Sparse Coding of Neural Word Embeddings for Multilingual Sequence Labeling

Gábor Berend

31/07/2017
Vancouver, ACL
Continuous word representations

apple $[1 \ 0 \ 0 \ 0 \ \ldots \ 0 \ 0 \ 0 \ 0 \ \ldots \ 0]$ ➞ $[3.2 \ -1.5]$

banana $[0 \ 0 \ 0 \ 0 \ \ldots \ 1 \ 0 \ 0 \ 0 \ \ldots \ 0]$ ➞ $[2.8 \ -1.6]$

doors $[0 \ 0 \ 0 \ 0 \ \ldots \ 0 \ 0 \ 1 \ 0 \ 0 \ \ldots \ 0]$ ➞ $[-1.1 \ 12.6]$

zebra $[0 \ 0 \ 0 \ 0 \ \ldots \ 0 \ 0 \ 0 \ 0 \ 0 \ \ldots \ 1]$ ➞ $[0.8 \ 0.5]$
Sparse & continuous representations

<table>
<thead>
<tr>
<th>Object</th>
<th>Vector</th>
<th>Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>apple</td>
<td>[3.2, -1.5]</td>
<td>[0, 0, 1.7, 0, 0, -0.2, 0]</td>
</tr>
<tr>
<td>banana</td>
<td>[2.8, -1.6]</td>
<td>[0, 0, 1.1, 0, 0, -0.4, 0]</td>
</tr>
<tr>
<td>door</td>
<td>[-1.1, 12.6]</td>
<td>[1.7, 0, -2.1, 0, 0, 0, -0.8]</td>
</tr>
<tr>
<td>zebra</td>
<td>[0.8, 0.5]</td>
<td>[0, 0, 1.3, 0, -1.2, 0]</td>
</tr>
</tbody>
</table>
Creating *sparse* word representations

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

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

- Embedding vector ($\in \mathbb{R}^m$)
- Dictionary ($\in \mathbb{R}^{m \times k}$)
- Sparse coefficients
Creating **sparse** word representations

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

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

- Embedding vector ($\in \mathbb{R}^m$)
- Dictionary ($\in \mathbb{R}^{mxk}$)
- Sparse coefficients
- Sparsity inducing regularization
Creating **sparse** word representations

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

\[
\min_{D \in C, \alpha_{i=1}^{V}} \sum_{i=1}^{V} \|w_i - D\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1
\]
Creating **sparse** word representations

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

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

- Similar formulation to Faruqui et al. (2015)

**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**
“Classical” sequence labeling

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

<table>
<thead>
<tr>
<th>X:</th>
<th>Fruit</th>
<th>flies</th>
<th>like</th>
<th>a</th>
<th>banana</th>
<th>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y:</td>
<td>NN</td>
<td>NN</td>
<td>VB</td>
<td>DT</td>
<td>NN</td>
<td>PUNCT</td>
</tr>
</tbody>
</table>

$\varphi$: 

“Classical” sequence labeling

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

X: Fruit  flies  like  a  banana  .
Y: NN  NN  VB  DT  NN  PUNCT
$\varphi$: pre2=Fr  pre2=fl  pre2=li  pre2=a  pre2=ba  pre2=.  suf2=it  suf2=es  suf2=ke  suf2=a  suf2=na  suf2=.
“Classical” sequence labeling

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

<table>
<thead>
<tr>
<th>X:</th>
<th>Fruit</th>
<th>flies</th>
<th>like</th>
<th>a</th>
<th>banana</th>
<th>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y:</td>
<td>NN</td>
<td>NN</td>
<td>VB</td>
<td>DT</td>
<td>NN</td>
<td>PUNCT</td>
</tr>
<tr>
<td>$\varphi$:</td>
<td>pre2=Fr</td>
<td>pre2=fl</td>
<td>pre2=li</td>
<td>pre2=a</td>
<td>pre2=ba</td>
<td>pre2=.</td>
</tr>
<tr>
<td></td>
<td>suf2=it</td>
<td>suf2=es</td>
<td>suf2=ke</td>
<td>suf2=na</td>
<td>suf2=na</td>
<td>suf2=.</td>
</tr>
</tbody>
</table>
Sequence labeling using **sparse** word representation

- Rely on the sparse coefficients from $\alpha$

$\phi(w_i) = \{ \text{sign}(\alpha_i[j]) j \mid \alpha_i[j] \neq 0 \}$

- $X$: Fruit      flies     like       a      banana      .
  - $Y$: NN       NN      VB       DT       NN       PUNCT
  - $\phi$:
Sequence labeling using **sparse** word representation

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

- E.g. $\overrightarrow{\text{Fruit}} \approx 1.1 \cdot \overrightarrow{d_28} - 0.4 \cdot \overrightarrow{d_{171}}$

<table>
<thead>
<tr>
<th>X:</th>
<th>Fruit</th>
<th>flies</th>
<th>like</th>
<th>a</th>
<th>banana</th>
<th>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y:</td>
<td>NN</td>
<td>NN</td>
<td>VB</td>
<td>DT</td>
<td>NN</td>
<td>PUNCT</td>
</tr>
</tbody>
</table>

