Word representations: A simple and general method for semi-supervised learning

Joseph Turian, Lev Ratinov, Yoshua Bengio

Presenter: Jiachen Tu
Abstract

Word Representations: A simple and general method for semi-supervised learning

• Unsupervised learning to learn word features

  • task-inspecific and model-agnostic approach

• Compared different word representations in a controlled way
Why Useful

Using unsupervised word representations as extra word features

• Improve generalization accuracy for existing supervised NLP systems

• Key questions addressed:
  • Which word features are good for what tasks?
  • Should we prefer certain word features?
  • Can we combine them?
Word Representation

- Vector associated with each word
  - Each dimension’s value corresponds to a word feature
Word Representation

Unsupervised Inducing Approaches

• Clustering
  • One-hot representation over a smaller vocabulary size
• Neural language model
  • Dense real-valued low-dimensional word embeddings
Word Representations

Distributional representations

• Based on a concurrence matrix $F$ of size $W \times C$
  • $W$: vocabulary size; $C$: context size
  • each row $F_w$ — representation of word $w$
  • each column $F_c$ — representation of context $c$
Word Representations

Clustering-based

- Brown clustering \(O(V \cdot K^2)\)
- Hierarchical clustering to maximize mutual information of bigrams

Input: a corpus of words

Output1: a partition of words into \(V\) clusters

Output2: a hierarchical word clustering

<table>
<thead>
<tr>
<th>word</th>
<th>representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>lawyer</td>
<td>1000001101000</td>
</tr>
<tr>
<td>newspaperman</td>
<td>100000110100100</td>
</tr>
<tr>
<td>stewardess</td>
<td>100000110100101</td>
</tr>
<tr>
<td>toxicologist</td>
<td>100000110100111</td>
</tr>
<tr>
<td>slang</td>
<td>100000110101010</td>
</tr>
<tr>
<td>babysitter</td>
<td>100000110101100</td>
</tr>
<tr>
<td>conspirator</td>
<td>100000110101101</td>
</tr>
<tr>
<td>womanizer</td>
<td>100000110111011</td>
</tr>
<tr>
<td>mailman</td>
<td>100000110110111</td>
</tr>
<tr>
<td>salesman</td>
<td>100000110110000</td>
</tr>
<tr>
<td>bookkeeper</td>
<td>1000001101100010</td>
</tr>
<tr>
<td>troubleshooter</td>
<td>1000001101100110</td>
</tr>
<tr>
<td>bouncer</td>
<td>1000001101100111</td>
</tr>
<tr>
<td>technician</td>
<td>100000110110100</td>
</tr>
<tr>
<td>janitor</td>
<td>100000110110101</td>
</tr>
<tr>
<td>saleswoman</td>
<td>1000001101100110</td>
</tr>
</tbody>
</table>
Word Representations

Clustering-based

• Other works
  • K-means-like non-hierarchical clustering for phrases
  • HMMs
  • …
Distributed representations
(word embeddings)

• Dense, low-dimensional, and real-valued
• Each dimension represents a latent feature of the word
• Typically induced using neural language models
Distributed representations

Collobert and Weston (2008) embeddings

- Neural language model (n-gram)  \( e \) is the lookup table and \( \oplus \) is concatenation

\[
x = (w_1, \ldots, w_n) \quad \rightarrow \quad e(w_1) \oplus \ldots \oplus e(w_n) \quad \rightarrow \quad s(x)
\]

\[
\tilde{x} = (w_1, \ldots, w_{n-q}, \tilde{w}_n), \text{ where } \tilde{w}_n \neq w_n \quad \rightarrow \quad s(\tilde{x})
\]

\[
L(x) = \max(0, 1 - s(x) + s(\tilde{x}))
\]
Distributed representations
Collobert and Weston (2008) embeddings

- Implementation
  - Corrupt the last word of each n-gram
  - Separate learning rate for the embeddings and for the neural network weights
    - Embeddings have a learning rate generally 1000-32000 times higher
  - Used moving average of the training loss on training examples before the weight update to save computing resources
Distributed representations

HLBL embeddings

- Hierarchical log-bilinear model
- Given an n-gram, the model concatenates the embeddings of the n-1 first words, and learns a linear model to predict the embedding of the last word
Supervised evaluation tasks

