

---

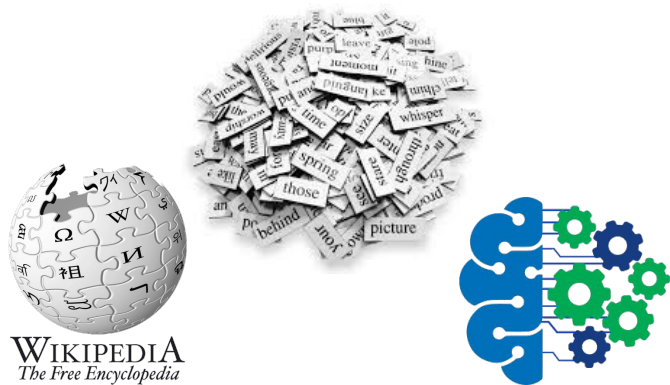
# GloVe: Global Vectors for Word Representation

— Karan Kishinani —  
Vatsal Srivastava

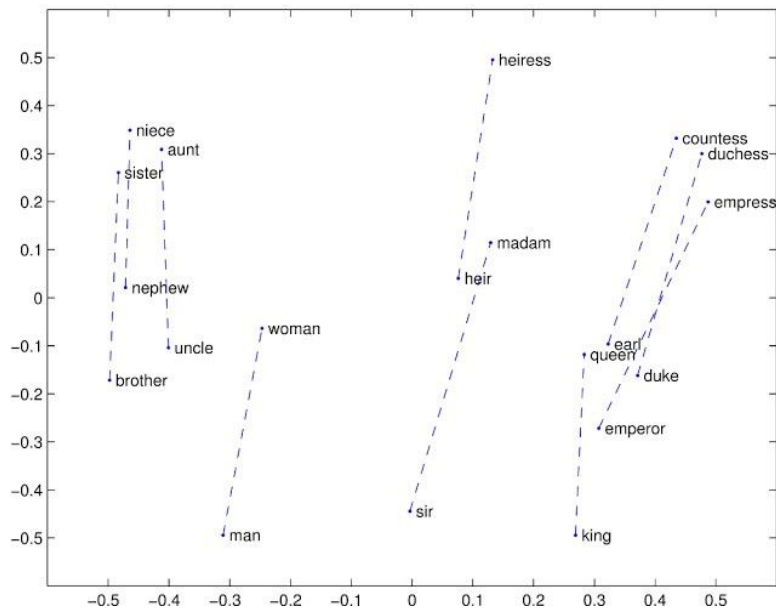
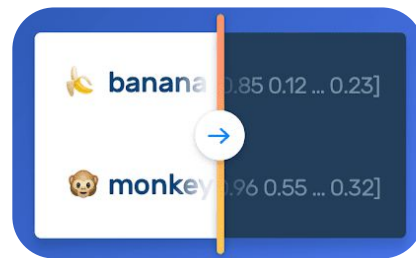
---

# What is GloVe?

- GloVe is an unsupervised learning algorithm for obtaining vector representations for words.



Training is performed on aggregated global word-word co-occurrence statistics from a corpus



The resulting representations showcase interesting linear substructures of the word vector space.

# What is GloVe? (more formally)

- A global log-linear regression model that combines the advantages of the two major model families in literature:
  - 1. Global Matrix Factorization
  - 2. Local Context window methods

# Past Approaches - 1. Global Matrix Factorization

- The process of using matrix factorization methods from linear algebra to perform rank reduction on a large term-frequency matrix
  - **term-document** frequencies, in which the rows are words and the columns are documents
    - **Latent Semantic Analysis (LSA)** where high dimensional matrix is reduced using **Singular Value Decomposition (SVD)**
  - **term-term** frequencies, which have words on both axes and measure co-occurrence (# times a word occurs in the context of another)
    - **Hyperspace Analogue to Language (HSL)**
      - Most frequent words contribute disproportionate amount to similarity measure (ex: *and*, *the*)

The diagram illustrates the SVD process with matrix dimensions and visual representations:

- $M = U \Sigma V^*$   
Dimensions:  $m \times n$  (M),  $m \times m$  (U),  $m \times n$  ( $\Sigma$ ),  $n \times n$  ( $V^*$ )
- $U = U^* I_m$   
Dimensions:  $m \times m$  (U),  $m \times m$  ( $U^*$ ),  $m \times m$  ( $I_m$ )
- $V = V^* I_n$   
Dimensions:  $n \times n$  (V),  $n \times n$  ( $V^*$ ),  $n \times n$  ( $I_n$ )

