Word Representations: A Simple and General Method for Semi-Supervised Learning

Turian, Ratinov, and Bengio (2010)

presented by Sam Thomson
Nov. 6, 2013

Overview

- Compares four different types of vector space word representations
- Looks at how much they help in two extrinsic tasks
  - named entity recognition (NER)
  - chunking

Vector Space Word Representations


seen
up
down
amonq
out
off
called
about
around
told
over
near
within
at
under
In
into
between
through
from
in
But
If
for
of
against
like
with
during
on
by
including
than
unless
until
as
since
before
after
without
while
when
if
because
despite
where
following
whether

The Way of the Ancients

- One-hot
  - W-dimensional vector, where W is the size of your vocabulary
  - one component (the word's) is 1, the rest are 0

Distributional Vectors

- Build up a big matrix or tensor of cooccurrance counts
  - WxC, where W is your vocab, and C is some context that the word appears in
  - Do dimensionality reduction

Distributional Vectors

- Design decisions:
  - Counts
    - raw counts?
    - binary?
    - normalized?
    - tf-idf?

Distributional Vectors

- Design decisions:
  - Context
    - k-word left window? right window?
    - neighbors in a syntax tree?
    - the whole document?

Distributional Vectors

- Design decisions:
  - What kind of dimensionality reduction?
    - matrix factorization?
    - random projection?
  - How many dimensions?

Clusters

- "Brown" clustering
  - hierarchical clustering
  - maximizes mutual info of bigrams
  - can be extended
    - bigrams -> trigrams
    - words -> phrases

Clusters

- For example:
  - http://www.ark.cs.cmu.edu/cdyer/en-600/cluster_viewer.html

Clusters

- Non-hierarchical
  - k-means
  - soft clustering
    - Hidden states of an HMM

Distributed "don't call me distributional" Vectors

- Collobert and Weston (C&W)
  - Predict word based on previous n-1 words
  - Discriminative neural network
    - Create a bunch of synthetic negative data, train with a max-margin objective

Distributed "don't call me distributional" Vectors

- Log-bilinear (HLBL)
  - Predict word based on previous n-1 words
  - Log-bilinear = log-linear with every conjunction of (input feature, output feature) thrown in
  - Optimize w.r.t. features also (treat them as parameters)

Results

Chunking

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**

NER

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**	-

Tune Your Hypers

- Scale embeddings to have std dev = 0.1
- More Brown clusters are always better
- Optimal dimension is task-specific
  - 50 for chunking
  - 200 for NER

Take-aways

- Which representation works best is task-specific
- Ideal would be to optimize for your task, but even these non-ideal versions help
- They capture different info. Throw them all in!