`Dsolve` – Morphological Segmentation for German using Conditional Random Fields

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

SFCM
Universität Stuttgart
17th September 2015

Outline

- Morphological analysis
- Existing approaches
- Morphological segmentation as sequence labeling (⇝ `Dsolve` approach)
- Experiments
- Discussion & Outlook

Morphological analysis

Goal

- identification & classification of
  - operations
  - operands

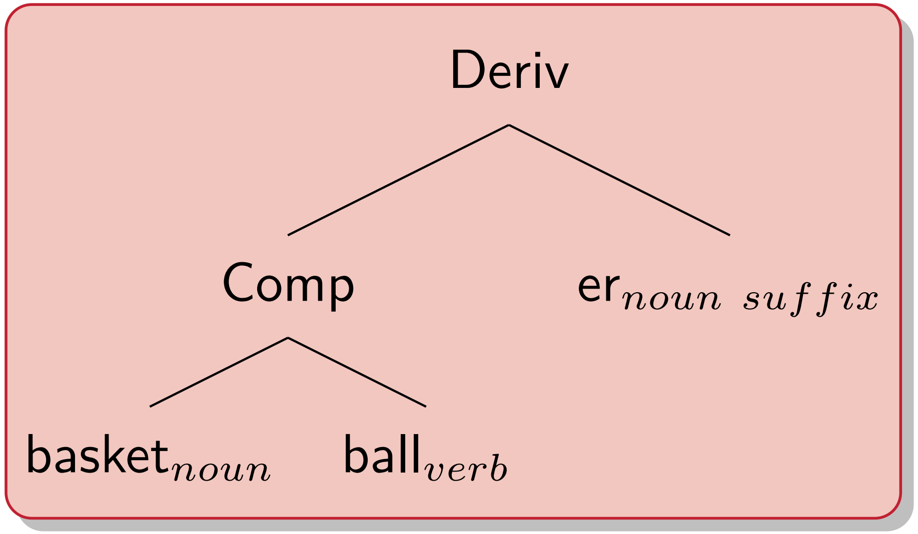
  …forming complex words

Operations

- compounding
- derivation
- inflection

Operands

- morphemes ($\leadsto$ *deep* analysis), or
- morphs ($\leadsto$ *surface* analysis)



Morphological analysis: ambiguity

. . . w.r.t. *Identification*

- > 1 segmentation possible

Ministern

[mini$_{adj}$][Stern$_{noun}$] [Minister$_{noun}$][n$_{dat.\ pl.}$]

'mini-star' 'ministers'

. . . w.r.t. *Classification*

- > 1 category available

Sammelei

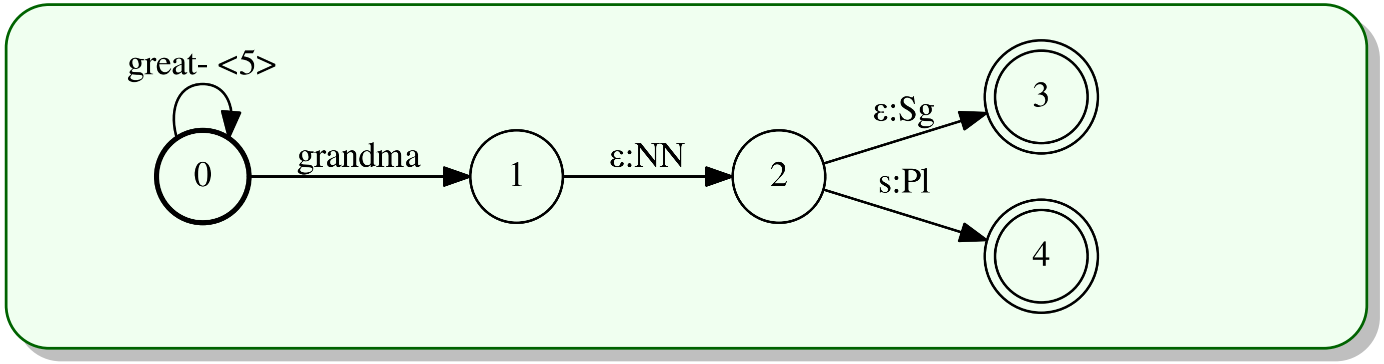
[sammel$_{verb}$][Ei$_{noun}$] [sammel$_{verb}$][ei$_{noun\ suffix}$]