Visual representations of the matrices are shown with colored blocks and numerical values:

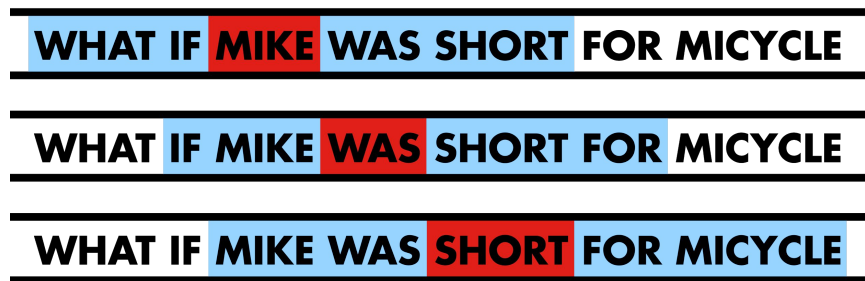
- $\Sigma$  is a diagonal matrix with values 0, 0, 0, 0, 0.
- $I_m$  is a 5x5 identity matrix with values 1, 0, 0, 0, 0; 0, 1, 0, 0, 0; 0, 0, 1, 0, 0; 0, 0, 0, 1, 0; 0, 0, 0, 0, 1.
- $I_n$  is a 5x5 identity matrix with values 1, 0, 0, 0, 0; 0, 1, 0, 0, 0; 0, 0, 1, 0, 0; 0, 0, 0, 1, 0; 0, 0, 0, 0, 1.

SVD Matrix Multiplication

# Past Approaches - 2. Local Context Window

- Learns semantics by passing a window over the corpus line-by-line and learning to either
  - Predict the surroundings of a given word (**Skip-gram model**), or
  - Predict a word given its surroundings (**Continuous bag-of-words model - CBOW**).









## CONTEXT WINDOW PROPAGATION



Context window is shaded blue and includes +/- 2 words around the relevant term

# Shortcomings in Past Work

The authors of the GloVe paper note that context window-based methods suffer from the disadvantage of **not learning from the global corpus statistics**. As a result, repetition and large-scale patterns may not be learned as well with these models as they are with global matrix factorization.

Matrix Factorization Method	Window-Based Method
 Fast Training	 Generate improved performance on other tasks
 Efficient usage of statistics	 Can capture complex patterns beyond word similarity
 Primarily used to capture word similarity	 Scale with corpus size
 Disproportionate importance given to large counts	 Inefficient usage of statistics

# Combining the best of both worlds: GloVe

- Benefits:
  - Fast training
  - Scalable to huge corpora
  - Good performance even with small corpus, and small vectors

## Co-occurrence Probabilities

- The authors of GloVe discovered via empirical methods that
  - Instead of learning the raw co-occurrence probabilities
  - Learn ratios of these co-occurrence probabilities
  - Ratio is better to discriminate subtleties in term-term relevance.

# Co-occurrence Probabilities (Example)

Suppose we want to study the relationship between two words:

$i = \textit{ice}$  and  $j = \textit{steam}$

Where  $P_{ij}$  = probability that word  $j$  appears in context of word  $i$

## CO-OCCURRENCE PROBABILITY

$$P_{ij} = P(j|i) = \frac{X_{ij}}{X_i} = \frac{X_{ij}}{\sum_k X_{ik}},$$

$X_{ij}$  = number of times word  $j$  occurs in the context of word  $i$ .

Table 1: Co-occurrence probabilities for target words *ice* and *steam* with selected context words from a 6 billion token corpus. Only in the ratio does noise from non-discriminative words like *water* and *fashion* cancel out, so that large values (much greater than 1) correlate well with properties specific to ice, and small values (much less than 1) correlate well with properties specific of steam.

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



# Naive Model

We want to take **data** from the corpus in the form of global **statistics** and learn a **function** that gives us **information about the relationship** between any two words in said corpus, given only the words themselves.

Authors discovered that ratios of co-occurrence probabilities are a good source of this information.

$$F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}.$$

# Vector Difference Model

Note that the **w**'s are real-valued word vectors. Now since we want to encode information about the **ratios between two words**, the authors suggest using vector differences as inputs to our function. Then we have the following:

$$F(w_i - w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}.$$

# Scalar-Input Model

So now we have two arguments, the **context word vector**, and the **vector difference** of the two words we're comparing.

