Latent-Variable Modeling of String Transductions with Finite-State Methods

Markus Dreyer, Jason Smith, Jason Eisner

EMNLP 2008

Daichi Mochihashi
The Institute of Statistical Mathematics

SNLP6
September 5, 2014

Motivation

- (Statistical) string processing should be an important part of NLP; but often neglected
- String transducer: map a string $s \mapsto t$ probabilistically.

Example

- Verb inflections

13SIA. lieb**te**, pick**te**, red**ete**, r**ieb**, tr**ieb**, zu**zog**
13SKE. lieb**e**, pick**e**, red**e**, r**eibe**, tr**eibe**, zu**ziehe**
2PIE. lieb**t**, pick**t**, red**et**, reib**t**, treib**t**, zuzieh**t**
13PKE.lieb**en**, pick**en**, red**en**, reib**en**, treib**en**, zuzieh**en**
2PKE. abbreche**t**, entgegentrete**t**, zuziehe**t**
z. ab**zu**brechen, entgegen**zu**treten, zu**zu**ziehen
rP. redet, r**eibt**, tr**eibt**, verb**i**nde**t**, überfischt
pA.**ge**redet, **ge**r**ieben**, **ge**tr**ieben**, verb**u**nde**n**, überfischt

- Noisy writing, abbreviation, jargons
  - William → Bill, 東京レーヨン → 東レ, for you → 4u

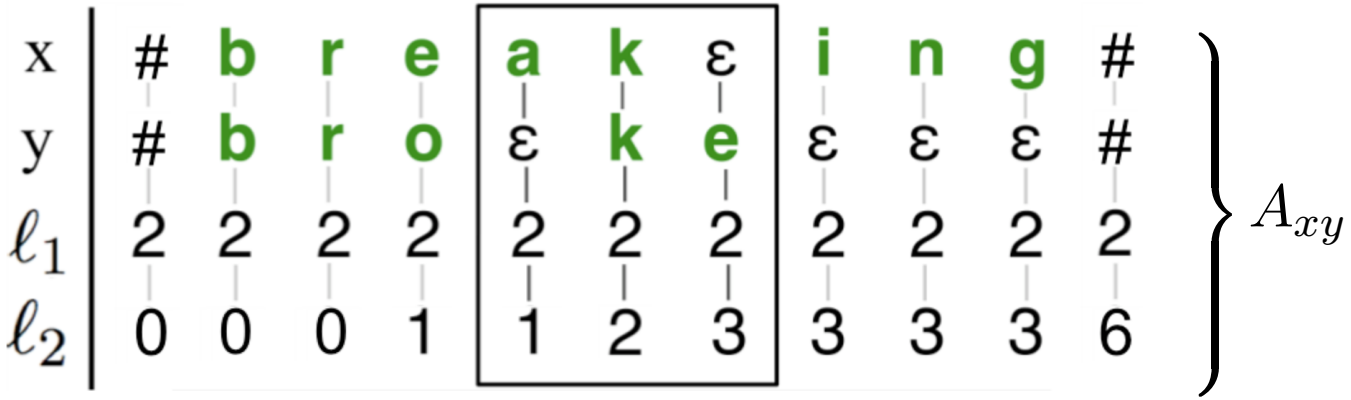
Note

- We still use very heuristic stemmer (like Porter stemmer) for stemming
- Should be replaced by ingenious statistical methods!

Task in this paper

- Inflectional morphology
  - Generate inflected form of unknown verb
  - eg. redet : geredet = tribt : ??? (answer: getrieben)
- Lemmatization
  - Generate a lemma of unknown verb
  - eg. amavissemus (we should have loved) $\rightarrow$ amare (love)

Method



- Consider alignment between "breaking" and "broke"
  - Marginalize over all possible alignments in the end
- Extend the alignment with *latent* states

  ℓ_1: type of the string pair (1-2 in this paper)
  ℓ_2: regions within the string (0-6 in this paper)

Model

- Log-linear random field

$$p(y|x;\theta) = \frac{\sum_{A \in A_{xy}} \exp \sum_i \theta_i f_i(A)}{\sum_y \sum_{A \in A_{xy}} \exp \sum_i \theta_i f_i(A)} \tag{1}$$

- Maximize data likelihood:

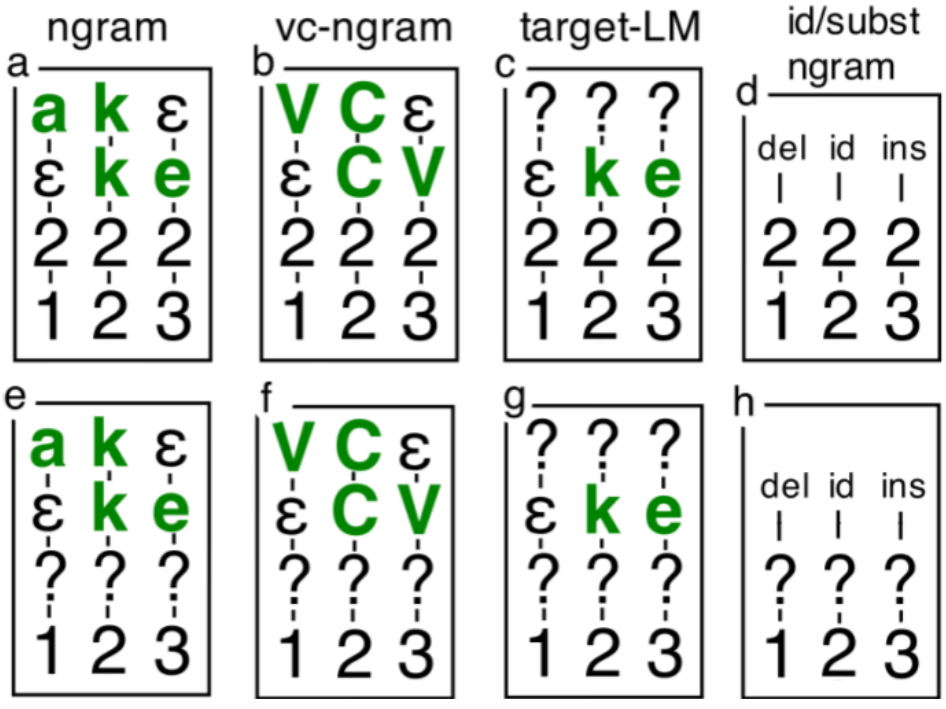
$$p(\mathcal{D}, \theta) = \sum_{(x,y) \in \mathcal{D}} \log p(y|x;\theta) + \frac{|\theta|^2}{2\sigma^2} \tag{2}$$

- What are the *features* $f_i(A)$ on A?

Learning

- WFST implicitly enumerates all alignments/latent type/string regions to yield gradients ("Expectation semiring", Eisner 2002, ACL)
- L-BFGS to optimize

Features



- Features will fire on
  - n-gram surface / vowel-consonant / language model likelihood / edit sequence
- Features will fire using another FST on feature templates

German inflection task

13SIA.	lieb**te**, pick**te**, red**ete**, r**ieb**, tr**ieb**, zu**zog**
13SKE.	lieb**e**, pick**e**, red**e**, r**eibe**, tr**eibe**, zu**ziehe**
2PIE.	lieb**t**, pick**t**, red**et**, reib**t**, treib**t**, zuzieh**t**
13PKE.	lieb**en**, pick**en**, red**en**, reib**en**, treib**en**, zuzieh**en**
2PKE.	abbreche**t**, entgegentrete**t**, zuziehe**t**
z.	ab**zu**brechen, entgegen**zu**treten, zu**zu**ziehen
rP.	redet, r**eibt**, tr**eibt**, verb**i**nde**t**, überfischt
pA.	**ge**redet, **ge**r**ieben**, **ge**tr**ieben**, verb**u**nde**n**, überfischt

- CELEX morphology dataset, 500 pairs for training, 1000 pairs to test
- 13SIA = 1st/3rd singular, ind. past, 13SKE = 1st/3rd singular subjunct. present, 2PIE = 2nd plural ind. present, z = infinitive, rP = imperative plural, pA = past participle

