**Word Representation**

**Old-School Methods of Word Representation**

1. **Count-Based Methods**
   * **Term Context Matrix**: Words are represented by how often they co-occur with other words.
   * **TF-IDF (Term Frequency-Inverse Document Frequency)**: Weights words based on their importance.
   * **PMI (Pointwise Mutual Information) & PPMI (Positive PMI)**: Measures how often two words appear together compared to random chance.
   * **Problems with Count-Based Methods:**
     + Large, sparse matrices (many zero entries).
     + Primarily used for word similarity, not dense embeddings.
     + Overemphasizes frequent co-occurrences (e.g., stop words).

**Prediction-Based Methods**

1. **Word2Vec (W2V)**
   * **Skip-gram & CBOW (Continuous Bag of Words)**: Predicts words given context.
   * **Advantages:**
     + Generates dense word embeddings.
     + Captures complex word relationships (e.g., analogies like King - Man + Woman = Queen).
     + Improves performance in NLP tasks such as sentiment analysis and named entity recognition.
   * **Disadvantages:**
     + Computationally expensive.
     + Scales with corpus size (i.e., larger corpus requires more training time).
     + Does not utilize statistical co-occurrence effectively.

**Comparison of Count-Based vs Prediction-Based Methods**

| **Feature** | **Count-Based Methods** | **Prediction-Based Methods** |
| --- | --- | --- |
| **Training Speed** | Fast (single pass over data) | Slow (iterative training) |
| **Memory Usage** | Large (sparse matrices) | Compact (dense embeddings) |
| **Statistical Co-occurrence** | Well captured | Less effective |
| **Embedding Quality** | Limited | High-quality representations |
| **Scalability** | Less affected by corpus size | Slows down as corpus grows |

**Glove (Global Vectors for Word Representation)**

* Developed in **2014** by **Chris Manning’s group (Jeffrey Pennington et al.)**.
* **Combines best aspects of count-based & prediction-based methods.**
* Steps:
  1. Builds a **co-occurrence matrix** (like count-based methods).
  2. Uses **ratio of co-occurrence probabilities** instead of raw counts.
  3. Captures meaningful word relationships efficiently.
* **Example:**
  1. Consider the words **"ice"** and **"steam"** with context words **"solid," "gas," "water," "fashion."**
  2. **"Ice"** is more likely to co-occur with **"solid"** and **"water"**, but less with **"gas"** or **"fashion"**.
  3. **"Steam"** is more likely to co-occur with **"gas"** and **"water"**, but not **"solid"**.
  4. By computing probability ratios, Glove captures word relationships better than raw counts.

**Advantages of Glove**

* **Leverages co-occurrence statistics** effectively.
* **Faster to train than Word2Vec** while producing high-quality embeddings.
* **Captures analogies and relationships** similar to Word2Vec.
* **More stable embeddings** since training doesn’t depend on random initialization like Word2Vec.

**Summary**

* **Count-based methods** (TF-IDF, PMI) are fast but have sparse representations.
* **Prediction-based methods** (Word2Vec) generate high-quality embeddings but are computationally expensive.
* **Glove combines both approaches**, using co-occurrence probabilities to generate dense embeddings efficiently.
* **Glove embeddings are widely used** in NLP for tasks like sentiment analysis, named entity recognition, and machine translation.

-------------------------------------------------------------------------------------------------------------------------

**1. Introduction to Learning Word Embeddings**

* Given a pair of words **I** and **J**, we define their corresponding embeddings as **w\_i** and **w\_j**.
* The similarity between these embeddings is captured using the **dot product**:  
  wi⋅wjw\_i \cdot w\_j
* This similarity should be modeled using the **probability** derived from a count-based approach.
* Instead of taking the probability directly, we use the **log probability**.

**2. Asymmetry in Context and Target Words**

* When **J** appears as a context word and **I** appears as a target word, the ratio is different from when their roles are reversed.
* This leads to an **asymmetric relationship** in probability calculations.

**3. Mathematical Formulation**

* **x\_{ij}**: Count of the number of times words **I** and **J** co-occur.
* **x\_i**: Total count of occurrences of word **I**.
* **Probability Formula:**  
  P(J∣I)=xijxiP(J|I) = \frac{x\_{ij}}{x\_i}  
  This is the probability of **J** occurring given that **I** has occurred.
* Similarly, when J is the target and I is the context: P(I∣J)=xijxjP(I|J) = \frac{x\_{ij}}{x\_j}
* We add these probabilities to derive the **final formula**: wi⋅wj=log⁡xij+12(log⁡xi+log⁡xj)w\_i \cdot w\_j = \log x\_{ij} + \frac{1}{2} ( \log x\_i + \log x\_j )
* The terms **log x\_i** and **log x\_j** are **word-specific biases**, which are modeled as bias terms **b\_i** and **b\_j** in the neural network.

**4. Objective Function for Training**

* The goal is to find **w\_i** and **w\_j** that minimize the difference between the **dot product similarity** and the **log co-occurrence probability**.
* Loss Function: (wi⋅wj+bi+bj−log⁡xij)2( w\_i \cdot w\_j + b\_i + b\_j - \log x\_{ij} )^2
* This is a **squared loss function** used for optimization.

**5. Addressing Disproportionate Weights**

* The function assigns **higher weights** to frequent words, which can skew results.
* To counteract this, a **weighting function** **f(x\_{ij})** is introduced.
* Properties of **f(x\_{ij})**:
  1. **Continuous function**
  2. **Non-decreasing** with co-occurrence frequency.
  3. **Small values for very high co-occurrence counts** to prevent overemphasis.
* The function used in **GloVe**: f(x)=(xxmax)α if x<xmax, else 1f(x) = \left( \frac{x}{x\_{max}} \right)^{\alpha} \text{ if } x < x\_{max}, \text{ else } 1
* Here, **x\_max** is the cutoff beyond which co-occurrence counts are treated equally.

**6. Comparison with Word2Vec**

* **GloVe combines count-based and prediction-based approaches**, unlike Word2Vec which relies purely on context prediction.
* **Advantages of GloVe:**
  + Faster training.
  + Scalable to large corpora.
  + Works well even with small datasets.

**7. Applications and Insights from GloVe Embeddings**

* **Analogy Tasks**: Example:
  + **Beijing** : **China** :: **Delhi** : **India**
  + **Man** : **Doctor** :: **Woman** : **Nurse** (illustrating biases in embeddings).
* **Word Evolution Over Time**:
  + Historical embeddings from decades of text (e.g., 1950s to 2010s) reveal **semantic shifts**.
  + Example: "Broadcast" originally referred to **spreading seeds** in agriculture but now means **media transmission**.

**8. Key Takeaways**

* **GloVe efficiently captures semantic relationships using co-occurrence probabilities**.
* It bridges the gap between **count-based and prediction-based embeddings**.
* **Weighting function helps prevent dominance of frequent words**.
* **Word analogies and bias detection** are interesting applications.
* **Useful in various NLP tasks** like text classification, machine translation, and semantic analysis.