Word Vectors - GloVe

Natural Language Processing

(based on revision of Chris Manning Lectures)

Announcement

- TA announcements (if any)...

Suggested Readings

1. https://web.stanford.edu/~jurafsky/slp3/6.pdf (vector semantics and embeddings)
2. Distributed Representations of Words and Phrases and their Compositionality (negative sampling paper)
3. GloVe: Global Vectors for Word Representation (original GloVe paper)
4. Co-occurrence matrix (hacks to co-occurrence matrix)
5. Improving Distributional Similarity with Lessons Learned from Word Embeddings
6. Evaluation methods for unsupervised word embeddings
7. A Latent Variable Model Approach to PMI-based Word Embeddings
8. Linear Algebraic Structure of Word Senses, with Applications to Polysemy
9. On the Dimensionality of Word Embedding

Negative Sampling

Negative sampling [Mikolov et al., NeuroIPS 2013]

- The normalization term is computationally expensive.

$$P(o|c) = \frac{\exp(\mathbf{u_o}^\top \mathbf{v_c})}{\sum_{w=1}^{V} \exp(\mathbf{u_w}^\top \mathbf{v_c})}$$

- **Idea**: Instead of using all vocabularies, we can just pick some “negative” samples. We just need to make the difference between positive and negative samples to be big.

Distributed Representations of Words and Phrases and their Compositionality, Mikolov et al., 2013, https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf

Negative sampling

- We draw *k* random negative samples
- We maximize the probability of real outside word appears, and minimize the probability that random words appear around the center word

$$\mathbf{J}_{\text{neg-sample}}(\mathbf{v}_c, o, \mathbf{U}) = -\log(\sigma(\mathbf{u}_o^T \mathbf{v}_c)) - \sum_{k=1}^{K} \log(\sigma(-\mathbf{u}_k^T \mathbf{v}_c))$$

- Negative sampling is a widely used technique in the deep learning field, not only NLP

Limitations of Word2vec

- Only look at local words
  - Does not utilize global co-occurrence statistics
  - **Possible solution**: use co-occurrence counts (GloVe)
- Does not work well with out-of-vocabulary
  - **Possible solution**: use character based or sub-words embedding (e.g., FastText)
- Doubtful whether contextual information was fully captured
  - **Possible solution**
    - passed these trained embeddings through some LSTM, and get the resulting encodings as embeddings (ELMo)
    - Provide some different prediction tasks such as fill in the blanks or fill in next sentence, so that the model can capture better context (BERT)

Co-occurrence matrix

Co-occurrence matrix

- 2 options: windows vs. full document

- **Window**: Similar to word2vec, use window around each word -> capture some syntactic and semantic information
- **Document**: similar to Latent Semantic Analysis

For word embeddings, we use window-based

Co-occurrence matrix: Example

- Window length 1 (more common: 5-10)
- Symmetric (irrelevant whether left or right context)
- Example corpus:
  - *"I like deep learning"*
  - *"I like NLP"*
  - *"I enjoy flying"*

counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

Co-occurrence matrix: Problem

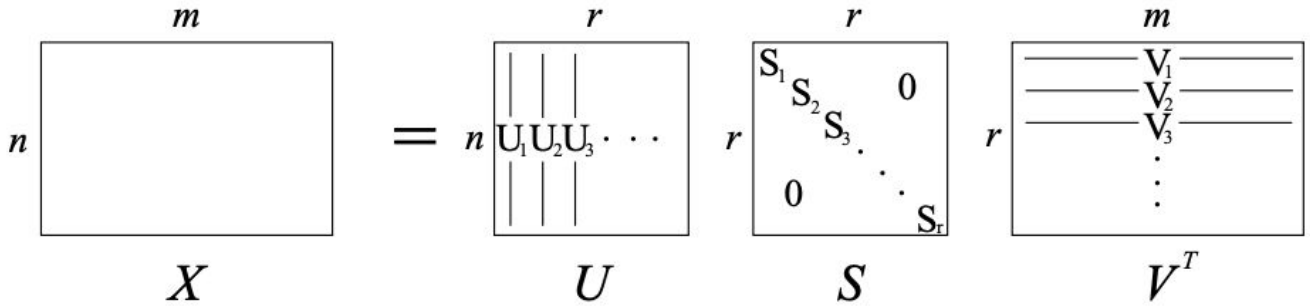
- **Simple count co-occurrence vectors**
  - Vectors increase in size with vocabulary
  - Very high dimensional: require a lot of storage (though sparse)

- **Low dimensional vectors**
  - **Idea**: store “most” of the important information in a fixed, small number of dimensions: a dense vector
  - Usually 25-1000 dimensions, similar to word2vec
  - How to reduce the dimensionality?

