Sparse Coding of Neural Word Embeddings for Multilingual Sequence Labeling

Gábor Berend

31/07/2017
Vancouver, ACL

Continuous word representations

apple	[1 0 0 0 … 0 0 0 0 0 … 0]	$\longrightarrow$	[3.2	-1.5]
...				
banana	[0 0 0 0 … 1 0 0 0 0 … 0]	$\longrightarrow$	[2.8	-1.6]
...				
door	[0 0 0 0 … 0 0 1 0 0 … 0]	$\longrightarrow$	[-1.1	12.6]
…				
zebra	[0 0 0 0 … 0 0 0 0 0 … 1]	$\longrightarrow$	[0.8	0.5]

Sparse & continuous representations

apple	[3.2 -1.5] →	[0 1.7 0 0 -0.2 0]
...		
banana	[2.8 -1.6] →	[0 1.1 0 0 -0.4 0]
...		
door	[-1.1 12.6] →	[1.7 0 -2.1 0 0 -0.8]
…		
zebra	[0.8 0.5] →	[0 0 1.3 0 -1.2 0]

Creating sparse word representations

- Assuming trained word embeddings w_i $(i=1,\ldots,|V|)$

$$\min_{D \in C, \alpha} \sum_{i=1}^{|V|} \| w_i - D\alpha_i \|_2^2$$

Embedding vector ($\in \mathbb{R}^m$)

Dictionary ($\in \mathbb{R}^{m \times k}$)

Sparse coefficients

Creating sparse word representations

- Assuming trained word embeddings w_i ($i=1,\ldots,|V|$)

$$\min_{D\in C,\alpha} \sum_{i=1}^{|V|} \|w_i - D\alpha_i\|_2^2 + \lambda\|\alpha_i\|_1$$

Embedding vector ($\in\mathbb{R}^m$)

Dictionary ($\in\mathbb{R}^{mxk}$)

Sparse coefficients

Sparsity inducing regularization

Creating sparse word representations

- Assuming trained word embeddings w_i ($i=1,\ldots,|V|$)

$$\min_{D \in C, \alpha} \sum_{i=1}^{|V|} \|w_i - D\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1$$

Convex set of matrices s.t. $\forall \|d_i\| \leq 1$

Embedding vector ($\in \mathbb{R}^m$)

Dictionary ($\in \mathbb{R}^{mxk}$)

Sparse coefficients

Sparsity inducing regularization

Creating sparse word representations

- Assuming trained word embeddings w_i $(i=1,\ldots,|V|)$

$$\min_{D\in C,\alpha} \sum_{i=1}^{|V|} \|w_i - D\alpha_i\|_2^2 + \lambda\|\alpha_i\|_1$$

Convex set of matrices s.t. $\forall \|d_i\| \leq 1$

Embedding vector ($\in \mathbb{R}^m$)

Dictionary ($\in \mathbb{R}^{mxk}$)

Sparse coefficients

Sparsity inducing regularization

  - Similar formulation to *Faruqui et al. (2015)*

“Classical” sequence labeling

- Calculate a set of (surface form) features using feature functions φ_j
  - φ_j could check for capitalization, suffixes, prefixes, neighboring words, etc.

X:	Fruit	flies	like	a	banana	.
Y:	NN	NN	VB	DT	NN	PUNCT
φ:						

"Classical" sequence labeling

- Calculate a set of (surface form) features using feature functions φ_j
  - φ_j could check for capitalization, suffixes, prefixes, neighboring words, etc.

X:	Fruit	flies	like	a	banana	.
Y:	NN	NN	VB	DT	NN	PUNCT
φ:	pre2=Fr suf2=it	pre2=fl suf2=es	pre2=li suf2=ke	pre2=a suf2=a	pre2=ba suf2=na	pre2=. suf2=.

“Classical” sequence labeling

- Calculate a set of (surface form) features using feature functions φ_j
  - φ_j could check for capitalization, suffixes, prefixes, neighboring words, etc.

X:	Fruit	flies	like	a	banana	.
Y:	NN	NN	VB	DT	NN	PUNCT
φ:	pre2=Fr suf2=it …	pre2=fl suf2=es …	pre2=li suf2=ke …	pre2=a suf2=a …	pre2=ba suf2=na …	pre2=. suf2=. ...

