[03-17-2025], Lecture Note: GenAI Responsibility: Overview

[Mohammed Muneeb, Taabish Sutriwala]

CS 516: Responsible Data Science and Algorithmic Fairness; Spring 2025
Abolfazl Asudeh; www.cs.uic.edu/∼ *asudeh/teaching/archive/cs516spring25/*

1 Overview

In this lecture, we discussed the intrinsic bias in word embeddings, particularly within the context of large language models (LLMs). The primary focus was on understanding embeddings, how they are generated, and methods to detect and measure biases.

2 Introduction to Embeddings:

Embeddings are dense vector representations of words, sentences, or even entire documents. They serve as a numerical form that machines can use to interpret and process textual or symbolic information in a meaningful way. Unlike traditional symbolic representations like one-hot encoding, embeddings map each item to a low-dimensional, continuous space.

3 Challenges with Basic Representations:

Basic representations like one-hot encoding and random vectors come with significant limitations. One-hot encoding assigns each word a unique binary vector where only one element is active. While simple, it does not capture any semantic similarity, sentence structure, or sentiment. Words like "dog" and "puppy" are treated as entirely unrelated, even though they are semantically close. Additionally, one-hot vectors are sparse and high-dimensional, making them inefficient and incapable of generalizing across similar words, which in turn requires large amounts of training data.

Random vectors, on the other hand, assign arbitrary dense vectors to words. While they may preserve some order when used in models like RNNs, they fail to capture meaningful relationships between words. Since the vectors are generated randomly, there is no inherent semantic information, and the results can vary between runs. Both approaches lack the ability to represent contextual or syntactic nuances, which is why more advanced embedding techniques are now preferred in modern NLP.

4 Vector Representation and Semantic Similarity:

In natural language processing, words are often represented as vectors in a high-dimensional continuous space. These vector representations encode semantic information, allowing words with similar meanings to occupy nearby positions in this space. For example, the words "king" and "queen" would have vectors that are close together, while unrelated words like "king" and "banana" would be far apart.

The semantic similarity between two words can be measured using cosine similarity, which calculates the cosine of the angle between their vectors. A smaller angle (closer to 1 in cosine value) indicates greater similarity. This method enables models to reason about word meaning in a geometric way, making it possible to group related terms, complete analogies, and enhance understanding of language context. Such representations are foundational in embedding models like Word2Vec, GloVe, and BERT.

5 Generating Semantic Similarities:

Semantic similarity between words can be inferred from how frequently they appear together in large text corpora, such as Wikipedia or news articles. This idea is based on the distributional hypothesis, which suggests that words appearing in similar contexts tend to have similar meanings. For instance, the words "doctor" and "nurse" often appear in similar sentences or paragraphs, indicating a semantic connection.

To quantify this relationship, statistical methods are used to analyze word co-occurrence patterns. While traditional methods like TF-IDF measure the importance of individual words in documents, techniques for semantic similarity extend this idea by analyzing pairwise co-occurrence how often two words appear together across contexts. This results in a similarity score that reflects their relationship. More advanced models, like GloVe, use co-occurrence matrices combined with optimization techniques to produce dense vector representations that naturally encode these semantic similarities.

6 Methods of Allocating Vectors:

Vector representations of words can be generated using different methodologies that capture the relationships and meanings between words. Two prominent techniques include physics-inspired models and neural network-based models, each offering unique insights into how semantic similarity is encoded.

6.1 Physics-Inspired Method (Spring Model)

In the Spring Model, words are treated as particles or nodes in space, and semantic similarity between words is modeled as springs connecting them. The closer the meaning of two words, the stronger and shorter the spring. The system aims to position the words in a way that minimizes the total energy of the system just like how physical systems reach equilibrium.

As a result, similar words end up closer together, while dissimilar words are pushed further apart. This model provides an intuitive and interpretable visualization of word similarity, often used in early embedding visualizations or graph layouts.