Since the authors wish to take scalar values to scalar values (note the ratio of probabilities is a scalar), the dot product of these two arguments is taken, and so the next iteration of our model looks like this:

$$F \left( (w_i - w_j)^T \tilde{w}_k \right) = \frac{P_{ik}}{P_{jk}}.$$

# Vector Homomorphism for interchangeability

The next issue we will resolve is that of the labeling of certain words as “context words”. The problem with this is that the distinction between ordinary word vectors and context word vectors is in reality arbitrary: there is no distinction. **We should be able to interchange them without causing problems.**

The way we work around this is by requiring that **F** be a homomorphism from the additive group of real numbers to the multiplicative group of positive real numbers.

**VECTOR HOMOMORPHISM DEFINITION WITH SUBTRACTION**

$$F(w_a^T v_a - w_b^T v_b) = \frac{F(w_a^T v_a)}{F(w_b^T v_b)}, \forall w_a, v_a, w_b, v_b \in V.$$

# GloVe Homomorphism Condition

## GLOVE HOMOMORPHISM CONDITION

$$F \left( (w_i - w_j)^T \tilde{w}_k \right) = \frac{F(w_i^T \tilde{w}_k)}{F(w_j^T \tilde{w}_k)}.$$

$$\frac{P_{ik}}{P_{jk}} = \frac{F(w_i^T \tilde{w}_k)}{F(w_j^T \tilde{w}_k)},$$

$$F(w_i^T \tilde{w}_k) = P_{ik} = \frac{X_{ik}}{X_i}.$$

# Finding an $\mathbf{F} \rightarrow$ Exponential Function

We want to find a function  $\mathbf{F}$  which behaves “nicely”, that gives us a natural homomorphism between the additive and multiplicative real numbers, i.e. a function that turns addition into multiplication, or vice versa, as long as we have an inverse where we need it.

For simplicity here, we'll use  $\mathbf{e}$ , hence let  $\mathbf{F}$  be the exponential function. Then taking the natural logarithm of both sides in the above equation, we get

$$w_i^T \tilde{w}_k = \log P_{ik} = \log \left( \frac{X_{ik}}{X_i} \right) = \log X_{ik} - \log X_i.$$

$$w_i^T \tilde{w}_k + b_i + \tilde{b}_k = \log X_{ik}.$$

# Finally, we get the Weighted Least Squares Regression Model

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

Where  $f$  is our weighting function, which we require to have:

1. A limit of 0 as  $x$  goes to 0
2. To be nondecreasing (to not overweight rare co-occurrences)
3. To be relatively small for large values of  $x$  (to not overweight frequent co-occurrences)

# Weighing Function

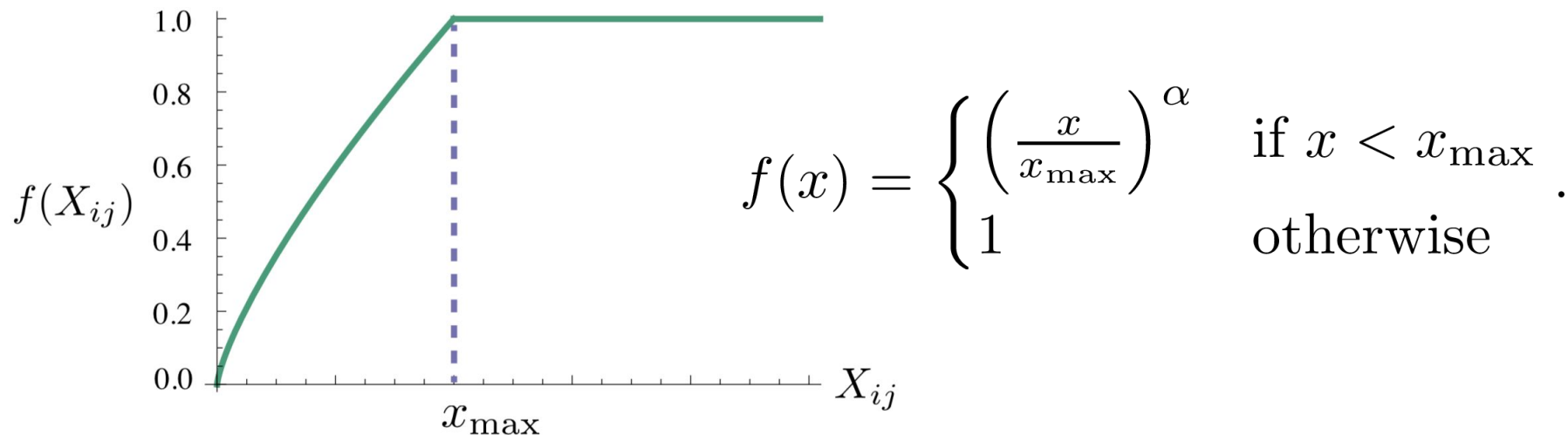


Figure 1: Weighing function  $f$  with  $\alpha = 3/4$ .



