Canonicalization techniques for computer-mediated communication

Bryan Jurish,[1] Kay-Michael Würzner,[2] Maria Ermakova,[1] Sophie Arana[1]
{jurish,wuerzner,ermakova}@bbaw.de

(1) Berlin-Brandenburgische Akademie der Wissenschaften (2) Universität Potsdam

GSCL-2013 Workshop *"Verarbeitung . . . internetbasierter Kommunikation"*

Darmstadt, 23rd September 2013

Overview

The Big Picture

- The Situation
- The Problem
- The Approach

Canonicalization Methods

- Type-wise Conflation
  - Transliteration, Phonetization, Rewrite Cascade
- Token-wise Disambiguation
  - Dynamic Hidden Markov Model
- Problems & Workarounds
  - Lexical, Morphological, Phonological, Suprasegmental

Summary & Outlook

— The Big Picture —

Deutsches Textarchiv (DTA)

Primary Goals

- digitize ∼ 1,300 print volumes, printed ∼ 1600-1900
  - first editions of respective works
  - detailed metadata, highly accurate transcriptions
- TEI-XML corpus encoding & storage
  - DTA base format (DTABf) dialect
- linguistic analysis (automated)
  - tokenization, normalization, PoS-tagging, lemmatization
- online search (DDC) `http://www.ddc-concordance.org`
  - lemma-based, PoS-sensitive, spelling-tolerant

In Numbers

(DTA+DTAE 2013-05-02)

1,276	transcribed works
353,245	digitized pages
567,200,587	unicode characters
81,049,777	tokens (alpha-numeric)
1,947,414	types (alpha-numeric)

TODO: UPDATE FOR IBK CORPUS

The Situation

CMC Text $\not\ni$ Orthographic Conventions

- also applies to historical text, OCR output, ...
- many non-standard graphemic forms

nicht *"not"*	`nciht, nert, net, nich, nicha, niiiiiiicht, nit, nnicht, ...`
tschüss *"bye"*	`schüssi, tschaui, tschö, tschööööö, tschüssi, tschüssie, tschüüüüüüüüüüüüssssss, ...`

Conventional NLP Tools $\Rightarrow$ Strict Orthography

- IR systems, PoS taggers, stemmers, lemmatizers, morphological analyzers, parsers, ...
- Fixed lexicon keyed by (ortho)graphic form
- Conventional lexemes only *($\approx$ newspaper domain)*

The Problem

	Conventional	**Tools**
$\oplus$	**Unconventional**	**Corpus**
=		***Soup***

- Corpus variants ***missing*** from application lexicon
  - ***low coverage*** (many unknown types)
  - ***poor recall*** (relevant data not retrieved)
  - ***spurious "noise"*** (poor model fit)
  - *. . . and more!*

The Approach: Canonicalization

a.k.a. (orthographic) 'standardization', 'normalization', . . .

c	u	l8r	doooood
↓	↓	↓	↓
see	*you*	*later*	*dude*

In a Nutshell

- *Map* each word w to a unique canonical cognate $\widetilde{w}$
- *Defer* application analysis to canonical forms
- DTA::CAB infrastructure developed for historical text