Co-occurrence matrix: SVD [Rhode et al. CogSci 2005]

Use SVD! (Basically PCA).

- **Singular Value Decomposition** of co-occurrence matrix X
- Factor *X* into $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$



An Improved Model of Semantic Similarity Based on Lexical Co-Occurrence, Rhode et al. 2005, https://arxiv.org/pdf/1301.3781.pdf

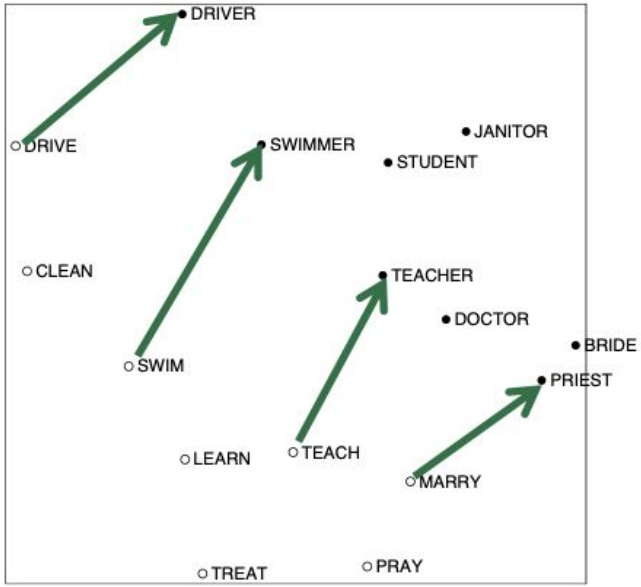
Co-occurrence matrix: Hacks to X [Rhode et al. CogSci 2005]

- Rhode et al. 2005 found that running an SVD on raw counts doesn't work well
- **Scaling the counts** in the cells can help a lot
  - Problem: frequent words (the, he, has) have too much impact
  - Fixes:
    - Log the frequencies
    - min(X, t), with t = 100 for example
    - Ignore these words
    - Ramped windows that count closer words more than further away words
    - Use Pearson correlations instead of counts, then set negative values to 0

An Improved Model of Semantic Similarity Based on Lexical Co-Occurrence, Rhode et al. 2005, https://arxiv.org/pdf/1301.3781.pdf

Co-occurrence matrix: Rhode results

Interesting semantic patterns emerge in the scaled vectors



GloVe

Towards GloVe: Count based vs. Prediction-based

Count-Based

Good:

- Fast-training
- Efficient usage of statistics

Bad:

- Scale badly to frequent words
- Use mainly to capture word similarity

Prediction-Based

Good:

- Transfer to better classification performances
- Capture complex patterns beyond word similarity

Bad

- Inefficient usage of statistics

Idea: Can we combine the best of both worlds? For example, we can take these two vectors and make sure they are as close to one another as possible

GloVe [Pennington et al., EMNLP 2014]

- Pennington et al. (2014) are aware that there are problems with **raw counts**
- They found that **ratios of co-occurrence probabilities** can encode meaning

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

GloVe: Global Vectors for Word Representation, Pennington et al. 2014, https://aclanthology.org/D14-1162.pdf

GloVe

- Statistics are scalar; turning it into a loss function is not a simple task. Through some bits of math, the authors find that the statistics can be expressed into the following (*Chaky walk through the paper*)

$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$

- f is just some capping function for frequent items or rare occurrences
- b_i and b_j are just some bias terms
- Fast training, scalable to large corpus
- Good performance, even with small corpus and small vectors

Read https://towardsdatascience.com/light-on-math-ml-intuitive-guide-to-understanding-glove-embeddings-b13b4f19c010 further if you are interested in easier version of the math derivation

GloVe results

Nearests words to frog

litoria

leptodactylidae

rana

eleutherodactylus

Word Vectors Evaluation

Word vectors evaluation

Intrinsic

- Evaluation on a specific/intermediate subtask (e.g., analogies)
- Fast to compute
- Not clear whether it will improve real task