'collector's egg' 'compilation'

Existing approaches: finite-state methods

- Finite lexicon & regular rules using (weighted) finite-state transducers

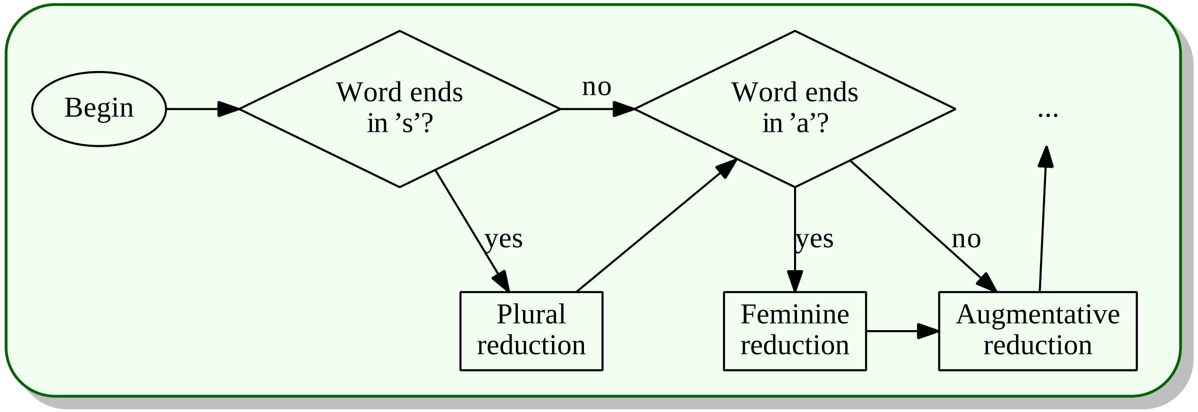
  (cf. Karttunen & Beesley, 2003)



- Tropical semiring weights as measure of complexity
  - word formation processes associated with non-negative costs
  - prefer minimal-cost (least complex) analyses
- German: e.g. `SMOR`, `TAGH` *(Schmid et al. 2004; Geyken & Hanneforth 2005)*

Existing approaches: affix removal

- Identify & remove bound morphemes (prefixes, suffixes) *(Porter 1980)*
  - assume remaining material is the stem
- Usually implemented as series of cascaded rewrite heuristics *(Moreira & Huyck 2001)*



- No (exhaustive) lexicon necessary
- Syllable (CV) structure supports affix removal
- Works best for non-compounding languages;
  - has also been applied to German *(Reichel & Weinhammer 2004)*

Existing approaches: morphology induction

Basic Idea

- bootstrap segmentation model from **un-annontated raw text**
- traceable back to Harris' notion of "Successor Frequency"

$$\boxed{\mathrm{SF}(w, i) = \mathrm{outDegree}\big(\mathrm{ptaNode}(w_1 \cdots w_i)\big)}$$

- SF peaks indicate morpheme boundaries

Heuristic Approaches

(e.g. Goldsmith 2001)

- minimum stem length, maximum affix length, minimum # stems / suffix, ...
- tend to under-segment words (poor recall)

Stochastic Approaches

(e.g. Creutz & Lagus 2002, 2005)

- incremental greedy MDL segmentation $\rightsquigarrow$ hierarchical model
- tend to over-segment words (poor precision)

Existing approaches: summary

	+rules	**-rules**
+lexicon	finite-state morphology	`Dsolve`
-lexicon	affix removal stemming	morphology induction

Existing approaches: summary

	+rules	**-rules**
+lexicon	finite-state morphology	`Dsolve`
-lexicon	affix removal stemming	morphology induction

- Lexicon- & grammar-creation $\rightsquigarrow$ *very labor-intensive*
- Hard to debug, hard to maintain
- Efficient implementations available
- **Very good** analysis quality

Existing approaches: summary

	+rules	**-rules**
+lexicon	finite-state morphology	`Dsolve`
-lexicon	affix removal stemming	morphology induction

- Grammar creation requires much less manual effort than FSM
- Hard to debug, tricky to implement efficiently
- Ambiguity handling $\rightsquigarrow$ difficult
- **Mediocre** analysis quality

Existing approaches: summary

	+rules	-rules
+lexicon	finite-state morphology	`Dsolve`
-lexicon	affix removal stemming	morphology induction