Chunking

- Syntactic sequence labeling task
  - identify parts of speech and short phrases present in a given sentence
- Baseline chunker
  - Linear CRF chunker (CRFsuite)
Supervised evaluation tasks

Chunking

- Data
  - The Penn Treebank [8936 training sentences]
    - Dev set: 1000 randomly sampled sentences
    - Model trained on the rest 7936 sentences and tuned to maximize the dev F1
- Model retrained using the hyperparameters on the full training set and evaluated on test
- Hyperparameters
  - L2-regularization sigma (2 or 3.2)
  - Scaling hyperparameter
Supervised evaluation tasks

Named entity recognition (NER)

- Sequence prediction problem
- Regularized averaged perceptron model
  - Greedy inference
  - BILOU text chunk representation
Supervised evaluation tasks

Named entity recognition (NER)

- Baseline experiments using the implementation from Ratinov and Roth (2009)
  - Removed gazetteers and non-local features
- Training stopped after the accuracy on the dev set did not improve for 10 epochs (~50-80 epochs total)
- Final model selected from the epoch that performed best on the dev set
Supervised evaluation tasks
Named entity recognition (NER)

Data

- Standard evaluation benchmark -- CoNLL03 (from Reuters newswire)
  - Training set: 204k words (14k sentences, 946 documents)
  - Test set: 46K words (3.5K sentences, 231 documents)
  - Dev set: 51K words (3.3K sentences, 216 documents)
- Out-of-domain (OOD) dataset -- MUC7
  - Post-processing steps to adapt the different annotation standard
Unlabeled Data

- Used for inducing word representations
- Data: RCV1 corpus (one year of Reuters English newswire)
- Preprocessing / cleaning
  - Removed all sentences that are less than 90% lowercase a-z
  - Assumed whitespace is not counted
- ~37 million words in 1.3 million sentences with 269K word types (vocabulary size)
Experiments and Results

Details of inducing word representations

- The Brown clusters [~3 days]
- The Collobert and Weston (C&W) embeddings [a few weeks / 50 epochs]
- The HLBL embeddings [7 days / 100 epochs]
Experiments and Results

Scaling of Word Embeddings

- Scale the word embeddings by a hyperparameter to control their standard deviation to ensure a bounded range

\[ E \leftarrow \sigma \cdot \frac{E}{\text{stddev}(E)} \]
Experiments and Results

Scaling of Word Embeddings

- All curves had similar shapes and optima on both tasks
- Choose scale factor s.t. The embeddings have a std of 0.1

\[ E \leftarrow \sigma \cdot \frac{E}{\text{stddev}(E)} \]
Experiments and Results

Capacity of Word Representations

- Capacity controls
  - Number of Brown clusters
  - Number of dimensions of the word embeddings
Experiments and Results

Capacity of Word Representations

- More Brown clusters are better
- Higher-dimensional word embeddings wouldn’t give higher accuracy
  - Optimal capacity of the word embeddings is task-specific
Experiments and Results

Chunking F1 results

- Combining representations leads to small increases in test F1

<table>
<thead>
<tr>
<th>System</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>94.16</td>
<td>93.79</td>
</tr>
<tr>
<td>HLBL, 50-dim</td>
<td>94.63</td>
<td>94.00</td>
</tr>
<tr>
<td>C&amp;W, 50-dim</td>
<td>94.66</td>
<td>94.10</td>
</tr>
<tr>
<td>Brown, 3200 clusters</td>
<td>94.67</td>
<td>94.11</td>
</tr>
<tr>
<td>Brown+HLBL, 37M</td>
<td>94.62</td>
<td>94.13</td>
</tr>
<tr>
<td>C&amp;W+HLBL, 37M</td>
<td>94.68</td>
<td>94.25</td>
</tr>
<tr>
<td>Brown+C&amp;W+HLBL, 37M</td>
<td>94.72</td>
<td>94.15</td>
</tr>
<tr>
<td>Brown+C&amp;W, 37M</td>
<td>94.76</td>
<td>94.35</td>
</tr>
<tr>
<td>Ando and Zhang (2005), 15M</td>
<td>-</td>
<td>94.39</td>
</tr>
<tr>
<td>Suzuki and Isozaki (2008), 15M</td>
<td>-</td>
<td>94.67</td>
</tr>
<tr>
<td>Suzuki andIsozaki (2008), 1B</td>
<td>-</td>
<td><strong>95.15</strong></td>
</tr>
</tbody>
</table>
Experiments and Results

