Word representations:

A simple and general method for semi-supervised learning

Joseph Turian, Lev Ratinov, Yoshua Bengjo

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*C
  - W: vocabulary size; C: context size
  - each row Fw — representation of word w
  - each column Fc — representation of context c

Word Representations

Clustering-based

- **Brown clustering O(V·K^2)**
  - Hierarchical clustering to maximize ... ns
  - Input: a corpus of words
  - Output1: a partition of words into ...
  - Output2: a hierarchical word cluste...

lawyer	1000001101000
newspaperman	100000110100100
stewardess	100000110100101
toxicologist	10000011010011
slang	1000001101010
babysitter	100000110101100
conspirator	1000001101011010
womanizer	1000001101011011
mailman	10000011010111
salesman	100000110110000
bookkeeper	1000001101100010
troubleshooter	10000011011000110
bouncer	10000011011000111
technician	1000001101100100
janitor	1000001101100101
saleswoman	1000001101100110

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

$x = (w_1, \ldots, w_n)$ —— $e(w_1) \oplus \ldots \oplus e(w_n)$ —— $s(x)$

$\tilde{x} = (w_1, \ldots, w_{n-q}, \tilde{w_n})$, where $\tilde{w_n} \neq w_n$ —— $s(\tilde{x})$

$$L(x) = \max(0, 1 - s(x) + s(\tilde{x}))$$

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 epchs 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

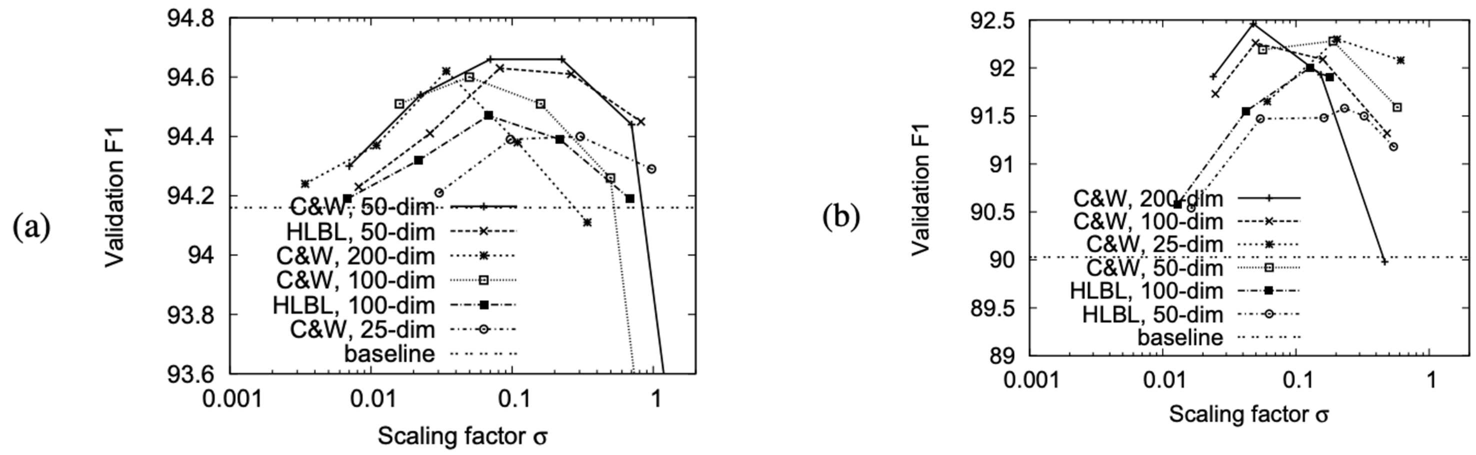
- The Brown clusters [~3 days]
- The Collobert and Weston (C&W) embeddings [a few weeks / 50 epochs]
- The HLBL embeddings [7 days / 100 epochs]

