deletion
- de-Wiki: glottal stop is not a big issue
- en-CELEX: more uniform distribution of errors, largest class is schwa $\leftrightarrow V$ (22%)

	ən/n̩	əl/l̩	əm/m̩	ʔ/ ¬ ʔ
seq	5114	756	307	172
gp	5010	633	299	146

Summary & Outlook

What We Did (instead of summer holidays)

- Novel conversion method based on three simple steps
  - Manually driven alignment/segmentation candidate generation
  - Candidate transcription with CRFs
  - Selection of the most likely candidate using N-gram LM
- Performance comparable to a state-of-the-art method

Still To Do

- Upper bound on performance imposed by segmentation heuristics *(?)*
- (Approximate) implementation using (weighted) finite-state methods *(?)*
  - Transducer (segmentation) $\leftrightarrow$ pair acceptor (LM)
  - Linear chain CRFs $\not\equiv$ (W)FSTs
- Extensions
  - Integrate results of preceding morphological analysis
  - Predict syllabification, stress patterns

The End

/ði ɛnd/

Thank you for listening!