Canonical Cognates

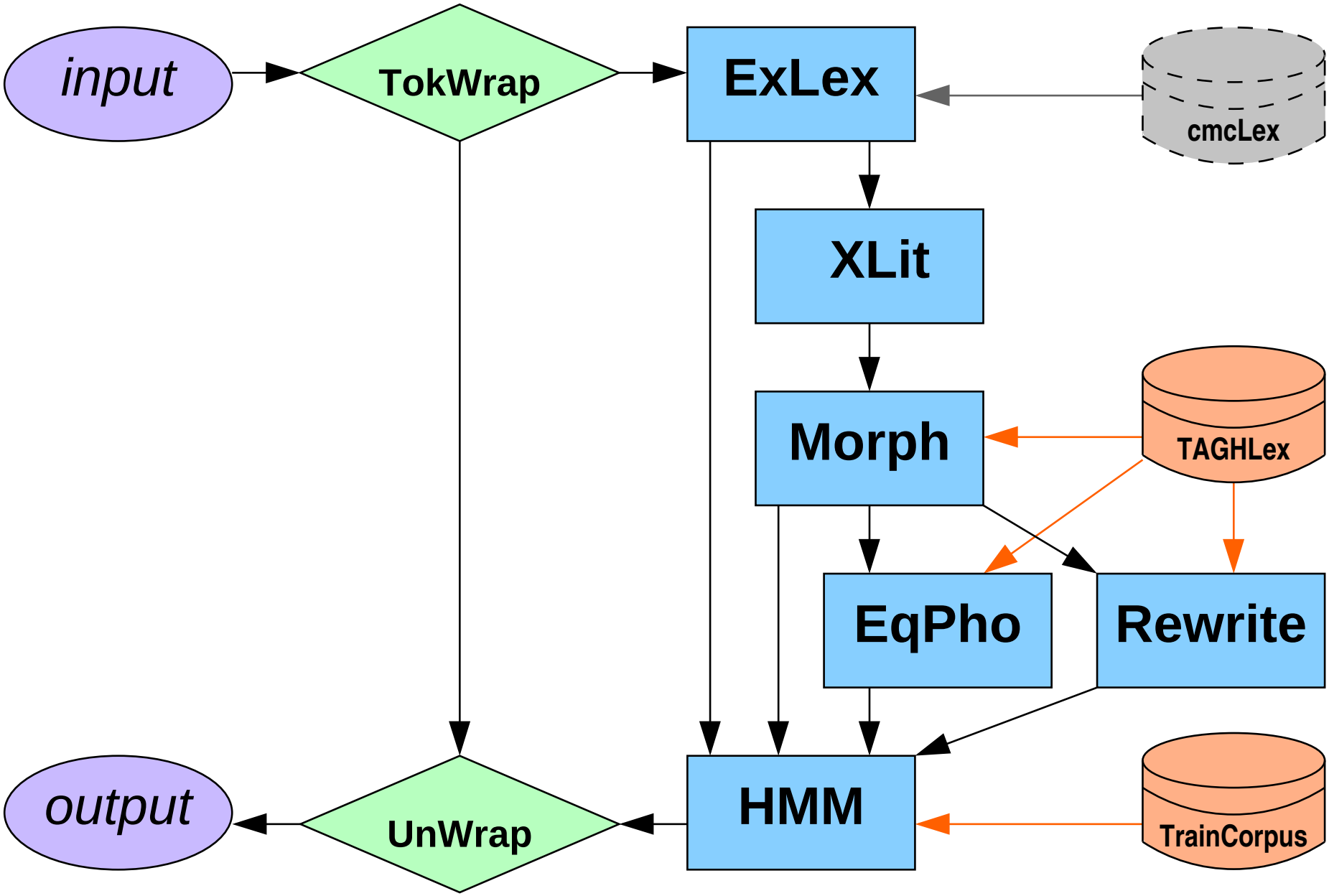
- Lexically active conventional equivalent(s) $\widetilde{w} \in \mathrm{Lex}$
- Preserve both root and relevant features of input

Conflation Relation $\sim_r$

- *Binary relation* on strings (words) in $\mathcal{A}^*$
- Prototypically a true *equivalence relation*

— Canonicalization Methods —

DTA::CAB System Architecture



Deterministic Transliteration (xlit)

Sketch

$$w \sim_{\text{xlit}} v :\Leftrightarrow \text{xlit}^*(w) = \text{xlit}^*(v)$$

- **Idea:** handle *non-standard characters* *(Jurish 2008, 2010b,c)*
- **Implementation:** $\mathcal{O}(1)$ character lookup table
- useful as *preprocessor* for subsequent methods

Examples

Foreign Scripts	北京 ↦	BeiJing	*"Beijing"*
	Πλάτων ↦	Platon	*"Plato"*
Diacritics	sçhon ↦	schon	*"already"*
	schøn ↦	schön	*"beautiful"*
Ligatures	œde ↦	öde	*"bleak"*
	Ærger ↦	Ärger	*"trouble"*
(pesudo-) Allographs	hu©ħ ↦	huch	*"oops"*

