Text Similarity Using Word2Vec

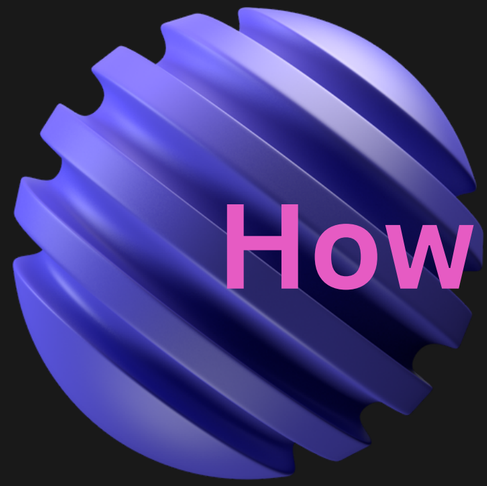
Improve productivity and time management

Contents

Introduction

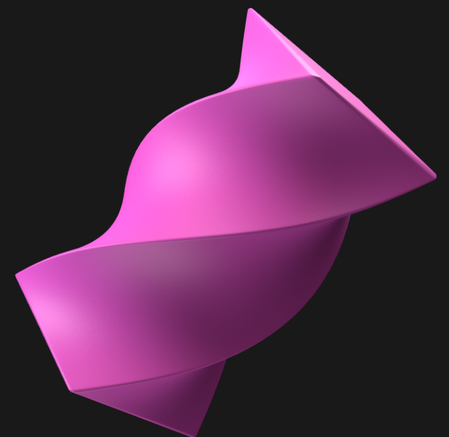
What is Text Similarity?

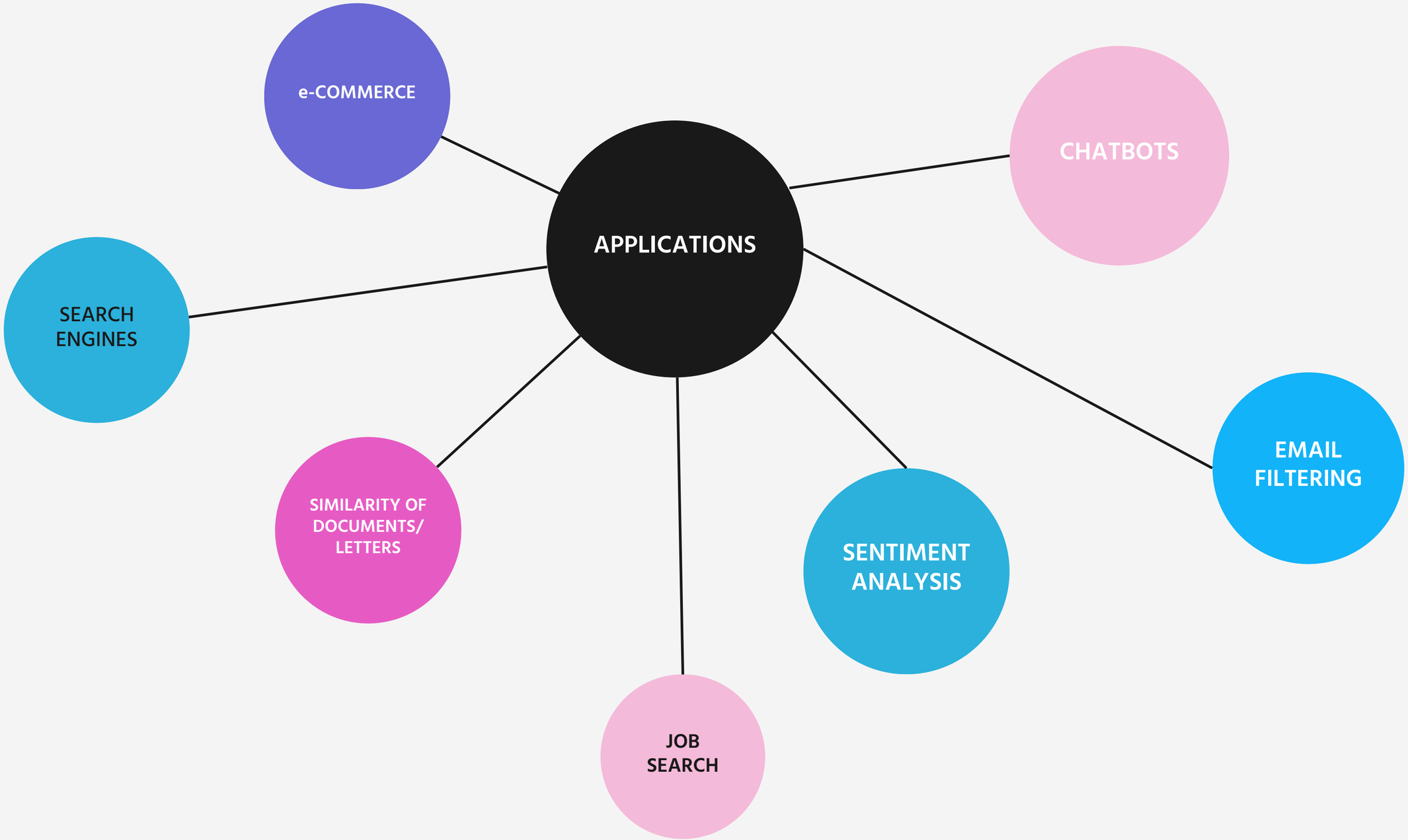
Text Similarity is the process of comparing a piece of text with another and finding the similarity between them. It’s basically about determining the degree of closeness of the text.