Extrinsic

- Evaluate on real task (e.g., question answering)
- Can take long time

Word vectors evaluation

Intrinsic

- Evaluate word vectors by how well they capture intuitive **semantic** (meaning) and **syntactic** (grammar) analogy questions.
- Analogy task consists of questions like, "*a* is to *b* as *c* is to ?"
  - The **semantic** questions are typically analogies about people or places, like "*Athens is to Greece as Berlin is to ?*".
  - The **syntactic** questions are typically analogies about verb tenses or forms of adjectives, for example "*dance is to dancing as fly is to ?*".
  - To correctly answer the question, the model should uniquely identify the missing term, with only an exact correspondence counted as a correct match.
  - The question is answered by finding the word *d* whose representation x_d is closest to $x_b - x_a + x_c$ according to this:

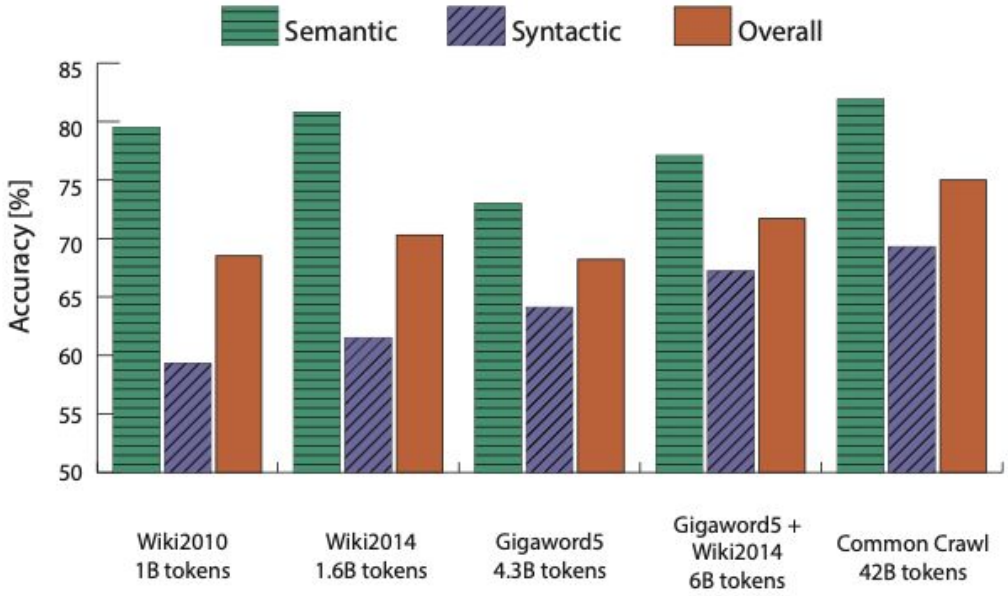
a:b :: c:?

man:woman :: king:?

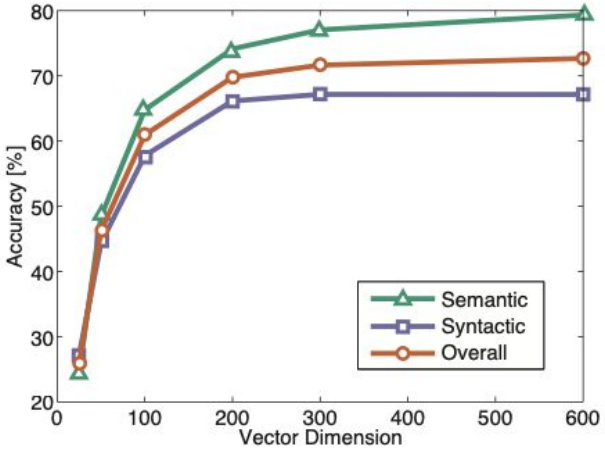
$$d = \underset{i}{\operatorname{argmin}} \frac{(\mathbf{x}_b - \mathbf{x}_a + \mathbf{x}_c)^\top \mathbf{x}_i}{||\mathbf{x}_b - \mathbf{x}_a + \mathbf{x}_c)||}$$