TAGH Morphology (morph)

Sketch

(Geyken & Hanneforth, 2006)

- Model (conventional) morphological processes as WFST
- Provides *weighted target language* for subsequent methods
  - analysis cost $\approx$ derivational complexity
- "Known-wins" filtering

$$w \mapsto w :\Leftarrow w \in \pi_1(M_{\text{morph}})$$

Overgeneration: 'known' form *shouldn't* always win

hab $\mapsto$	*hab$_{\texttt{VVIMP}}$	$\neq$ habe	*"have"*
Andre $\mapsto$	*André$_{\texttt{NE}}$	$\neq$ Andere	*"other"*
muste $\mapsto$	*mus|te	$\neq$ muss|te	*"must"*
Grad $\mapsto$	*Grad	$\neq$ Gerade	*"directly"*
schafsinnig $\mapsto$	*schaf|sinnig	$\neq$ scharfsinnig	*"perceptive"*

- **Workaround:** *safety heuristics*, e.g. *$\texttt{V.imp}$, *$\texttt{NE}$, *$f = 0$

Phonetic Equivalence (eqpho)

Sketch

$$w \sim_{\text{pho}} v :\Leftrightarrow \text{pho}(w) = \text{pho}(v)$$

- **Idea:** conflate words by *phonetic form* *(Jurish, 2008, 2010b)*
- **Implementation:** text-to-speech rule-set *(Möhler et al., 2001)*
  - modified & compiled as FST M_{pho}
  - online $k-$best equivalence cascade search *(Jurish, 2010a)*

$$C_{\text{eqpho}} := M_{\text{pho}} \circ M_{\text{pho}}^{-1} \circ \pi_1(M_{\text{morph}})$$

Examples

Successes	betrib	$\mapsto$	Betrieb	*"operation"*
	eindoitig	$\mapsto$	eindeutig	*"univocal"*
	hallloooo	$\mapsto$	hallo	*"hello"*
Failures	mirsch	$\not\mapsto$	mich	*"me"*
	nischt	$\not\mapsto$	nicht	*"not"*

- **Workarounds:** target language pruning, cascade lookup cutoffs

Rewrite Cascade (rw)

Sketch

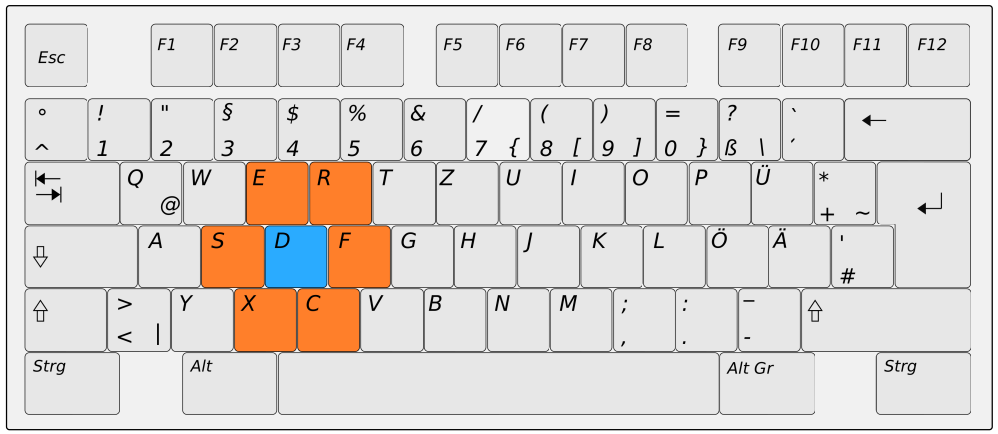
$$\begin{aligned} \mathrm{best}_{\mathrm{rw}}(w) &:= \arg\min_{v \in \mathcal{A}^*} [\![M_{\mathrm{rw}} \circ \pi_1(M_{\mathrm{morph}})]\!](w, v) \\ w \sim_{\mathrm{rw}} v &:\Leftrightarrow \mathrm{best}_{\mathrm{rw}}(w) = \mathrm{best}_{\mathrm{rw}}(v) \end{aligned}$$