Experiments and Results

Scaling of Word Embeddings

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

$$E \leftarrow \sigma \cdot E / stddev(E)$$

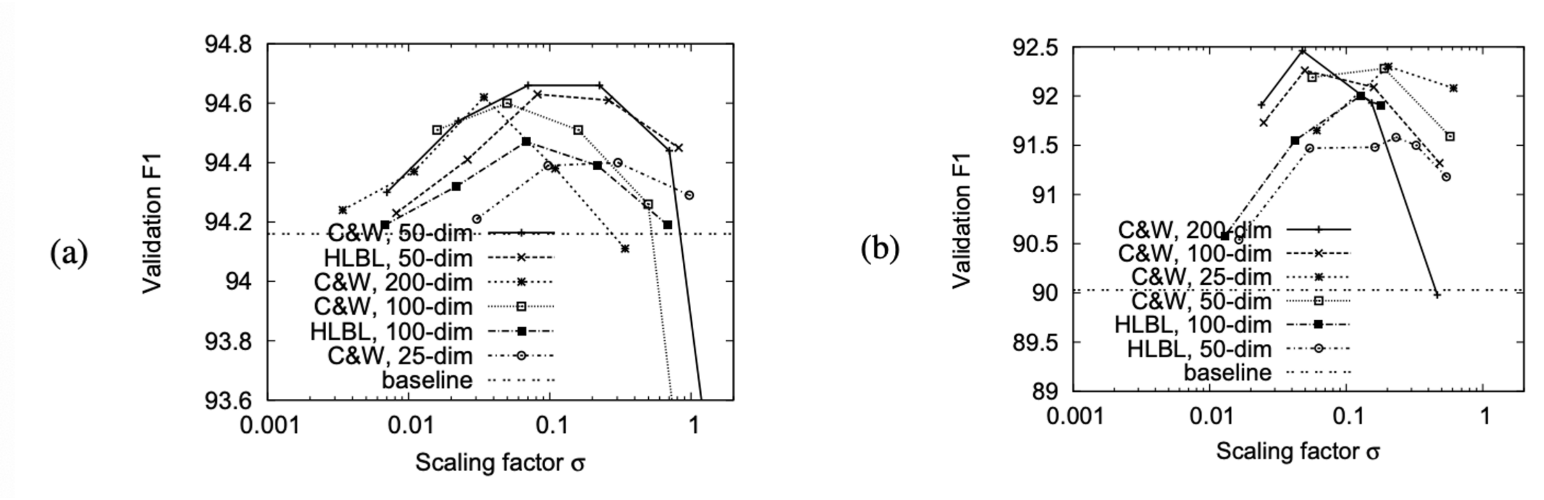


Experiments and Results

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

$$E \leftarrow \sigma \cdot E / stddev(E)$$

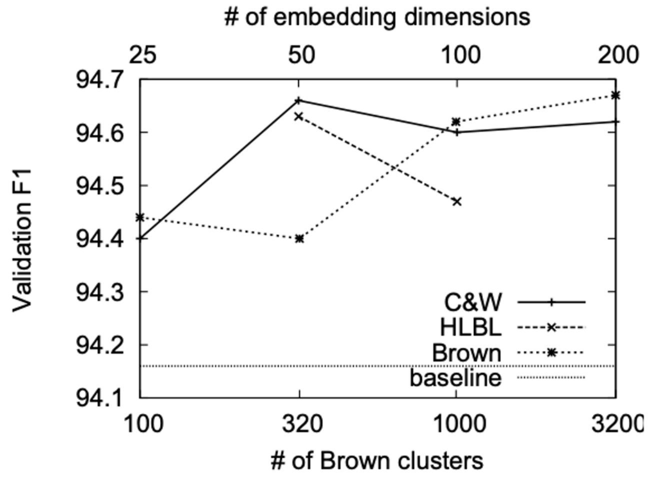


Experiments and Results

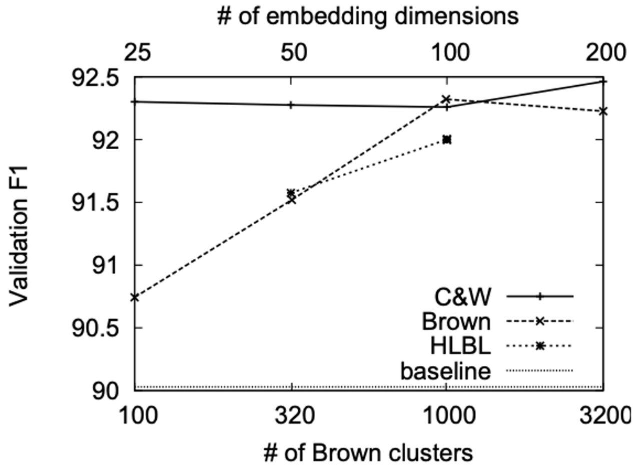
Capacity of Word Representations

- Capacity controls
  - Number of Brown clusters
  - Number of dimensions of the word embeddings

(a)



(b)