German inflection task: Result

	Features							Task			
	ng	vc	tlm	tlm-coll	id	lat.cl.	lat.reg.	13SIA	2PIE	2PKE	rP
ngrams	x							82.3 (.23)	88.6 (.11)	74.1 (.52)	70.1 (.66)
ngrams+x	x				x			82.8 (.21)	88.9 (.11)	74.3 (.52)	70.0 (.68)
	x		x					82.0 (.23)	88.7 (.11)	74.8 (.50)	69.8 (.67)
	x		x		x			82.5 (.22)	88.6 (.11)	74.9 (.50)	70.0 (.67)
	x		x	x				81.2 (.24)	88.7 (.11)	74.5 (.50)	68.6 (.69)
	x		x	x	x			82.5 (.22)	88.8 (.11)	74.5 (.50)	69.2 (.69)
	x	x						82.4 (.22)	88.9 (.11)	74.8 (.51)	69.9 (.68)
	x	x			x			83.0 (.21)	88.9 (.11)	74.9 (.50)	70.3 (.67)
	x	x	x					82.2 (.22)	88.8 (.11)	74.8 (.50)	70.0 (.67)
	x	x	x		x			82.9 (.21)	88.6 (.11)	75.2 (.50)	69.7 (.68)
	x	x	x	x				81.9 (.23)	88.6 (.11)	74.4 (.51)	69.1 (.68)
	x	x	x	x	x			82.8 (.21)	88.7 (.11)	74.7 (.50)	69.9 (.67)
ngrams+x +latent	x	x	x	x	x	x		84.8 (.19)	**93.6 (.06)**	75.7 (.48)	81.8 (.43)
	x	x	x	x	x		x	**87.4 (.16)**	**93.8 (.06)**	**88.0 (.28)**	83.7 (.42)
	x	x	x	x	x	x	x	**87.5 (.16)**	93.4 (.07)	**87.4 (.28)**	**84.9 (.39)**
Moses3								73.9 (.40)	92.0 (.09)	67.1 (.70)	67.6 (.77)
Moses9								85.0 (.21)	**94.0 (.06)**	82.3 (.31)	70.8 (.67)
Moses15								85.3 (.21)	**94.0 (.06)**	82.8 (.30)	70.8 (.67)

- Latent class and regions helped much for high performance!
  - Moses3 : same window as the proposed model
  - Moses9,15 : more information than the proposed model

Lemmatization task

	Without rootlist (generation)						With rootlist (selection)					
	Wicentowski (2002)			This paper			Wicentowski (2002)			This paper		
Lang.	Base	Af.	WFA.	n	n+x	n+x+l	Base	Af.	WFA.	n	n+x	n+x+l
Basque	85.3	81.2	80.1	91.0 (.20)	91.1 (.20)	**93.6 (.14)**	94.5	94.0	**95.0**	90.9 (.29)	90.8 (.31)	90.9 (.30)
English	91.0	94.7	93.1	92.4 (.09)	93.4 (.08)	**96.9 (.05)**	98.3	98.6	98.6	**98.7 (.04)**	**98.7(.04)**	**98.7(.04)**
Irish	43.3	-	70.8	96.8 (.07)	97.0 (.06)	**97.8 (.04)**	43.9	-	89.1	**99.6 (.02)**	**99.6 (.02)**	**99.5 (.03)**
Tagalog	0.3	80.3	81.7	80.5 (.32)	83.0 (.29)	**88.6 (.19)**	0.8	91.8	96.0	97.0 (.07)	97.2 (.07)	**97.7 (.05)**

Table 3: Exact-match accuracy and average edit distance (the latter in parentheses) on the 8 lemmatization tasks (2 tasks $\times$ 4 languages). The numbers from Wicentowski (2002) are for his Base, Affix and WFAffix models. The numbers for our models are for the feature sets *ngrams*, *ngrams+x*, *ngrams+x+latent*. The best result per task is in **bold** (as are statistically indistinguishable results when we can do the comparison, i.e., for our own models). Corpus sizes: Basque 5,842, English 4,915, Irish 1,376, Tagalog 9,479.

- We can compute $>$ 90% of the lemma of unknown verbs from data!

What are learned

- Latent string class (1-2) delineated *regular* and *irregular* conjugations
- Latent string regions (0-6) corresponded to:
  - 0 $\rightarrow$ different prefixes (eg. *entgegen* in *entgegenzutreten*)
  - 1 $\rightarrow$ insertion sequence (eg. (ϵ,ge))
  - 3 $\rightarrow$ vowel change (eg. $i \rightarrow$ e) (93.7%)
  - 5 $\rightarrow$ typical suffixes (eg. (t,en), (et,en), (t,n)) (92.7%)
  - etc..

Conclusion

- Statistical string transducer augmented with latent states
  - Marginalize over all possible alignments
- Latent states: string types, morphological regions
- Only WFST enables learning this complex model
- Applicable for many languages (German, Arabic, Latin, Finnish, $\cdots$) with no simple inflections