Sequence labeling using sparse word representation

- Rely on the sparse coefficients from α
  - $\phi(w_i) = \{ sign(\alpha_i[j])\, j \mid \alpha_i[j] \neq 0 \}$

X:	Fruit	flies	like	a	banana	.
Y:	NN	NN	VB	DT	NN	PUNCT
φ:						

Sequence labeling using sparse word representation

- Rely on the sparse coefficients from α
  - $\phi(w_i) = \{ sign(\alpha_i[j])\, j \mid \alpha_i[j] \neq 0 \}$
    - E.g. $\overrightarrow{Fruit} \approx 1.1 \cdot \overrightarrow{d}_{28} - 0.4 \cdot \overrightarrow{d}_{171}$

X:	Fruit	flies	like	a	banana	.
Y:	NN	NN	VB	DT	NN	PUNCT
φ:						

Sequence labeling using sparse word representation

- Rely on the sparse coefficients from α
  - $\phi(w_i) = \{ sign(\alpha_i[j])\, j \mid \alpha_i[j] \neq 0 \}$
    - E.g. $\overrightarrow{Fruit} \approx 1.1 \cdot \overrightarrow{d}_{28} - 0.4 \cdot \overrightarrow{d}_{171}$

X:	Fruit	flies	like	a	banana	.
Y:	NN	NN	VB	DT	NN	PUNCT
φ:	P28 N171					

Sequence labeling using sparse word representation

- Rely on the sparse coefficients from α
  - $\phi(w_i) = \{ sign(\alpha_i[j])\, j \mid \alpha_i[j] \neq 0 \}$
    - E.g. $\overrightarrow{Fruit} \approx 1.1 \cdot \vec{d}_{28} - 0.4 \cdot \vec{d}_{171}$

X:	Fruit	flies	like	a	banana	.
Y:	NN	NN	VB	DT	NN	PUNCT
φ:	P28 N171	P77 P88 ...	N11 N62 ...	N88 N40 ...	P28 N210 ...	N21 P67 ...

Experimental setup

- Linear chain CRF (CRFsuite implementation)
- Part of Speech tagging
  - 12 languages from the CoNLL-X shared task
  - Google Universal Tag Set (12 tags)

Experimental setup

- Linear chain CRF (CRFsuite implementation)
- Part of Speech tagging
  - 12 languages from the CoNLL-X shared task
  - Google Universal Tag Set (12 tags)
- Hyperparameter settings
  - polyglot/w2v/Glove
  - m=64
  - k=1024
  - Varying λs

$$\min_{D\in C,\alpha} \sum_{i=1}^{|V|} \|w_i - D\alpha_i\|_2^2 + \lambda\|\alpha_i\|_1$$

Embedding vector ($\in\mathbb{R}^m$)

Dictionary ($\in\mathbb{R}^{mxk}$)

Sparse coefficients

Baselines

- Feature rich baseline (FR)
  - Standard feature set borrowed from CRFsuite
    - Previous, next word, word combinations, …
  - 2 variants:
    - Character+word level features (FR_{w+c})
    - Word level features alone (FR_w)

Baselines

- Feature rich baseline (FR)
  - Standard feature set borrowed from CRFsuite
    - Previous, next word, word combinations, …
  - 2 variants:
    - Character+word level features (FR_{w+c})
    - Word level features alone (FR_w)

$FR_{w+c} \supset FR_w$

Baselines

- Feature rich baseline (FR)
  - Standard feature set borrowed from CRFsuite
    - Previous, next word, word combinations, …
  - 2 variants:
    - Character+word level features (FR_{w+c})
    - Word level features alone (FR_{w})
- Brown clustering
  - Derive features from prefixes of Brown cluster IDs

Baselines

- Feature rich baseline (FR)
  - Standard feature set borrowed from CRFsuite
    - Previous, next word, word combinations, …
  - 2 variants:
    - Character+word level features (FR_{w+c})
    - Word level features alone (FR_{w})
- Brown clustering
  - Derive features from prefixes of Brown cluster IDs
- Features from **dense** embeddings
  - $\phi(w_i) = \{ j : \alpha_i[j] \mid \forall j \in 1, \ldots, 64 \}$

Continuous vs. sparse embeddings

- Results averaged over 12 languages

	Dense	Sparse
polyglot	91.17%	94.44%
CBOW	88.30%	93.74%
SG	86.89%	93.63%
Glove	81.53%	91.92%