- **Idea:** map words to *nearest* conventional type *(Jurish, 2010b,c)*
  - generalized string edit distance *(Levenshtein, 1966)*
  - computable even for infinite lexica *(Mohri, 2002; Jurish, 2010a)*
- **Implementation:** online $(k = 1)$−best cascade search
  - editor WFST M_{rw} compiled from *keyboard adjacency model*
  - weight interpolation constants λ_{rw} and λ_{morph}

Keyboard Adjacency

- limit search space
- attenuate edit costs
- insert, delete, replace

  . . . but **NOT** transpose!



Error Model M_{rw}

Generic Edit Operations

Deletion	guet ↦ gut	*"good"*
Insertion	is ↦ ist	*"is"*
Substitution	goht ↦ geht	*"goes"*
Transposition	udn ↦ und	*"and"*
Doubling	das ↦ dass	*"that"*
Un-doubling	neuees ↦ neues	*"new"*
Un-potentiation	juhuuuu ↦ juhu	*"woo-hoo"*

Keyboard-Adjacent Edit Operations

Deletion	iommer ↦ immer	*"always"*
	hoi ↦ hi	*"hi"*
Insertion	z ↦ zu	*"to"*
	schaf ↦ scharf	*"sharp"*
Substitution	guvkt ↦ guckt	*"looks"*

Token-wise Disambiguation (hmm)

Idea

(Mays et al., 1991; Brill & Moore, 2000; Jurish, 2010c)

- Allow high-recall overgeneration at type level
- Recover precision using token-level context

Implementation: Dynamic Hidden Markov Model (HMM)

- **States** are word-conflator pairs $\mathcal{Q} = (\mathcal{W} \cup \{\mathbf{u}\}) \times R$
- **Observations** are input strings $\mathcal{O}_S = \bigcup_{i=1}^{n_S} \{w_i\} \subset \mathcal{A}^*$
- **Transitions** *(static)* $A\big(\langle \widetilde{w}_i, r_i \rangle_{i=1}^m\big) \approx p\big(\widetilde{w}_m \big| \widetilde{w}_1^{m-1}\big)$
- **Lexicon** *(dynamic)*: Maxwell-Boltzmann distribution

$$B\big(\langle \widetilde{w}, r \rangle, w\big) \approx \frac{b^{\beta d_r(w,\widetilde{w})}}{\sum_{r' \in R} \sum_{\widetilde{w}' \in \downarrow[w]_{r'}} b^{\beta d_{r'}(w,\widetilde{w}')}}$$

  - b, β are global model parameters ($b \geq 1, \beta \leq 0$)
  - $d_r(w, \widetilde{w})$ depends on conflator r
- **Lookup** (`moot`): Viterbi Algorithm *(Viterbi, 1967)*

HMM Example

Input `hallooo` `wida` `muste` `kuz` `w€g`

HMM Example

Input	hallooo	wida	muste	kuz	wEg
xlit	*hallooo*	*wida*	*muste*	*kuz*	*wEg*

HMM Example

Input	`hallooo`	`wida`	`muste`	`kuz`	`wɛg`
xlit	*hallooo*	*wida*	*muste*	*kuz*	*wEg*
pho	{*hallo*}	{*wider*, *wieder*}	{*muste*}	{*Cuts*, *Kuts*}	{*weck*, *weg*}

HMM Example

Input	`hallooo`	`wida`	`muste`	`kuz`	`wEg`
xlit	*hallooo*	*wida*	*muste*	*kuz*	*wEg*
pho	{*hallo*}	{*wider*, *wieder*}	{*muste*}	{*Cuts*, *Kuts*}	{*weck*, *weg*}
rw	*hallo*	*Weda*	*musste*	*kurz*	*weg*