- Least labor-intensive (given an induction algorithm)
- No direct influence on resulting grammar (only via training-corpus selection)
- Inherent ranking of multiple available analyses
- **Insufficient** analysis quality (for production applications)

Segmentation $\sim$ Labeling: binary classification

- Sequence classification
  - Set of observation symbols O, set of classes C
  - Map an observation $o = o_1 \dots o_n$ onto the most probable string of classes $c = c_1 \dots c_n$ using an underlying statistical model
- Observations O: surface character alphabet *(Klenk & Langer 1989)*
- Classes $C = \{0, 1\}$ where

$$c_i = \begin{cases} 1 & \text{if } o_i \text{ is followed by a morph boundary} \\ 0 & \text{otherwise} \end{cases}$$

- Example *Ge.folg.s.leute.n* ("henchmen$_{[\text{DATIVE}]}$")

G	e	f	o	l	g	s	l	e	u	t	e	n
0	1	0	0	0	1	1	0	0	0	0	1	0

Segmentation ∼ Labeling: span-based classes

- Span-based annotation (Ruokolainen et al. 2013)
- Observations O: surface character alphabet
- Classes $C = \{B, I, E, S\}$ where

$$c_i = \begin{cases} S & \text{if } o_i \text{ is preceded and followed by a morph boundary} \\ B & \text{otherwise, if } o_i \text{ is preceded by a morph boundary} \\ E & \text{otherwise, if } o_i \text{ is followed by a morph boundary} \\ I & \text{otherwise} \end{cases}$$

- Example ⟨*Ge*⟩⟨*folg*⟩⟨*s*⟩⟨*leute*⟩⟨*n*⟩ ("henchmen$_{[\textsc{dative}]}$")

G	e	f	o	l	g	s	l	e	u	t	e	n
B	E	B	I	I	E	S	B	I	I	I	E	S

Segmentation ∼ Labeling: typed boundary classes

- Classification of morph boundaries
- Observations O: surface character alphabet
- Classes $C = \{+, \#, \sim, 0\}$ where

$$c_i = \begin{cases} + & \text{if } o_i \text{ is the final character of a prefix} \\ \# & \text{otherwise, if } o_i \text{ is is the final character of a free morph} \\ \sim & \text{otherwise, if } o_{i+1} \text{ is the initial character of a suffix} \\ 0 & \text{otherwise} \end{cases}$$

- Example *Ge+folg∼s#leute∼n* ("henchmen$_{[\text{DATIVE}]}$")

G	e	f	o	l	g	s	l	e	u	t	e	n
0	+	0	0	0	∼	#	0	0	0	0	∼	0

Dsolve

- ***Surface*** analysis of German words using sequence labeling
- ***Type-sensitive*** classification scheme
- Conditional Random Field model predicts boundary **location** and **type**
- Features for an input string $o = o_1 \ldots o_n$ use only observable 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 window size* or "order" of the `Dsolve` model ($\not\equiv$ CRF order)

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

- Trained on modest set of manually annotated data

Experiments

Materials

- Manual annotation of 15,522 distinct German word-forms
  - types and locations of word-internal morph boundaries
- For reference: canoo.net, *Etymologisches Wörterbuch des Deutschen*

Boundary type	#/Boundaries	#/Words
prefix-stem (+)	4,078	3,315
stem-stem (#)	5,808	5,543
stem-suffix (~)	11,182	8,347
TOTAL	21,068	11,967

- Published under the CC BY-SA 3.0 license:
  http://kaskade.dwds.de/gramophone/de-dlexdb.data.txt

Experiments

Method

- Report inter-annotator agreement for a data subset
- Compare morph boundary **detection** of `Dsolve`$[\pm\text{types}]$ CRF approach to
  - Morfessor FlatCat *(Grönroos et al. 2014)*
  - Span-based morph annotation *(Ruokolainen et al. 2013)*
- Test performance w.r.t morph boundary **classification** $[+\text{types}]$
- Test model orders $1 \leq N \leq 5$ using 10-fold cross validation
- Report **precision** (pr), **recall** (rc), **harmonic average** (F), and **word accuracy** (acc)