- Key inspections
  - polyglot > CBOW > SG > Glove

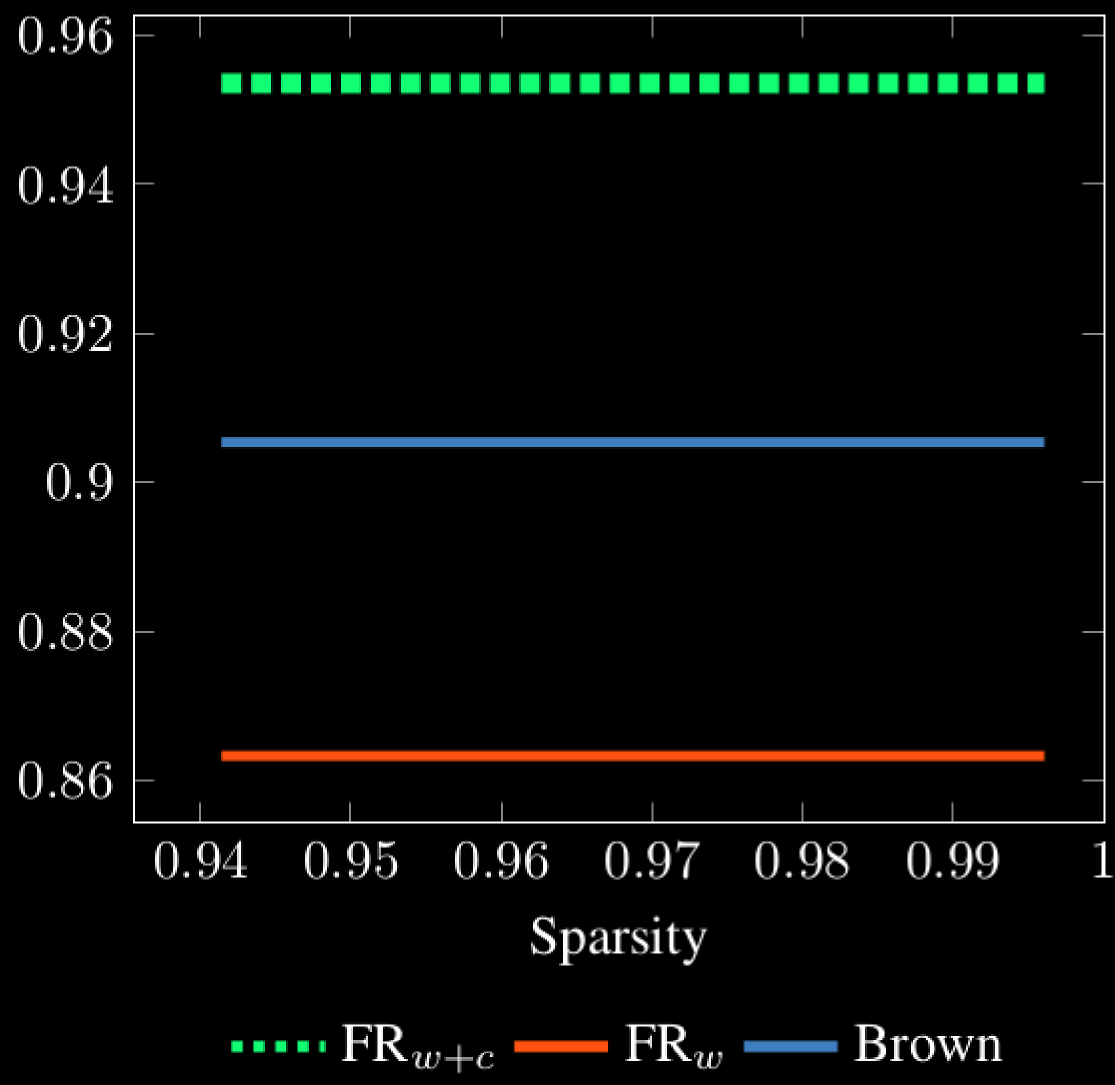
Continuous vs. sparse embeddings

- Results averaged over 12 languages

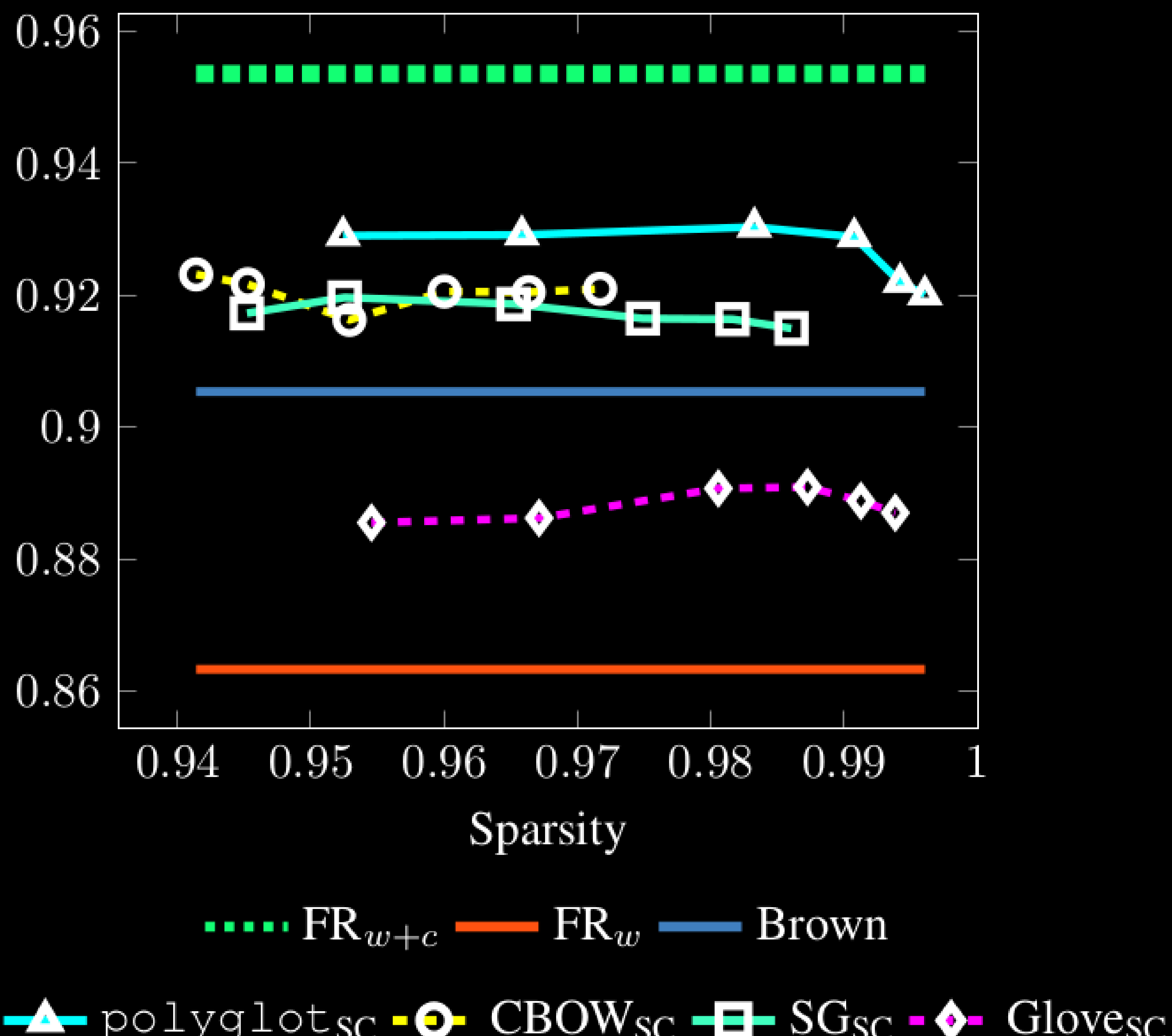
	Dense	Sparse	Improvement
polyglot	91.17%	94.44%	+3.3
CBOW	88.30%	93.74%	+5.4
SG	86.89%	93.63%	+6.7
Glove	81.53%	91.92%	+10.4

- Key inspections
  - polyglot > CBOW > SG > Glove
  - Sparse embeddings >> dense embeddings