HMM Example

Input	hallooo	wida	muste	kuz	wEg
xlit	*hallooo*	*wida*	*muste*	*kuz*	*wEg*
pho	{*hallo*}	{*wider*, *wieder*}	{*muste*}	{*Cuts*, *Kuts*}	{*weck*, *weg*}
rw	*hallo*	*Weda*	*musste*	*kurz*	*weg*
hmm	*hallo*	*wieder*	*musste*	*kurz*	*weg*

HMM Example

Input	`hallooo`	`wida`	`muste`	`kuz`	`wEg`
xlit	*hallooo*	*wida*	*muste*	*kuz*	*wEg*
pho	{*hallo*}	{*wider*, *wieder*}	{*muste*}	{*Cuts*, *Kuts*}	{*weck*, *weg*}
rw	*hallo*	*Weda*	*musste*	*kurz*	*weg*
hmm	*hallo*	*wieder*	*musste*	*kurz*	*weg*
Output	**hallo**	**wieder**	**musste**	**kurz**	**weg**

Some Anticipated Difficulties

Lexical

- neologisms & “missing” lexemes
- **Workaround:** static exception lexicon M_{exlex}

ne $\mapsto$ eine *“a”*
ma $\mapsto$ mal *“once”*

Morphological*

- “missing” productive processes
- **Workaround:** M_{morph} plug-in(s)

glaub $\mapsto$ glaube *“believe”*
welch $\mapsto$ welchen *“which”*

* . . . or typographical simulations of phonological processes acting on inflectional morphemes?

Phonetic / Phonological

- identity criterion is too strict
- **Workaround:** M_{rw} plug-in(s)