# Computational Complexity of the Model

Depends on the number of non-zero elements in the matrix  $X$ .

Naive estimate -  $O(|V|^2)$

Paper assumes the number of co-occurrences of word  $i$  with word  $j$ ,  $X_{ij}$ , can be modeled as a power-law function of the frequency of that word pair  $r_{ij}$

$$X_{ij} = \frac{k}{(r_{ij})^\alpha} .$$

Authors conclude that the complexity is  $O(|C|^{0.8})$  which is somewhat better than online window-based methods which scale like  $O(|C|)$ .

# Experiments and Model Evaluation

**Word Analogy** - 19544 questions

Semantic - "Athens is to Greece as Berlin is to \_\_?"

Syntactic - "dancing is to dance as flying is to \_\_?"

**Word Similarity**

**Named Entity Recognition** - CoNLL-2003 - collection of articles from Reuters newswire articles annotated with 4 entity types: person, location, organization and miscellaneous

# Experiments and Model Evaluation

Model trained on 5 corpora of varying sizes:

- 2010 Wikipedia dump with 1 billion tokens
- 2014 Wikipedia dump with 1.6 billion tokens
- Gigaword5 with 4.3 billion tokens
- Common Crawl web-data with 42 billion tokens
- Combination of Gigaword5 with Wikipedia 2014 with 6 billion tokens

Each corpus is tokenized and lower-cased and a vocabulary of the 400k most frequent words is used to construct a matrix of co-occurrence counts  $X$ .

# Model Analysis and Comparison

A similarity score is obtained from the word vectors by first normalizing each feature across the vocabulary and then calculating the cosine similarity. Then compute Spearman's rank correlation coefficient between this score and the human judgments.

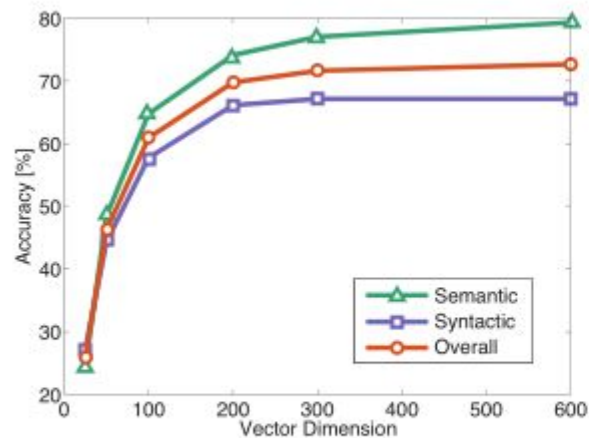
GloVe outperforms CBOW while using a corpus of half the size.

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	<u>72.7</u>	75.1	56.5	37.0
CBOW <sup>†</sup>	6B	57.2	65.6	68.2	57.0	32.5
SG <sup>†</sup>	6B	62.8	65.2	69.7	<u>58.1</u>	37.2
GloVe	6B	<u>65.8</u>	<u>72.7</u>	<u>77.8</u>	53.9	<u>38.1</u>
SVD-L	42B	74.0	76.4	74.1	58.3	39.9
GloVe	42B	<b><u>75.9</u></b>	<b><u>83.6</u></b>	<b><u>82.9</u></b>	<b><u>59.6</u></b>	<b><u>47.8</u></b>
CBOW*	100B	68.4	79.6	75.4	59.4	45.5

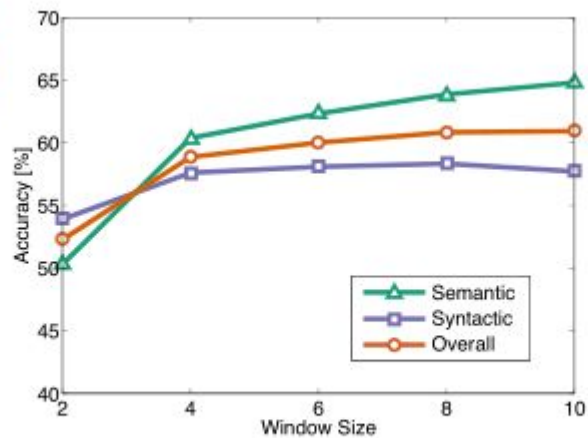
# Model Analysis and Comparison

F1 score on NER task with 50d vectors. Discrete is the baseline without word vectors. The GloVe model outperforms all other methods on all evaluation metrics, except for the CoNLL test set, on which the HPCA method does slightly better.