GloVe evaluation

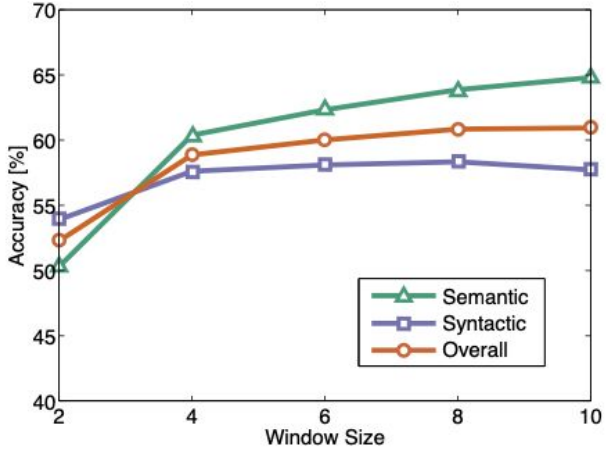
Model	Dim.	Size	Sem.	Syn.	Tot.
ivLBL	100	1.5B	55.9	50.1	53.2
HPCA	100	1.6B	4.2	16.4	10.8
GloVe	100	1.6B	67.5	54.3	60.3
SG	300	1B	61	61	61
CBOW	300	1.6B	16.1	52.6	36.1
vLBL	300	1.5B	54.2	64.8	60.0
ivLBL	300	1.5B	65.2	63.0	64.0
GloVe	300	1.6B	80.8	61.5	70.3
SVD	300	6B	6.3	8.1	7.3
SVD-S	300	6B	36.7	46.6	42.1
SVD-L	300	6B	56.6	63.0	60.1
CBOW†	300	6B	63.6	67.4	65.7
SG†	300	6B	73.0	66.0	69.1
GloVe	300	6B	77.4	67.0	71.7
CBOW	1000	6B	57.3	68.9	63.7
SG	1000	6B	66.1	65.1	65.6
SVD-L	300	42B	38.4	58.2	49.2
GloVe	300	42B	**81.9**	**69.3**	**75.0**