How does text similarity work?

We must first change the input text into a more machine-readable form by turning it into embeddings. These embeddings are then transformed into vectors that the computer can understand in order to calculate the similarity.




APPLICATIONS
e-COMMERCE
CHATBOTS
SEARCH ENGINES
EMAIL FILTERING
SIMILARITY OF DOCUMENTS/ LETTERS
SENTIMENT ANALYSIS
JOB SEARCH

Arxiv Dataset

For nearly 30 years, ArXiv has served the public and research communities by providing open access to scholarly articles from a wide range of disciplines. This rich corpus of information offers significant, but sometimes overwhelming depth.

Word2Vec

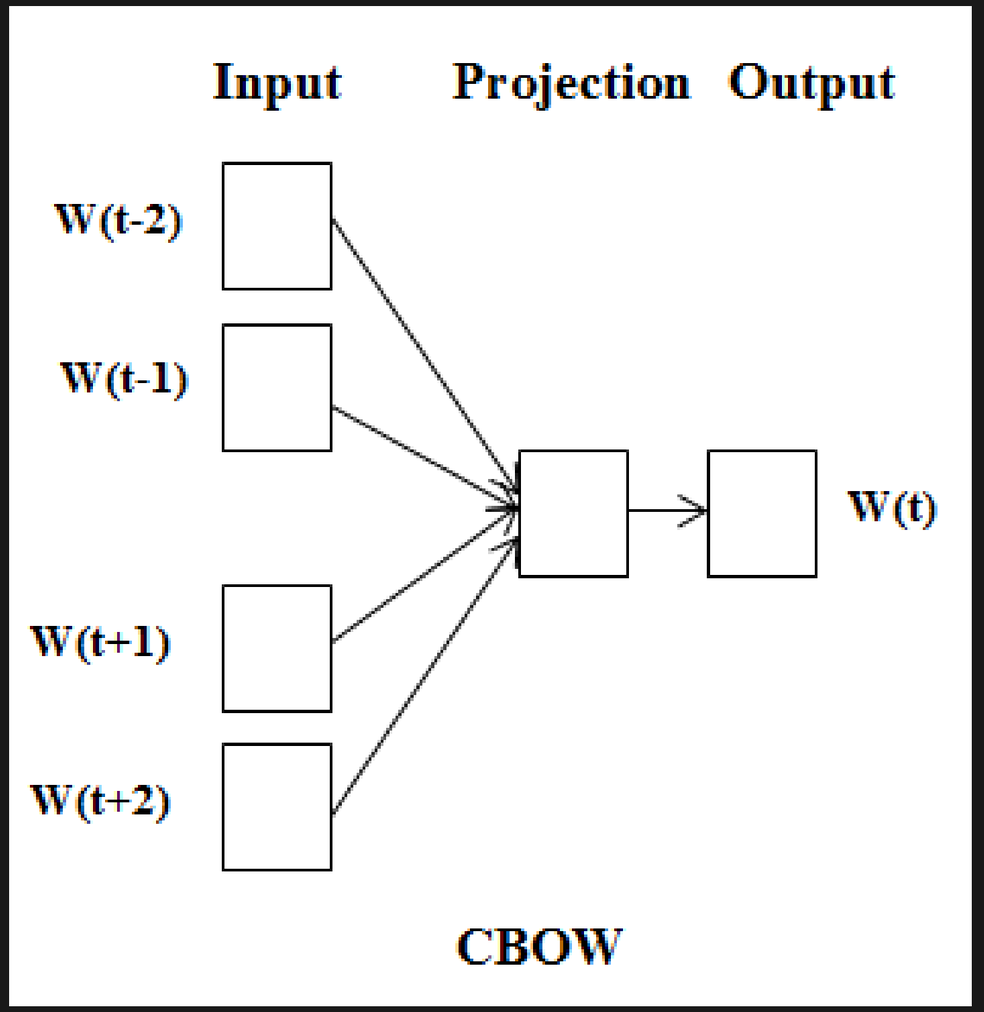
A predictive technique for creating word embeddings is Word2Vec. Word2Vec is a pre-trained two-layer neural network, in contrast to other techniques that needed to be "trained" on the working corpus.

