Word2Vec Skip-Gram

TF-IDF Challenges and Need for word2Vec

TF-IDF (Term Frequency-Inverse Document Frequency) is a useful technique for determining the importance of words in a document relative to a collection of documents (corpus). However, it has several limitations:

1. Lack of Semantic Understanding:

- TF-IDF treats words as independent units and doesn't understand the underlying meaning or context. Synonyms or words with similar meanings are treated as distinct, potentially leading to inaccurate relevance scores.
- It fails to capture semantic relationships between words (e.g., "car" and "automobile" would be considered different).

2. Ignores Word Order and Context:

- TF-IDF operates on a "bag-of-words" model, meaning it disregards the order in which words appear in a document. This can be problematic for understanding the true meaning of sentences, especially when negation or specific phrasing is involved. For example, "not good" and "good not" would be treated the same.
- It doesn't capture the context in which a word is used, which can be crucial for disambiguation.

3. Difficulty with Polysemy:

- Words with multiple meanings (polysemous words) are treated as a single entity by TF-IDF. It cannot differentiate between the different senses of a word based on its context.

4. Sensitivity to Document Length:

- Longer documents tend to have higher term frequencies, which can inflate the TF scores and make longer documents seem more important than shorter ones, even if they are not more relevant. Normalization techniques can help mitigate this, but it remains a potential issue.

5. Challenges with Rare Terms:

- While IDF aims to give higher weight to rare terms, extremely rare terms that appear in very few documents might get disproportionately high IDF scores, potentially overemphasizing their importance even if they are not truly significant.

6. Doesn't Capture Phrase Meaning:

- TF-IDF treats individual words. It doesn't inherently understand the meaning of multi-word phrases (e.g., "machine learning") as single units. Techniques like n-grams can be used to address this, but they increase the dimensionality of the data.

7. Sparsity of the Matrix:

- When dealing with large vocabularies, the TF-IDF matrix can become very sparse (mostly zeros), which can be computationally inefficient and may pose challenges for some machine learning algorithms.

8. Out-of-Vocabulary (OOV) Words:

- TF-IDF cannot handle words that were not present in the training corpus. Any new words encountered during testing or in new documents will be ignored.

Word2vec - skip gram model

The Word2Vec Skip-Gram model is a popular technique in Natural Language Processing (NLP) used to learn vector representations (embeddings) of words from a large corpus of text. These embeddings capture semantic relationships between words, meaning words that appear in similar contexts in the text will have vector representations that are close to each other in the vector space.

Because these words frequently appear in the vicinity of "king," a Skip-Gram model would learn to place their vector embeddings close to the vector embedding of "king" in the vector space. This reflects their semantic relationship – they are all concepts associated with royalty and governance.

You shall know a word by the company it keeps.

J.R. Firth, 1957

1. The Core Idea:

The Skip-Gram model works by taking a **center word** as input and trying to predict the **surrounding context words** within a defined window size. The underlying assumption is that words that frequently appear near each other are semantically related.

Example 1: **Words related to "king"**

- **queen:** "The **king** and the **queen**..."
- **throne:** "The **king** sat upon his **throne**."
- **royal:** "The **king** issued a **royal** decree."
- **crown:** "The **king** wore a golden **crown**."
- **kingdom:** "The **king** ruled his **kingdom** wisely."
- **prince:** "The young **prince**, son of the **king**..."
- **power:** "The **king** held great **power**."
- **monarch:** "The **king**, a powerful **monarch**..."

Example 2: Words related to "coffee"

You'd often find "coffee" near words like:

- **cup:** "She held a **cup** of **coffee**."
- **drink:** "He ordered his favorite **drink**, **coffee**."
- **morning:** "**Morning** wouldn't be the same without **coffee**."
- **cafe:** "They met at the local **cafe** for **coffee**."
- **hot:** "A **hot** cup of **coffee** warmed her hands."
- **brew:** "The aroma of freshly **brew**ed **coffee** filled the air."
- **taste:** "The strong **taste** of **coffee** woke him up."
- **java:** "He needed his daily dose of **java**, **coffee**."

Mathematically Represented: Each word in the vocabulary is assigned a unique vector of real numbers. These vectors are the mathematical representation of the words.

High-Dimensional Space: While the dimensionality of these vectors (e.g., 100, 300) is much lower than the vocabulary size (which can be hundreds of thousands or millions), it's still considered a high-dimensional space compared to our typical 2D or 3D understanding. This high dimensionality allows the model to capture subtle nuances of meaning and relationships.

Vectors Close to Each Other: The training process of Skip-Gram (and other word embedding models) aims to position the vectors of words that frequently appear in similar contexts close to each other in this high-dimensional space. The "closeness" is typically measured using distance metrics like cosine similarity.

Similar Meanings: The proximity of vectors in the embedding space reflects the semantic similarity between the corresponding words. Words that are used in similar ways, have related meanings, or are interchangeable in certain contexts will have vectors that are close together.