Implementation

- `wapiti` for CRF training and application *(Lavergne et al. 2010)*

Experiments: evaluation measures

Given a finite set W of annotated words and a finite set of boundary classes C (with the non-boundary class $0 \in C$), we associate with each word $w = w_1 w_2 \ldots w_m \in W$ two partial boundary-placement functions

$$\begin{aligned} B_{\mathrm{relevant},w} &: \mathbb{N} \to C\backslash\{0\} : i \mapsto c :\Leftrightarrow c \text{ occurs at position } i \text{ in } w \\ B_{\mathrm{retrieved},w} &: \mathbb{N} \to C\backslash\{0\} : i \mapsto c :\Leftrightarrow c \text{ predicted at position } i \text{ in } w \end{aligned}$$

and define

$$\begin{array}{llll} \textbf{Precision} & \mathrm{pr} & := & \frac{|\mathrm{relevant} \cap \mathrm{retrieved}|}{|\mathrm{retrieved}|} \\ \textbf{Recall} & \mathrm{rc} & := & \frac{|\mathrm{relevant} \cap \mathrm{retrieved}|}{|\mathrm{relevant}|} \\ \textbf{F-score} & \mathrm{F} & := & \frac{2 \cdot \mathrm{pr} \cdot \mathrm{rc}}{\mathrm{pr} + \mathrm{rc}} \\ \textbf{Accuracy} & \mathrm{acc} & := & \frac{|\{w \in W \mid B_{\mathrm{retrieved},w} = B_{\mathrm{relevant},w}\}|}{|W|}, \text{ where:} \end{array}$$

$$\begin{aligned} \mathrm{relevant} &:= \{(w,i,c) \mid (i \mapsto c) \in B_{\mathrm{relevant},w}\} \\ \mathrm{retrieved} &:= \{(w,i,c) \mid (i \mapsto c) \in B_{\mathrm{retrieved},w}\} \end{aligned}$$

Experiments: inter-annotator agreement

- Independent 2nd manual annotation of a data subset ($n = 1000$) by an expert
- Our own annotation serves as the "gold standard" (i.e. *relevant*)

Boundary Symbol	pr%	rc%	F%	acc%
$+$	92.05	97.20	94.56	n/a
#	96.01	93.28	94.63	n/a
$\sim$	93.28	92.66	92.97	n/a
TOTAL$[+\text{types}]$	93.74	93.74	93.74	87.40
TOTAL$[-\text{types}]$	96.20	96.20	96.20	87.40

- Reasonably high agreement with discrepancies particularly w.r.t.:
  - latinate word formation (e.g. *volunt(˜)aristisch*, "voluntaristic")
  - prefixion $\leftrightarrow$ compounding (e.g. **weg+gehen* vs. *weg#gehen*, "to leave")