Results on Hungarian

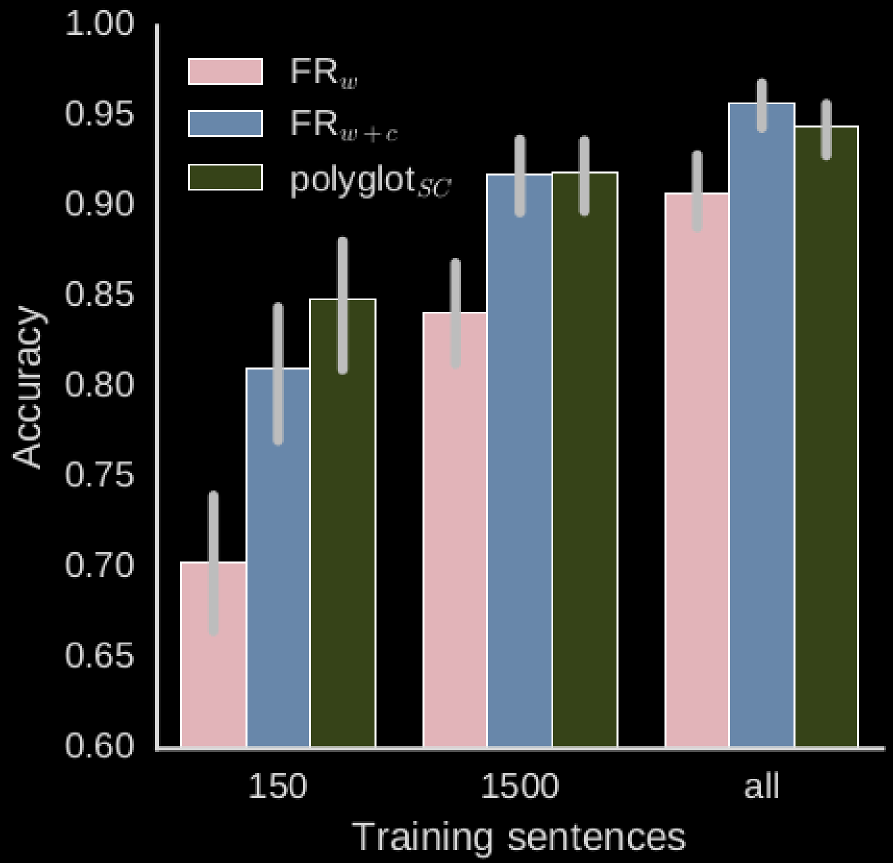


Results on Hungarian



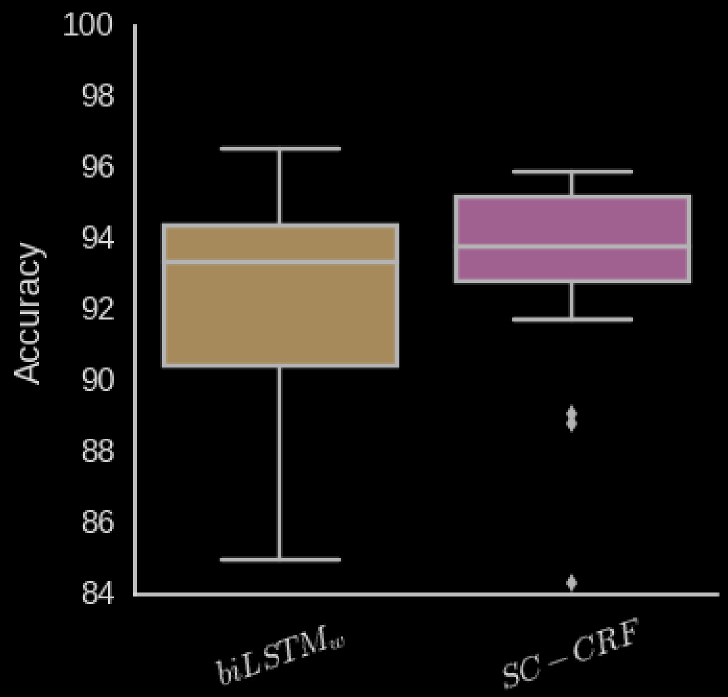
Experiments on generalization

- Training data artificially decreased
  - First 150 and 1500 sentences



Comparison with biLSTMs

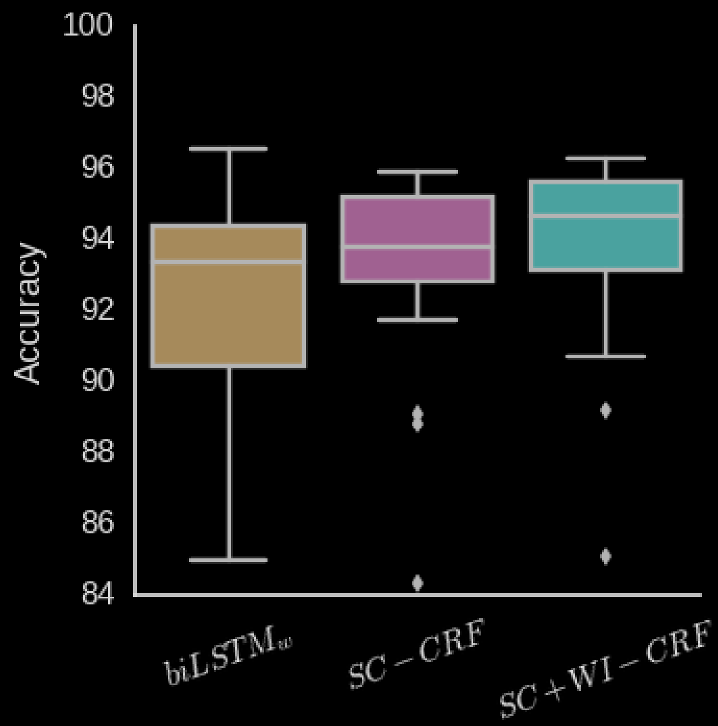
- POS tagging experiments on UD v1.2 treebanks
- Same settings as before (k=1024, λ=0.1)
- biLSTM results from *Plank et al. (2016)*



Method	Avg. accuracy
biLSTM_w	92.40%
SC-CRF	93.15%

Comparison with biLSTMs

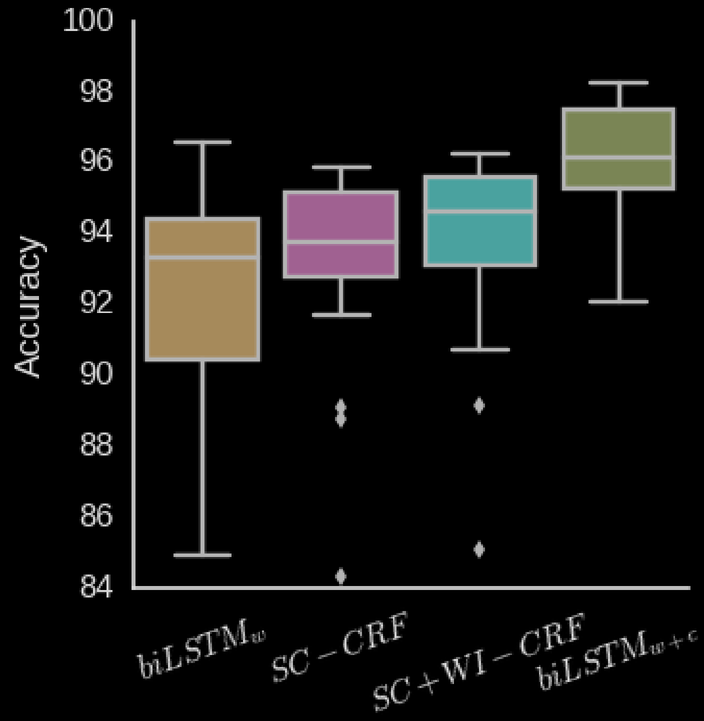
- POS tagging experiments on UD v1.2 treebanks
- Same settings as before (k=1024, λ=0.1)
- biLSTM results from *Plank et al. (2016)*



Method	Avg. accuracy
biLSTM_w	92.40%
SC-CRF	93.15%
SC+WI-CRF	93.73%

Comparison with biLSTMs

- POS tagging experiments on UD v1.2 treebanks
- Same settings as before (k=1024, λ=0.1)
- biLSTM results from *Plank et al. (2016)*



Method	Avg. accuracy
biLSTM_{w}	92.40%
SC-CRF	93.15%
SC+WI-CRF	93.73%
biLSTM_{w+c}	95.99%

Further experiments in the paper

- Quantifying the effects of further hyperparameters
  - Different window sizes for training dense embeddings
- Comparison of different sparse coding techniques
  - E.g. non-negativity constraint
- NER experiments (on 3 languages)

Conclusion

- Simple, yet accurate approach
- Robust across languages and tasks
- Favorable generalization properties
- Competitive results to biLSTMs
- Sparse representations accessible: begab.github.io