6.2 Neural Network Method (Word2Vec)

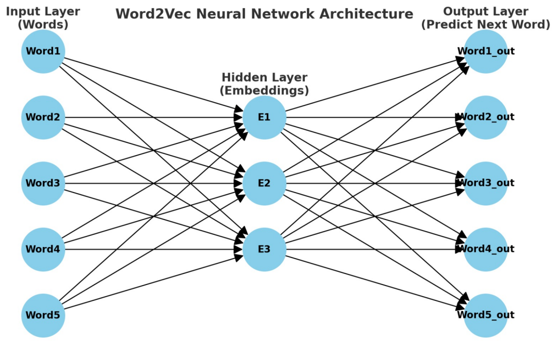
The Word2Vec model is a neural network-based approach that learns embeddings by predicting context. It comes in two main variants:

- **CBOW (Continuous Bag of Words)**: Predicts a word given its surrounding context.
- **Skip-gram**: Predicts surrounding words given a central word.

The architecture includes:

- An input layer representing the current word.
- A hidden embedding layer with no activation, which stores the learned word vectors.
- An output layer using softmax to predict word probabilities over the vocabulary.

Word2Vec is trained on large corpora like Wikipedia using backpropagation and gradient descent to optimize the prediction accuracy. The resulting embeddings capture both syntactic patterns (like plural vs singular) and semantic relationships (like "Paris" being to "France" what "Tokyo" is to "Japan").



7 Limitations of Standard Embeddings:

Traditional word embeddings, such as those produced by Word2Vec or GloVe, are trained using fixed-size context windows. These models predict a word based on a limited number of neighboring words, which means they only capture local context and often miss out on the broader grammatical and syntactic structure of the sentence. For instance, the word "bank" will have the same vector whether it appears in "river bank" or "financial bank", because the model doesn't fully capture long-range dependencies or sentence-level meaning.

This limitation makes standard embeddings context-independent, meaning each word has a single fixed vector regardless of how it is used. This becomes a problem in more complex NLP tasks where word meaning is highly dependent on its usage in context.

To overcome this, modern architectures like transformers introduce attention mechanisms, which allow the model to weigh and consider all words in a sentence when generating the representation for a particular word. This results in contextual embeddings, where the same word can have different vectors depending on its role in the sentence. Models such as BERT and GPT leverage this approach to generate richer, more accurate language representations that capture both semantic meaning and syntactic structure across the entire input.

8 Measuring Bias in Embeddings:

Word embeddings are powerful tools for capturing semantic relationships, but they can also encode social biases present in the data they are trained on. These biases can propagate unfair associations and stereotypes in downstream applications. Measuring and mitigating these biases is essential for building responsible AI systems.

- **Bias Definition**

  Bias in embeddings refers to situations where neutral words exhibit disproportionate similarity to one demographic group over another. For instance, the word "doctor" might be closer to the vector for "he" than "she", suggesting an unintended gender bias in the learned representation. These biases are not

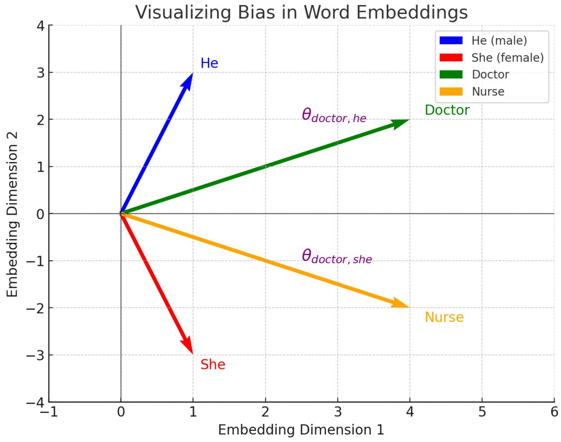
- **Vector Representations for Demographic Groups**