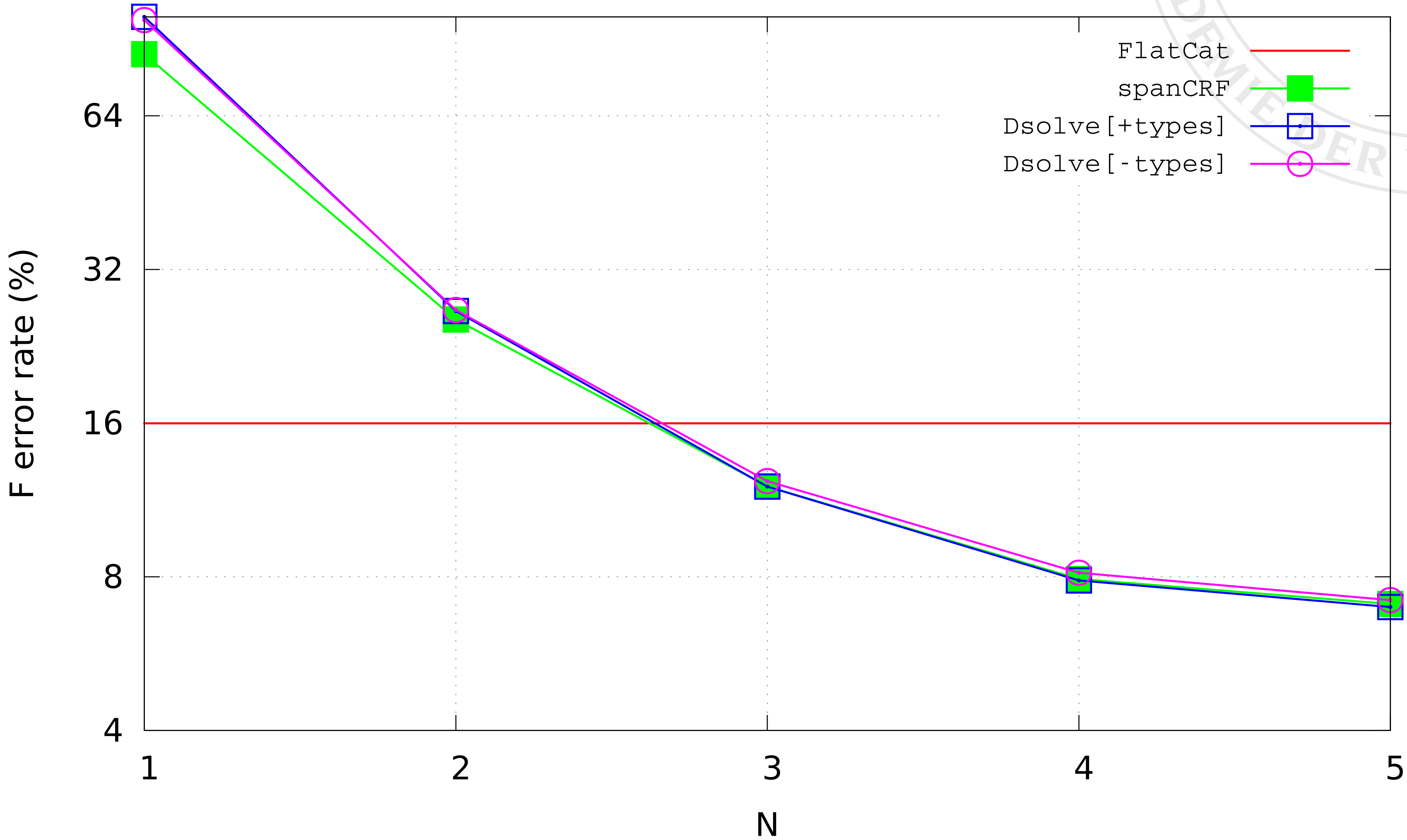
Experiments: boundary detection

Comparison of three different approaches (*retrieved*) with manual annotation as "gold standard" (i.e. *relevant*)

Method	Variant	N	pr%	rc%	F%	acc%
FlatCat	–	–	79.18	89.48	84.01	75.27
spanCRF	–	1	40.33	9.57	15.47	24.13
spanCRF	–	2	77.35	71.80	74.47	55.04
spanCRF	–	3	88.43	87.52	87.97	74.49
spanCRF	–	4	92.83	91.33	92.08	82.57
spanCRF	–	5	93.56	92.29	92.92	84.45
Dsolve	+types	1	36.36	0.02	0.04	22.84
Dsolve	+types	2	79.45	68.32	73.47	53.16
Dsolve	+types	3	89.36	86.64	87.98	74.35
Dsolve	+types	4	93.49	90.81	92.13	82.55
Dsolve	+types	5	94.46	91.63	93.02	84.36
Dsolve	−types	1	56.34	0.72	1.42	23.03
Dsolve	−types	2	77.53	69.61	73.36	52.94
Dsolve	−types	3	88.81	86.58	87.68	73.70
Dsolve	−types	4	92.93	90.78	91.85	81.92
Dsolve	−types	5	93.89	91.73	92.80	83.98

- CRF-based approaches outperform FlatCat
- Performance increases with context size ("lexicalization")
- Dsolve[+types] with higher F-score than Dsolve[−types]