$\varphi$: 
Sequence labeling using **sparse** word representation

- Rely on the sparse coefficients from $\alpha$
  \[\phi(w_i)=\{\text{sign}(\alpha_i[j])j|\alpha_i[j]\neq0\}\]
- E.g. $\text{Fruit} \approx 1.1 \cdot \vec{d}_{28} - 0.4 \cdot \vec{d}_{171}$

<table>
<thead>
<tr>
<th>X:</th>
<th>Fruit</th>
<th>flies</th>
<th>like</th>
<th>a</th>
<th>banana</th>
<th>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y:</td>
<td>NN</td>
<td>NN</td>
<td>VB</td>
<td>DT</td>
<td>NN</td>
<td>PUNCT</td>
</tr>
<tr>
<td>$\phi$:</td>
<td>P28 N171</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Sequence labeling using **sparse word representation**

- Rely on the sparse coefficients from $\alpha$
  
  $\phi(w_i) = \{ \text{sign} (\alpha_i[j]) j | \alpha_i[j] \neq 0 \}$

- E.g. $\vec{\text{Fruit}} \approx 1.1 \cdot \vec{d}_{28} - 0.4 \cdot \vec{d}_{171}$

<table>
<thead>
<tr>
<th>X:</th>
<th>Fruit</th>
<th>flies</th>
<th>like</th>
<th>a</th>
<th>banana</th>
<th>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y:</td>
<td>NN</td>
<td>NN</td>
<td>VB</td>
<td>DT</td>
<td>NN</td>
<td>PUNCT</td>
</tr>
<tr>
<td>$\phi$:</td>
<td>P28</td>
<td>P77</td>
<td>N11</td>
<td>N88</td>
<td>P28</td>
<td>N21</td>
</tr>
<tr>
<td></td>
<td>N171</td>
<td>P88</td>
<td>N62</td>
<td>N40</td>
<td>N210</td>
<td>P67</td>
</tr>
</tbody>
</table>
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 $\lambda$s

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

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

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

\[
\text{FR}_{w+c} \supset \text{FR}_w
\]
Baselines

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

- **Brown clustering**
  - Derive features from prefixes of Brown cluster IDs
Baselines

- Feature rich baseline (FR)
  - Standard feature set borrowed from CRF-suite
    - 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
  - \(\phi(w_i) = \{ j : \alpha_i[j] | \forall j \in 1, \ldots, 64 \} \)
## Continuous vs. sparse embeddings

- Results averaged over 12 languages

<table>
<thead>
<tr>
<th></th>
<th>Dense</th>
<th>Sparse</th>
</tr>
</thead>
<tbody>
<tr>
<td>polyglot</td>
<td>91.17%</td>
<td>94.44%</td>
</tr>
<tr>
<td>CBOW</td>
<td>88.30%</td>
<td>93.74%</td>
</tr>
<tr>
<td>SG</td>
<td>86.89%</td>
<td>93.63%</td>
</tr>
<tr>
<td>Glove</td>
<td>81.53%</td>
<td>91.92%</td>
</tr>
</tbody>
</table>

- Key inspections
  - polyglot > CBOW > SG > Glove
Continuous vs. sparse embeddings

- Results averaged over 12 languages

<table>
<thead>
<tr>
<th></th>
<th>Dense</th>
<th>Sparse</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>polyglot</td>
<td>91.17%</td>
<td>94.44%</td>
<td>+3.3</td>
</tr>
<tr>
<td>CBOW</td>
<td>88.30%</td>
<td>93.74%</td>
<td>+5.4</td>
</tr>
<tr>
<td>SG</td>
<td>86.89%</td>
<td>93.63%</td>
<td>+6.7</td>
</tr>
<tr>
<td>Glove</td>
<td>81.53%</td>
<td>91.92%</td>
<td>+10.4</td>
</tr>
</tbody>
</table>

- 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, $\lambda=0.1$)
- biLSTM results from *Plank et al. (2016)*

<table>
<thead>
<tr>
<th>Method</th>
<th>Avg. accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>biLSTM$_w$</td>
<td>92.40%</td>
</tr>
<tr>
<td>SC-CRF</td>
<td>93.15%</td>
</tr>
</tbody>
</table>
Comparison with biLSTMs

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Avg. accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>biLSTM&lt;sub&gt;W&lt;/sub&gt;</td>
<td>92.40%</td>
</tr>
<tr>
<td>SC-CRF</td>
<td>93.15%</td>
</tr>
<tr>
<td>SC+WI-CRF</td>
<td>93.73%</td>
</tr>
</tbody>
</table>
Comparison with biLSTMs

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Avg. accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>biLSTM(_w)</td>
<td>92.40%</td>
</tr>
<tr>
<td>SC-CRF</td>
<td>93.15%</td>
</tr>
<tr>
<td>SC+WI-CRF</td>
<td>93.73%</td>
</tr>
<tr>
<td>biLSTM(_{w+c})</td>
<td>95.99%</td>
</tr>
</tbody>
</table>
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