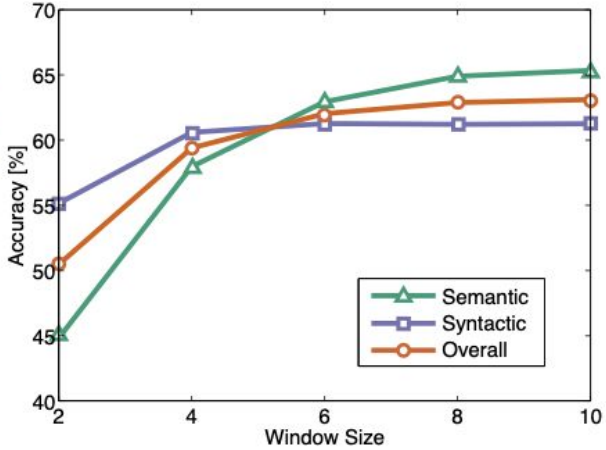
GloVe evaluation



(a) Symmetric context

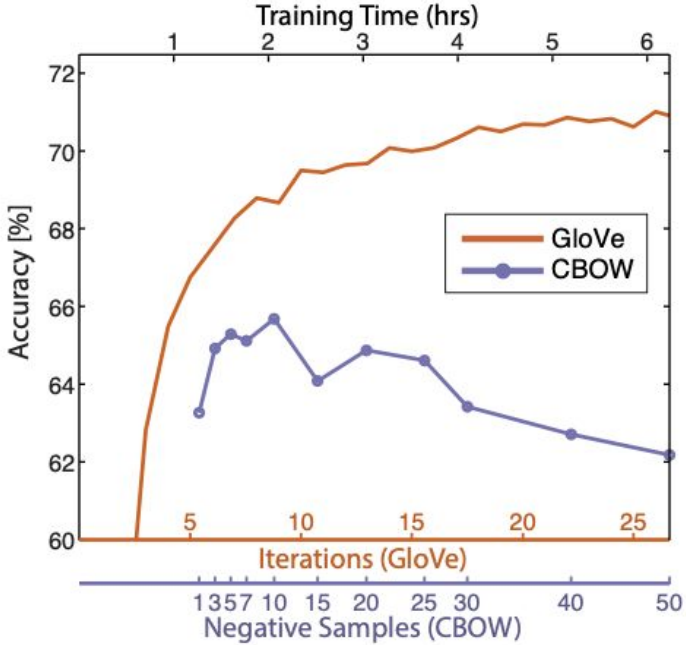


(b) Symmetric context

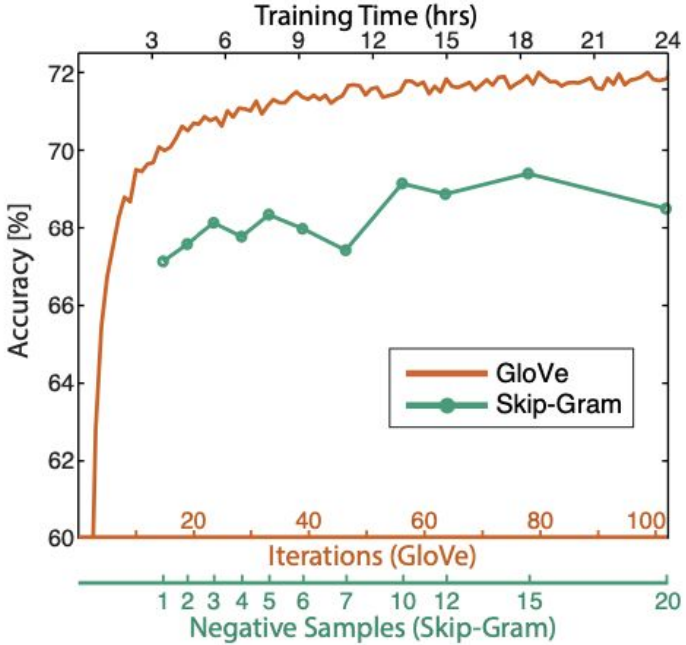


(c) Asymmetric context

GloVe evaluation



(a) GloVe vs CBOW



(b) GloVe vs Skip-Gram

Another intrinsic word evaluation

Correlation with **human judgements**

Example dataset: **WordSim353**: http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/

Word 1	Word 2	Human (mean)
tiger	cat	7.35
tiger	tiger	10
book	paper	7.46
computer	internet	7.58
plane	car	5.77
professor	doctor	6.62
stock	phone	1.62
stock	CD	1.31
stock	jaguar	0.92

Model	Size	WS353	MC	RG	SCWS	RW
SVD	6B	35.3	35.1	42.5	38.3	25.6
SVD-S	6B	56.5	71.5	71.0	53.6	34.7
SVD-L	6B	65.7	72.7	75.1	56.5	37.0
CBOW†	6B	57.2	65.6	68.2	57.0	32.5
SG†	6B	62.8	65.2	69.7	58.1	37.2
GloVe	6B	65.8	72.7	77.8	53.9	38.1
SVD-L	42B	74.0	76.4	74.1	58.3	39.9
GloVe	42B	**75.9**	**83.6**	**82.9**	**59.6**	**47.8**
CBOW*	100B	68.4	79.6	75.4	59.4	45.5

Extrinsic word evaluation

Directly see whether these word vectors improve accuracy in any NLP tasks. For example, here is the task called **named entity recognition**: identifying person, organization, or location.

Note: CoNLL-2003 English benchmark dataset for NER is a collection of documents from Reuters newswire articles, annotated with four entity types: person, location, organization, and miscellaneous.

Model	Dev	Test	ACE	MUC7
Discrete	91.0	85.4	77.4	73.4
SVD	90.8	85.7	77.3	73.7
SVD-S	91.0	85.5	77.6	74.3
SVD-L	90.5	84.8	73.6	71.5
HPCA	92.6	**88.7**	81.7	80.7
HSMN	90.5	85.7	78.7	74.7
CW	92.2	87.4	81.7	80.2
CBOW	93.1	88.2	82.2	81.1
GloVe	**93.2**	88.3	**82.9**	**82.2**

Summary

- **Negative sampling** is much better than the naive softmax approach. (It is not only use in NLP, e.g., FaceNet). There remains **three big limitations** of word2vec:
  - **Local windows** (use co-occurrence statistics in GloVe)
  - **Out-of-vocabulary** (FastText using character-based embedding)
  - **Not sure about its contextual meanings** (ELMo)
- Using **window-based co-occurrence statistics** can help encode meanings but can suffer from high dimensionality; addressed by SVD but not sure whether anything important is lost
- **GloVe** combines best of count-based vectors and prediction-based vectors
- Two types of evaluation: **intrinsic** and **extrinsic**.
- We haven't yet discussed about **FastText** (character-based embedding) and **ELMo** (context). Coming later! We will probably cover RNNs first, since you need to understand those first.