Boundary detection: results



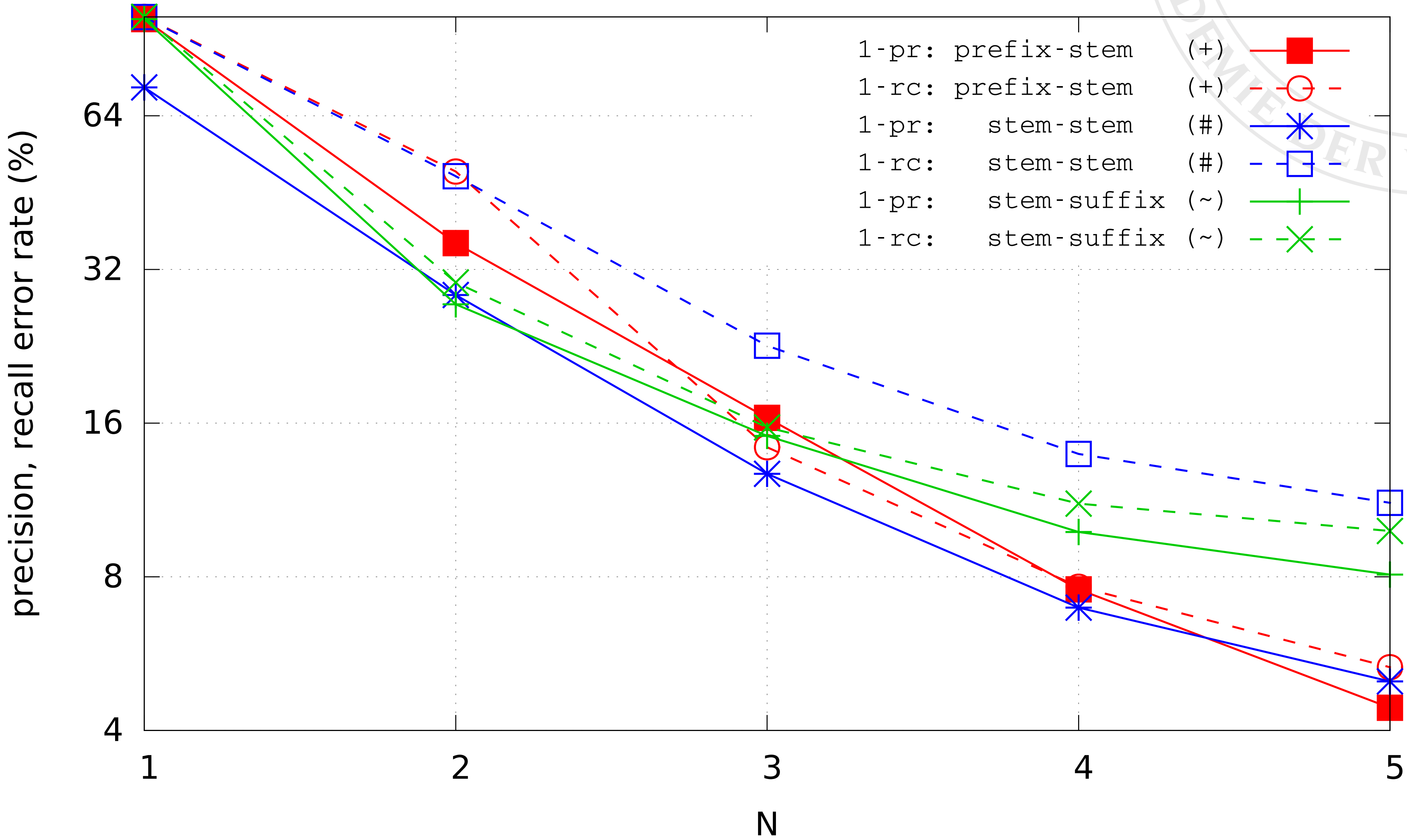
Experiments: boundary classification

Detailed results for `Dsolve` boundary classification by boundary type

	Prefix-Stem (+)			Stem-Stem (#)			Stem-Suffix (∼)		
N	pr%	rc%	F%	pr%	rc%	F%	pr%	rc%	F%
1	–	0.00	–	27.27	0.05	0.10	–	0.00	–
2	63.97	50.25	56.28	71.47	51.27	59.71	72.65	69.83	71.21
3	83.62	85.65	84.63	87.27	77.31	81.99	84.89	84.31	84.60
4	92.44	92.35	92.39	93.04	86.07	89.42	90.21	88.87	89.54
5	95.57	94.68	95.12	95.01	88.83	91.81	91.92	90.16	91.03

- Highest F-score for detection of prefix boundaries (closed set of affixes)
- Suffix boundary detection suffers from high ambiguity of ‘*e*’
  - e.g. *Flieg∼e* (“fly”) vs. *Löwe* (“lion”)
- Precision-oriented compound detection (again an indication for lexicalization)

Boundary classification: results



Summary & Outlook

What We Did (instead of summer holidays)

- CRF-based, supervised approach to morphological segmentation
- Classification of morph boundaries $\rightsquigarrow$ performance increase
- Training materials freely available

What Now?

- Investigate influence of larger N & training corpus size
- Classification of morphs
- Morph-based classifier (vs. character-based variant presented here)
- Use as post-processor for a finite-state morphology
  - e.g. SMOR: good compound detection but many lexicalized affixes

The End

Thank you for listening!