Can Word embedding find the inherent bias in the corpus?

Ramaseshan Ramachandran

August 6, 2026

1 Inherent bias in the Corpus

Word embeddings are numerical representations of words that computers can understand and work with. They are constructed by training on a large text corpus so that words occurring in similar lexical/semantic contexts get similar low-dimensional vectors. They are constructed using count-based co-occurrence factorization or prediction-based models using a very large corpus[**?**].

These models are often trained on news corpora, business documents, and web content where certain words dominate the discourse around the term in question. For example, the word *apple* may have lexically and semantically associated with apple products, corporate-heavy context, leading to stronger semantic associations with technology rather than the fruit. These word embeddings often reflect real-world biases from the human-created corpus as they inevitably encode the biases present in their training corpora.

2 $\overrightarrow{apple} - \overrightarrow{iphone}$

Word embedding exposes how the corpus treats the polysemous word "apple". The question is whether it is predominantly associated with the fruit, companies or technological products. The resulting vector space reveals the inherent bias toward one interpretation over another based on the frequency and context of co-occurring terms in the training data. Find the example below that shows inherent bias in the word *apple*.

Similar words for apple

When $\vec{\text{apple}}$ is heavily influenced by technology contexts (due to Apple Inc.'s prominence in digital text), the subtraction $\vec{\text{apple}} - \vec{\text{iphone}}$ may yield vectors closer to corporate or brand-related concepts rather than agricultural or food-related terms. This demonstrates how corpus composition directly shapes semantic representations, making the absence of balanced representation a critical consideration in embedding-based applications.

apple	0	raisin	0.574
iphone	0.266	pecan	0.576
ipad	0.287	cranberry	0.584
apples	0.356	butternut	0.588
blackberry	0.361	cider	0.591
ipod	0.365	apricot	0.603
macbook	0.383	tomato	0.607
mac	0.391	rosemary	0.615
android	0.391	rhubarb	0.616
google	0.395	feta	0.618
microsoft	0.418	apples	0.622
ios	0.433	avocado	0.623
iphones	0.445	fennel	0.630
touch	0.446	chutney	0.631
sony	0.447	spiced	0.632

The arithmetic thus serves as a diagnostic tool for understanding what cultural, commercial, or contextual biases your word vectors have absorbed from their training environment.

When analyzing word vectors, the word "apple" serves as a classic example of the challenges and nuances of representing meaning numerically. Because word embeddings are learned from the contexts in which words appear, the resulting vector for "apple" is often more similar to vectors for technology companies than for fruits [1][2].

3 The "Apple" Dilemma: Company vs. Fruit

In many standard pre-trained models, such as those trained on large news datasets, the word "apple" appears far more frequently in discussions about technology than about agriculture or food [1][3]. It co-occurs with terms like "iPhone," "MacBook," "iOS," "stock," and the names of other major tech corporations like **Google, IBM, and Samsung** [3]. In contrast, its appearances alongside words like "pie," "orchard," "tree," or other fruits are comparatively rare in these corpora.

Word embedding algorithms learn to place words with similar contexts close to each other in a high-dimensional space. As a result, the single, static vector representing "apple" in models like word2vec is mathematically positioned nearer to the vectors for other tech companies than to vectors for fruits like "grape" or "banana" [1][2]. This reflects the statistical reality of the training data, where "Apple" the company dominates the discourse.

This can lead to practical issues. For instance, a search system relying on such embeddings might struggle to distinguish between the brand and the fruit, potentially returning recipes for apple pie to a user searching for Apple products [4].

4 Contextual Embeddings as a Solution

This ambiguity highlights a limitation of static word embeddings, where one word has only one vector representation regardless of its use [5]. Modern NLP models, such as BERT, address this by creating **contextual embeddings**. These models generate a unique vector for a word each time it appears, based on the surrounding words in the sentence [5][6].

With a contextual model: * In the sentence, "I ate an **apple**," the vector for "apple" is influenced by the word "ate" and will be more similar to the vector for "fruit" [6]. * In the sentence, "I bought an **Apple**," the vector for "Apple" is influenced by "bought" and will be closer to vectors for "company" or "product" [6].

This dynamic approach allows the model to differentiate between meanings based on the immediate context, resolving the ambiguity inherent in words with multiple senses [7][5].