sehn $\sim$ sehen *“see”*
zehn $\not\sim$ Zehen *“ten”*

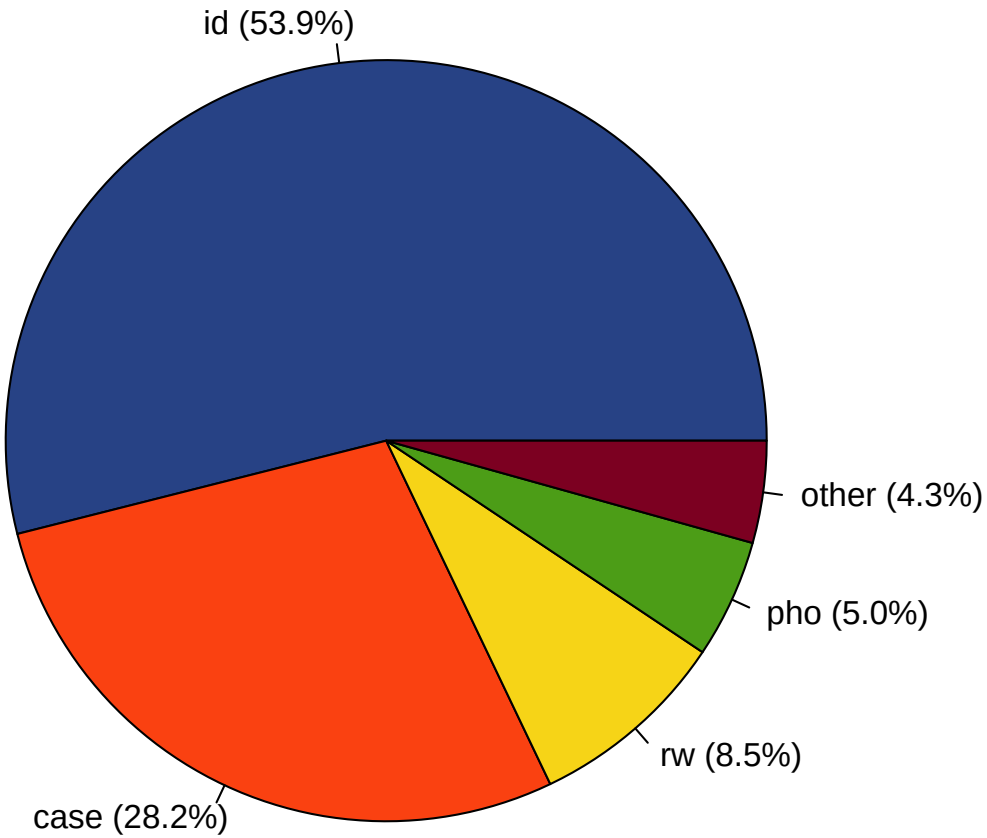
Suprasegmental

- unclear token boundaries
- **Workaround:** ***???***

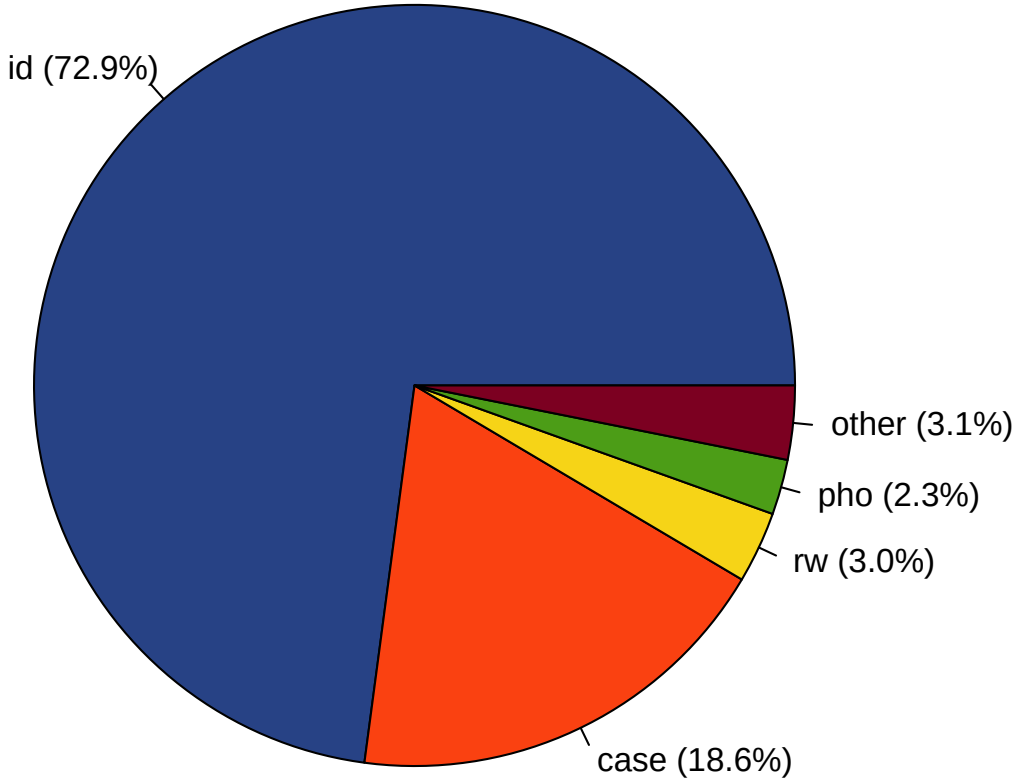
hat’s $\mapsto$ hat es *“has it”*
kannste $\mapsto$ kannst du *“can you”*

Observed Mappings

by Type $(N = 4340)$



by Token $(N = 13611)$



Concluding Remarks

Unconventional Text and Conventional Tools

won't play together nicely "out of the box"

Canonicalization Methods

- transliteration — $\sim$ *quick and dirty*
- phonetic equivalence — $\sim$ *elegant but coarse*
- rewrite cascade — $\sim$ *flexible but costly*
- HMM disambiguator — $\sim$ *precision recovery*

Future Work

- part-of-speech annotations — *STTS*
- plug-in resources — M_{exlex}, M_{rw}, M'_{morph}
- model parameter optimization — $\geq$*27 free parameters*

C U L8R DØØD$

(“The End”)

Thank you for listening!