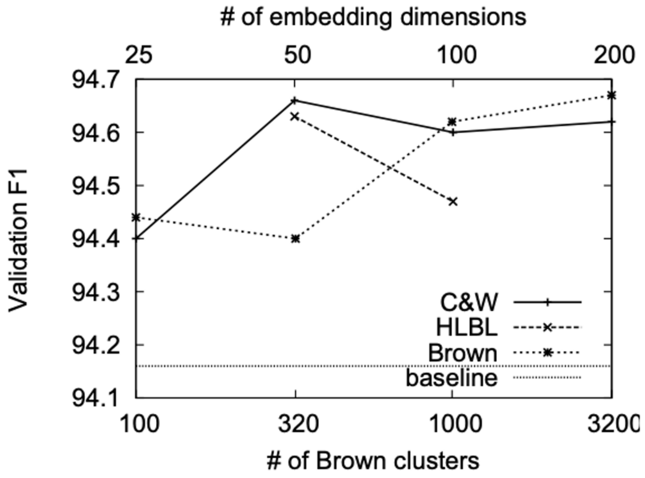


Experiments and Results

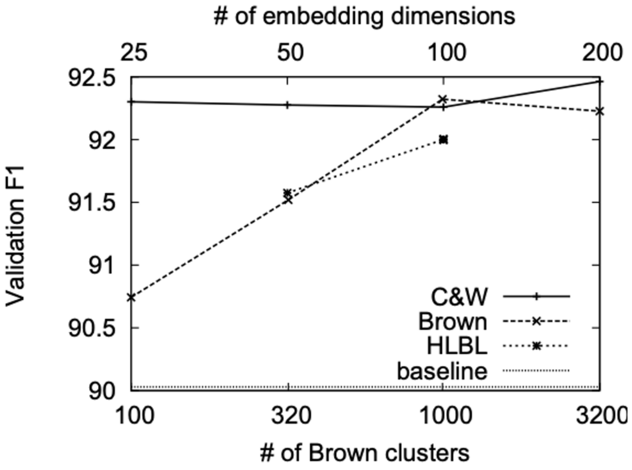
Capacity of Word Representations

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

(a)



(b)



Experiments and Results

Chunking F1 results

- Combining representations leads to small increases in test F1

System	Dev	Test
Baseline	94.16	93.79
HLBL, 50-dim	94.63	94.00
C&W, 50-dim	94.66	94.10
Brown, 3200 clusters	**94.67**	**94.11**
Brown+HLBL, 37M	94.62	94.13
C&W+HLBL, 37M	94.68	94.25
Brown+C&W+HLBL, 37M	94.72	94.15
Brown+C&W, 37M	94.76	94.35
Ando and Zhang (2005), 15M	-	94.39
Suzuki and Isozaki (2008), 15M	-	94.67
Suzuki and Isozaki (2008), 1B	-	**95.15**

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

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

Final results

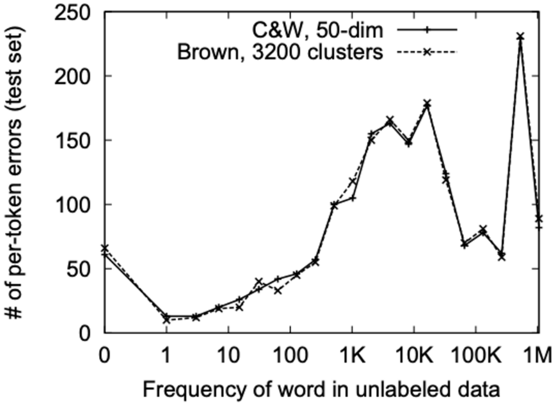
- accuracy can be increased further by combining the features from different types of word representations
- if only one word representation is to be used, **Brown clusters** have the highest accuracy

Final results

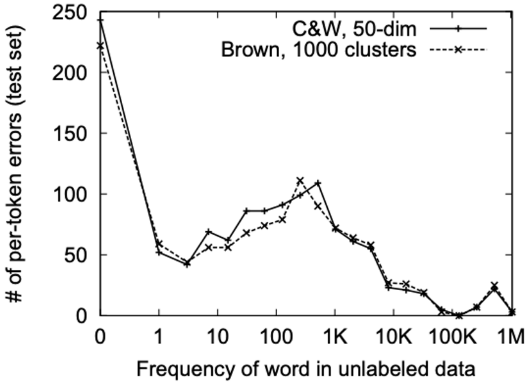
Per-token errors

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

(a)



(b)



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!!

Appendix