Pre-requisite: Basics of neural network

Read the concept from the below link

https://www.tensorflow.org/text/tutorials/word2vec

Here's how Skip-Gram works with different window sizes, using the example sentence: "I like to play football very much".

First, let's tokenize the sentence and create a vocabulary with IDs:

- **Tokens:** ["I", "like", "to", "play", "football", "very", "much"]

Vocabulary with word-to-ID mapping:

```
{
  "I": 0,
  "like": 1,
  "to": 2,
  "play": 3,
  "football": 4,
  "very": 5,
  "much": 6
}
```

Here's how Skip-Gram works with different window sizes, using the example sentence: "I like to play football very much".

First, let's tokenize the sentence and create a vocabulary with IDs:

- **Tokens:** ["I", "like", "to", "play", "football", "very", "much"]

Vocabulary with word-to-ID mapping:

```
{
  "I": 0,
```

```
    "like": 1,
    "to": 2,
    "play": 3,
    "football": 4,
    "very": 5,
    "much": 6
}
```

Window Size = 1

For each target word, we consider only the word immediately to its left and right.

- **Target Word: "I"**
    - There is no word to the left.
    - Context word to the right: "like"
    - Pair: (I, like)
- **Target Word: "like"**
    - Context word to the left: "I"
    - Context word to the right: "to"
    - Pairs: (like, I), (like, to)
- **Target Word: "to"**
    - Context word to the left: "like"
    - Context word to the right: "play"
    - Pairs: (to, like), (to, play)
- **Target Word: "play"**
    - Context word to the left: "to"
    - Context word to the right: "football"
    - Pairs: (play, to), (play, football)
- **Target Word: "football"**
    - Context word to the left: "play"
    - Context word to the right: "very"
    - Pairs: (football, play), (football, very)

- **Target Word: "very"**
    - Context word to the left: "football"
    - Context word to the right: "much"
    - Pairs: (very, football), (very, much)
- **Target Word: "much"**
    - Context word to the left: "very"
    - There is no word to the right.
    - Pair: (much, very)

Window Size = 2

For each target word, we consider the two words to its left and the two words to its right.

- **Target Word: "I"**
    - There are no words to the left.
    - Context words to the right: "like", "to"
    - Pairs: (I, like), (I, to)
- **Target Word: "like"**
    - Context word to the left: "I"
    - Context words to the right: "to", "play"
    - Pairs: (like, I), (like, to), (like, play)
- **Target Word: "to"**
    - Context words to the left: "I", "like"
    - Context words to the right: "play", "football"
    - Pairs: (to, I), (to, like), (to, play), (to, football)
- **Target Word: "play"**
    - Context words to the left: "like", "to"
    - Context words to the right: "football", "very"
    - Pairs: (play, like), (play, to), (play, football), (play, very)
- **Target Word: "football"**
    - Context words to the left: "to", "play"
    - Context words to the right: "very", "much"

    - Pairs: (football, to), (football, play), (football, very), (football, much)
- **Target Word: "very"**
    - Context words to the left: "play", "football"
    - Context word to the right: "much"
    - Pairs: (very, play), (very, football), (very, much)
- **Target Word: "much"**
    - Context words to the left: "football", "very"
    - There are no words to the right.
    - Pairs: (much, football), (much, very)

Window Size = 3

For each target word, we consider the three words to its left and the three words to its right.

- **Target Word: "I"**
    - There are no words to the left.
    - Context words to the right: "like", "to", "play"
    - Pairs: (I, like), (I, to), (I, play)
- **Target Word: "like"**
    - Context word to the left: "I"
    - Context words to the right: "to", "play", "football"
    - Pairs: (like, I), (like, to), (like, play), (like, football)
- **Target Word: "to"**
    - Context words to the left: "I", "like"
    - Context words to the right: "play", "football", "very"
    - Pairs: (to, I), (to, like), (to, play), (to, football), (to, very)
- **Target Word: "play"**
    - Context words to the left: "I", "like", "to"
    - Context words to the right: "football", "very", "much"
    - Pairs: (play, I), (play, like), (play, to), (play, football), (play, very), (play, much)
- **Target Word: "football"**

    - Context words to the left: "like", "to", "play"
    - Context words to the right: "very", "much"
    - Pairs: (football, like), (football, to), (football, play), (football, very), (football, much)
- **Target Word: "very"**
    - Context words to the left: "to", "play", "football"
    - Context words to the right: "much"
    - Pairs: (very, to), (very, play), (very, football), (very, much)
- **Target Word: "much"**
    - Context words to the left: "play", "football", "very"
    - There are no words to the right.
    - Pairs: (much, play), (much, football), (much, very)

Code (For better understanding of how a Neural network is trained using this dataset)

https://colab.research.google.com/drive/1h5jjO6oUNfKsufIRIN7iOqtiR6X9CDnh#scrollTo=tDIfVW2Msfvl