A|Character|wise Windowed Approach to|Hebrew Morphological Seg|mentation

AMIR ZELDES
amir.zeldes@georgetown.edu

GEORGETOWN UNIVERSITY

Paper code + datasets:
https://github.com/amir-zeldes/RFTokenizer
Full NLP pipeline for Hebrew:
https://github.com/amir-zeldes/HebPipe

QR scan:

C@GU Corpling@GU

SIGMORPHON 2018, October 31, Brussels

Overview

- **RFTokenizer:** a new trainable segmenter for Morphologically Rich Languages
- Based on character-wise binary classification
- Provides best Hebrew segmentation accuracy to date: (>yap/UDPipe/Shao et al. 18)
  - 98.19% in domain (SOA +≈4% on **SPMRL shared task**, Seddah et al. 2014)
  - 97.63% out of domain (SOA +≈5% on a new Wikipedia dataset, **Wiki5K**)

Segmenting Hebrew

- Like Arabic and similar languages, Hebrew has whitespace-separated **super-tokens** representing stress-bearing phrases, most vowels are not written out:
  - מהבית <m.h.byt> [me.ha.bajit] – from.the.house
  - ושמצאוהו <w.š.mc'w.hw> [ve.še.mtsa'u.hu] – and.that.they-found.him
- Constituent **sub-tokens** are hard to recognize and can be highly ambiguous: (Adler & Ehadad 2006)

> בצלם
> ⟨b.cl.m⟩ be.cil.am - in.shadow.their
> ⟨b.clm⟩ (be./b.a.)celem - in.(a/the).image
> ⟨b.clm⟩ (be./b.a.)calam in.(a/the).photographer
> ⟨bcl.m⟩ bcal.am - onion.their
> ⟨bclm⟩ becelem - Betzelem (organization)

- Note this example has 7 distinct analyses, but only two positions are candidates for a boundary: after <b> and after <l>!

To alter or not to alter?

- Previous approaches aim at outputting analyzed dictionary forms:
  - -> Token text is altered: *b.byt* [ba.bajit] "in the house" -> *b.h.byt* [be.ha.bajit] can lead to errors: *pwly* "poly-" -> *plh 't 'ni* [pala et ani] "he plucked ACC I"
  - -> unexpressed articles and prepositions inserted: *byth* "her daughter" -> *bt šl hy'* [bat **šel** hi] "daughter **of** she"
- The current approach performs pure character level segmentation

Advantages:

- Input text reconstructible from output
- Tokens align to text positions
- Use standard, token-fed NLP on output
- Useful for:
  - NER (tokens preserve entity text)
  - NMT (segment embeddings)
  - Character/word-level models match

Disadvantages:

- Zero articles moved to morphological features (+Def)
- Need separate morphological analyzer (e.g. Marmot, Müller et al. 2013)
- Lose joint segmentation and disambiguation information for joint inference (cf. previous SOA: **yap**, More & Tsarfaty 2016)

Features and learning approaches

- **Character features:**
  - Use characters in +/-2 character window from boundary candidate
  - Use first/last character of preceding/next super-token
  - Extra feature for each char 'is vowel' , for c ∈ {א,ה,ו,י} (= ', h, w, y)

Red = 'is vowel'

+2 -2

חשבנו שמהפכני הוא

First/last of next

First/last of prev

- **Numerical features:**
  - Corpus frequency ratio **(rfreq)** of current super-token to substring on left and substring on right of window (IsraBlog dataset, Linzen 2009)
  - Lengths of this, previous and next super-tokens
  - Position of current window center

$$rfreq = \frac{f(left) \cdot f(right)}{f(supertoken)}$$

- **Lexicon lookup**
  - MILA lexicon used in previous work (More & Tsarfaty 2016) has very many, complex/hierarchical and sometimes sparse categories
  - We collapse POS>UPOS (Petrov et. 2012), add "CPLX" affix if entry also contains clitics
  - Extend via WikiData named entities
  - Look up range of substrings around window and prev/next word (**Table 1**)
  - Lookup value is a **concatenation** of matched POS tags

location	substring	lexicon response
super token	[šmhpkny]	_
str so far	[šmh]...	ADV|NOUN|VERB
str remaining	..[pkny]	_
str -1 remain	..[hpkny]	_
str -2 remain	.[mhpkny]	ADJ|NOUN|CPLXN
str from -4	[__šmh]....	_
str from -3	[_šmh]....	_
str from -2	[šmh]....	ADV|NOUN|VERB
str from -1	.[mh]....	ADP|ADV
str to +1	..[hp]...	_
str to +2	..[hpk]..	NOUN|VERB
str to +3	..[hpkn].	_
str to +4	..[hpkny]	_
prev string	[xšbnw]	VERB
next string	[hw']	PRON|COP