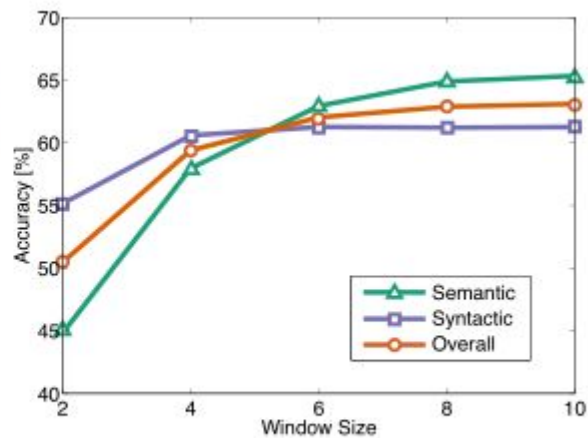
Model	Dev	Test	ACE	MUC7
Discrete	91.0	85.4	77.4	73.4
SVD	90.8	85.7	77.3	73.7
SVD-S	91.0	85.5	77.6	74.3
SVD-L	90.5	84.8	73.6	71.5
HPCA	92.6	<b>88.7</b>	81.7	80.7
HSMN	90.5	85.7	78.7	74.7
CW	92.2	87.4	81.7	80.2
CBOW	93.1	88.2	82.2	81.1
GloVe	<b>93.2</b>	88.3	<b>82.9</b>	<b>82.2</b>