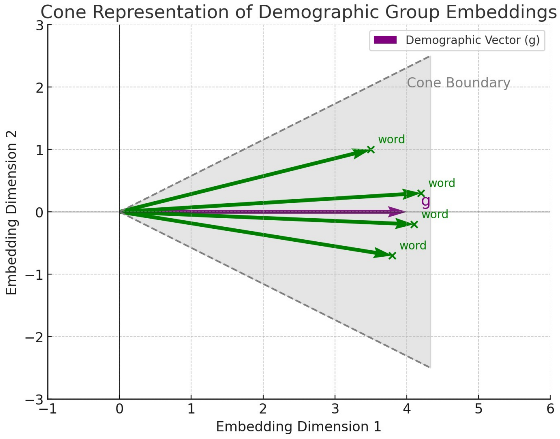
  To detect such bias, we first need a way to represent demographic groups as vectors. This is done by aggregating multiple words that are representative of a demographic. For gender, this might include:

  - Group 1 (Male): he, him, man, male, father
  - Group 2 (Female): she, her, woman, female, mother

The average vector of all words in each group is computed to serve as the group representation in the embedding space. Using an average helps smooth individual variations and provides a more robust reference point for comparison. Statistical techniques such as confidence intervals can also be applied to ensure the reliability of these group representations.

$$E[\vec{g}_i] = \frac{1}{K}\sum \vec{v}_i$$





9 Bias Measurement: WEAT (Word Embedding Association Test)

The Word Embedding Association Test (WEAT) is a widely used method for quantifying bias in embeddings. It measures how strongly two sets of target words (e.g., occupations like "engineer", "nurse") are associated with two attribute groups (e.g., male vs female terms). The idea is to compute the average semantic similarity difference between groups for these target words.

WEAT Bias Score Formula:

$$\text{WEAT} = \frac{1}{L}\left[\sum_{a_i \in A}\left(\text{Sim}(\vec{a}_i, \vec{g}_1) - \text{Sim}(\vec{a}_i, \vec{g}_2)\right) + \sum_{b_i \in B}\left(\text{Sim}(\vec{b}_i, \vec{g}_2) - \text{Sim}(\vec{b}_i, \vec{g}_1)\right)\right]$$

Where:

- A and B are two sets of stereotype-associated words.
- $\vec{g}_1$ and $\vec{g}_2$ are the demographic group vectors (e.g., male and female).
- Sim($\vec{a}_i$,$\vec{g}_1$) is the similarity (usually cosine) between word i and group vector g.
- L is a normalization constant based on the number of words.

A positive bias score indicates stronger association of words in set A with group g1, and words in set B with group g2, potentially revealing biased patterns.

10 Example and Impact of Embedding Bias:

A well-known example of embedding bias occurred in earlier versions of Google Translate. When translating from gender-neutral languages like Turkish or Finnish to English, the system often introduced gendered stereotypes. For instance, the sentence “O bir doktor” in Turkish, which means “They are a doctor” (gender-neutral), was often translated as “He is a doctor”, while “O bir hemşire” (“They are a nurse”) became “She is a nurse”. This reflects an underlying gender bias in the word embeddings used by the translation model — associating doctor with male and nurse with female.

Such biases are not trivial. They can reinforce harmful stereotypes, impact user trust, and introduce discrimination in systems that rely on natural language processing. In real-world applications like hiring algorithms, recommendation systems, or chatbots, biased embeddings can lead to unequal treatment of different demographic groups. Addressing these issues is critical to developing fair and responsible AI systems.

11 Conclusion Key Insights:

Word embeddings play a central role in modern natural language processing by enabling machines to understand and manipulate language through vector representations. However, these embeddings are not without flaws — they can encode and even amplify intrinsic social biases present in training data. Such biases can affect the fairness and trustworthiness of downstream applications, making their detection and mitigation essential.

Detecting bias in embeddings involves a combination of geometric and statistical analyses. By examining how neutral or occupation-related words relate to demographic group vectors in the embedding space, researchers can uncover subtle yet significant associations that reflect real-world stereotypes. Tools like the Word Embedding Association Test (WEAT) and its sentence-level extension, SEAT, offer structured and quantifiable ways to measure these associations. These methods allow us to evaluate not just the presence of bias, but its magnitude — providing a foundation for future work in debiasing and ensuring fairer AI systems.