NER F1 results

- Combining different word representations on NER seems gives larger improvements on test F1
- Brown clusters are superior
  - Better representation for rare words

<table>
<thead>
<tr>
<th>System</th>
<th>Dev</th>
<th>Test</th>
<th>MUC7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>90.03</td>
<td>84.39</td>
<td>67.48</td>
</tr>
<tr>
<td>Baseline+Nonlocal</td>
<td>91.91</td>
<td>86.52</td>
<td>71.80</td>
</tr>
<tr>
<td>HLBL 100-dim</td>
<td>92.00</td>
<td>88.13</td>
<td>75.25</td>
</tr>
<tr>
<td>Gazetteers</td>
<td>92.09</td>
<td>87.36</td>
<td>77.76</td>
</tr>
<tr>
<td>C&amp;W 50-dim</td>
<td>92.27</td>
<td>87.93</td>
<td>75.74</td>
</tr>
<tr>
<td>Brown, 1000 clusters</td>
<td>92.32</td>
<td>88.52</td>
<td>78.84</td>
</tr>
<tr>
<td>C&amp;W 200-dim</td>
<td>92.46</td>
<td>87.96</td>
<td>75.51</td>
</tr>
<tr>
<td>C&amp;W+HLBL</td>
<td>92.52</td>
<td>88.56</td>
<td>78.64</td>
</tr>
<tr>
<td>Brown+HLBL</td>
<td>92.56</td>
<td>88.93</td>
<td>77.85</td>
</tr>
<tr>
<td>Brown+C&amp;W</td>
<td>92.79</td>
<td>89.31</td>
<td>80.13</td>
</tr>
<tr>
<td>HLBL+Gaz</td>
<td>92.91</td>
<td>89.35</td>
<td>79.29</td>
</tr>
<tr>
<td>C&amp;W+Gaz</td>
<td>92.98</td>
<td>88.88</td>
<td>81.44</td>
</tr>
<tr>
<td>Brown+Gaz</td>
<td><strong>93.25</strong></td>
<td><strong>89.41</strong></td>
<td><strong>82.71</strong></td>
</tr>
<tr>
<td>Lin and Wu (2009), 3.4B</td>
<td>-</td>
<td>88.44</td>
<td>-</td>
</tr>
<tr>
<td>Ando and Zhang (2005), 27M</td>
<td>93.15</td>
<td>89.31</td>
<td>-</td>
</tr>
<tr>
<td>Suzuki and Isozaki (2008), 37M</td>
<td>93.66</td>
<td>89.36</td>
<td>-</td>
</tr>
<tr>
<td>Suzuki and Isozaki (2008), 1B</td>
<td><strong>94.48</strong></td>
<td>89.92</td>
<td>-</td>
</tr>
<tr>
<td>All (Brown+C&amp;W+HLBL+Gaz), 37M</td>
<td>93.17</td>
<td>90.04</td>
<td>82.50</td>
</tr>
<tr>
<td>All+Nonlocal, 37M</td>
<td>93.95</td>
<td>90.36</td>
<td>84.15</td>
</tr>
<tr>
<td>Lin and Wu (2009), 700B</td>
<td>-</td>
<td><strong>90.90</strong></td>
<td>-</td>
</tr>
</tbody>
</table>
Final results

- accuracy can be increased further by combining the features from different types of word representations
- if only one word representation is to be used, **Brown clusters** have the highest accuracy
Final results

Per-token errors

- Chunking
  - Both incur almost identical # of errors & errors are concentrated around the more common words
  - Non-rare words have good representations
- NER
  - Brown clusters incur fewer errors for rare words
Conclusions

- Brown clusters and word embeddings both can improve the accuracy of a near-state-of-the-art supervised NLP system.
- Combining different word representations can improve accuracy further.
- Brown clustering induces better representation for rare words than C&W embeddings.
  - Brown makes a single hard clustering decision, whereas the embedding for a rare word is close to its initial value since it hasn’t received many training updates.
- Default method for scaling parameter:
  - Choose scale factor s.t. The embeddings have a std of 0.1.
Questions to investigate further:

- For NER task, why does the word representations brought larger gains on the out-of-domain data than on the in-domain data?
- Comparison to other task-specific semi-supervised methods
- Novel methods to improve the current word representations
Thank you!!