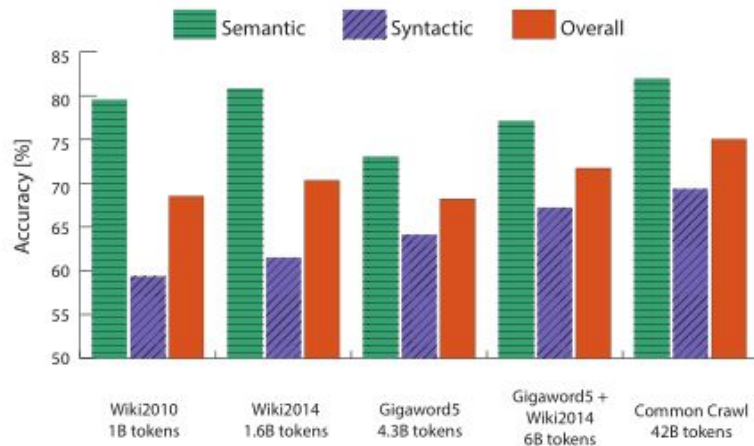
(a) Symmetric context



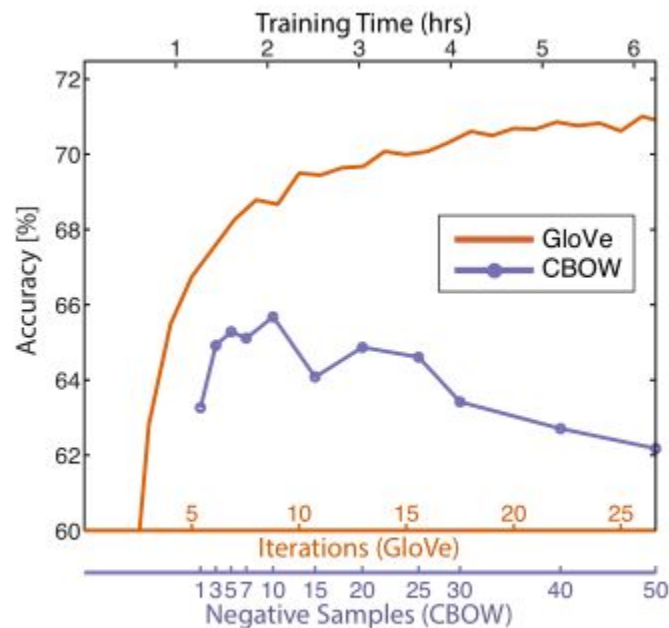
(b) Symmetric context



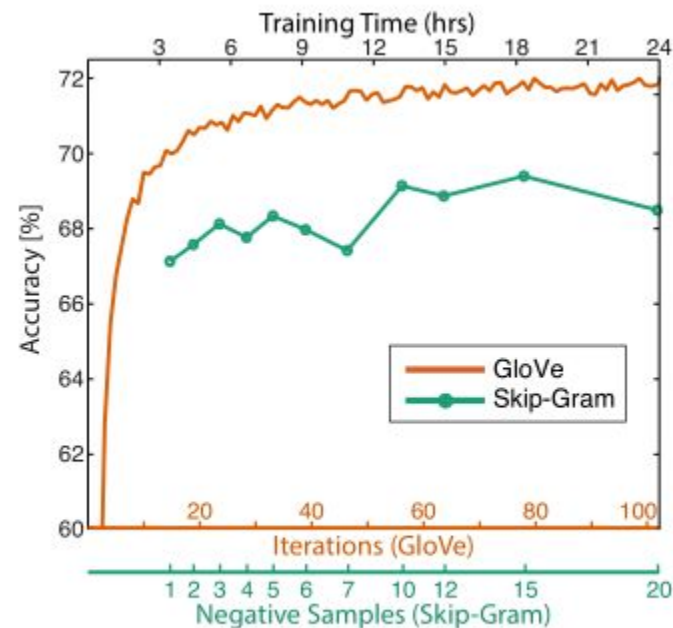
(c) Asymmetric context



Overall accuracy on the word analogy task as a function of training time, which is governed by the number of iterations for GloVe and by the number of negative samples for CBOW (a) and skip-gram (b). Training done for 300-dimensional vectors on the same 6B token corpus (Wikipedia 2014 + Gigaword 5) with the same 400,000 word vocabulary, and with a symmetric context window of size 10.



(a) GloVe vs CBOW



(b) GloVe vs Skip-Gram

# Analysis - Advantages of GloVe

## *Advantages:*

1. The goal of GloVe is very straightforward, to enforce the word vectors to capture sub-linear relationships in the vector space. Thus, it proves to perform better than Word2vec in the word analogy tasks.
2. GloVe adds some more practical meaning into word vectors by considering the relationships between word pair and word pair rather than word and word.
3. GloVe gives lower weight for highly frequent word pairs so as to prevent the meaningless stop words like “the”, “and” will not dominate the training progress.



# Analysis - Disadvantages of GloVe

## *Disadvantages:*

1. The model is trained on the co-occurrence matrix of words, which takes a lot of memory for storage. Especially, if you change the hyper-parameters related to the co-occurrence matrix, you have to reconstruct the matrix again, which is very time-consuming.

# Analysis - Limitations of GloVe

GloVe does not solve the problems like:

- How to learn the representation for out-of-vocabulary words
- Handling misspellings
- How to separate some opposite word pairs. For example, “good” and “bad” are usually located very close to each other in the vector space, which may limit the performance of word vectors in NLP tasks like sentiment analysis.

# Analysis: GloVe vs word2vec

- Word2Vec is a "predictive" model that predicts context given word.
- GloVe learns by constructing a co-occurrence matrix (words X context) that basically count how frequently a word appears in a context.
- Since it's going to be a gigantic matrix, we factorize this matrix to achieve a lower-dimension representation.

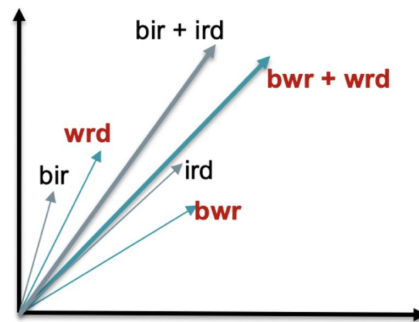
# Future Work that has been done

Misspelling Oblivious Word Embeddings **23 May 2019** ( <https://arxiv.org/pdf/1905.09755.pdf> )



bird  $\Rightarrow$  {bir, ird}

bwr  $\Rightarrow$  {bwr, wrd}



How it works:

$$(1 - \alpha) \underbrace{\sum_{i=1}^n \sum_{w_c \in C_i} \log \mathcal{P}(w_c | w_i; \theta)}_{\text{Semantic Loss}} + \underbrace{\frac{n\alpha}{|M|} \sum_{(w_m, w_e) \in M} \log \mathcal{P}(w_e | w_m; \theta)}_{\text{Spell Correction Loss}}$$