Table 1: Lexicon lookup features for character 3 in the super-token *š.mhpkny*. Overflow positions (e.g. substring from char -4 for the third character) return '_'.

- **Word embeddings**
  - Only used for NN approaches (300d, from Wikipedia)
- **ML algorithms**
  - Ensembles: RF, GBM, ExtraTrees
  - NNs: DNN, CNN, LSTM classifiers
  - Best in each class: **ExtraTrees RF, DNN** (using scikit-learn and TensorFlow)

Main results

	%perfect	P	R	F
SPMRL				
Baseline	69.65	–	–	–
UDPipe	89.65	93.52	68.82	79.29
yap	94.25	86.33	**96.33**	91.05
RF (ET)	**98.19**	**97.59**	**96.57**	**97.08**
DNN	97.27	95.9	**95.01**	95.45
Wiki5K				
Baseline	67.61	–	–	–
UDPipe	87.39	92.03	64.88	76.11
yap	92.66	85.55	**92.34**	88.81
RF (ET)	**97.63**	**97.41**	**95.31**	**96.35**
DNN	95.72	94.95	**92.22**	93.56

Ablation tests

- Lexicon critical
- WikiData helps, lexicon is still not complete
- Vowel features help to generalize but only a little
- See paper for error analysis

	%perf	P	R	F
SPMRL	98.19	97.59	96.57	97.08
-wikidata	98.01	97.25	96.35	96.80
-vowels	97.99	97.55	95.97	96.75
-letters	97.77	96.98	95.73	96.35
-letr-vowl	97.57	97.56	94.44	95.97
-lexicon	94.79	92.08	91.46	91.77
Wiki5K	97.63	97.41	95.31	96.35
-wikidata	97.33	96.64	95.31	95.97
-vowels	97.51	97.56	94.87	96.19
-letters	97.27	96.89	94.71	95.79
-letr-vowl	96.72	97.17	92.77	94.92
-lexicon	94.72	92.53	91.51	92.01

Discussion

- Why does this outperform joint inference SOA?
  - Parses are sparse, char-wise data is dense
  - Most important syntactic information is preserved, e.g.:
    - *kdy* 'in order to' is SCONJ, 3 chars (k..y) -> next word is to-infinitive
  - Local decisions do not require coherent analyses!
  - Better handling of OOV cases
- Why doesn't DNN beat RF? Needs more data?
  - Need better embeddings (not optimized for this task)
  - Possible issues handling imbalanced problem

References

- Linzen, T. 2009. *Corpus of Blog Postings Collected from the Israblog Website*. TAU, Tech. Report.
- More, A./Tsarfaty, R. 2016. Data-driven morphological analysis and disambiguation for morphologically-rich languages and universal dependencies. *COLING 2016*. Osaka, 337–348.
- Müller, T./Schmid, H./Schütze, H. 2013. Efficient higher-order CRFs for morphological tagging. *EMNLP 2013*. Seattle, WA, 322–332.
- Petrov, S./Das, D./McDonald, R. 2012. A universal part-of-speech tagset. *LREC 2012*. Istanbul, 2089–2096.
- Seddah, D./Kübler, S./Tsarfaty, R. 2014. SPMRL 2014 shared task. *Statistical Parsing of MRLs and Syntactic Analysis of Non-Canonical Languages*. Dublin, 103–109.
- Shao, Y./Hardmeier, C./Nivre, J. 2018. Universal word segmentation: Implementation and interpretation. *TACL* 6, 421–435.
- Straka, M./Hajič, J./Straková, J. 2016. UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. *LREC 2016*. Portorož, 4290–4297.