It uses the text corpus as input and produces a collection of feature vectors that represent the words in the corpus. One of two neural network-based techniques is utilized:

- Skip-Gram
- Continuous Bag of Words (CBOW)

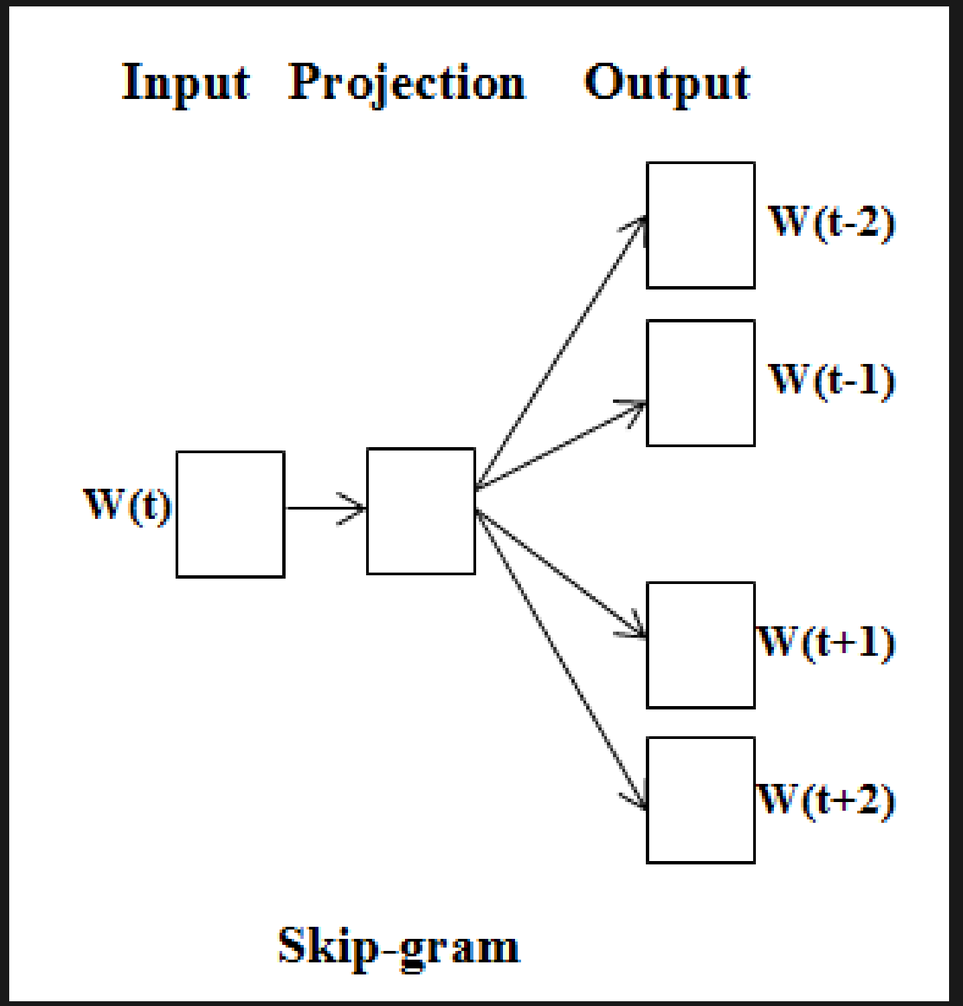
Continuous Bag Of Words

Takes the context of each word as the input and tries to predict the word corresponding to the context. Here, context simply means the surrounding words.



Skip-Gram Model

The Skip-gram model learns the vector representation of the target word as it predicts the words in the context. The context words are used to create representations.



Code-Snippets

```
w2v_model = Word2Vec(min_count=15,  # min_count=20,
                     window=5,  # window=2,
                     size=300,
                     sample=6e-5,
                     alpha=0.03,
                     min_alpha=0.0007,
                     negative=20,
                     workers=cores-1)
```

```
w2v_model.wv.most_similar(positive=["ai"])
```

```
[('artificial_intelligence', 0.689173698425293),
 ('ethic', 0.5939954519271851),
 ('agi', 0.5767300128936768),
 ('intelligent_agent', 0.5041818618774414),
 ('artificial_life', 0.5033911466598511),
 ('cognitive_science', 0.5016610622406006),
 ('superhuman', 0.5014196634292603),
 ('ethical', 0.4807817339897156),
 ('intelligence', 0.4680931568145752),
 ('assistant', 0.46774715185165405)]
```

LINK for the complete code

THANK YOU