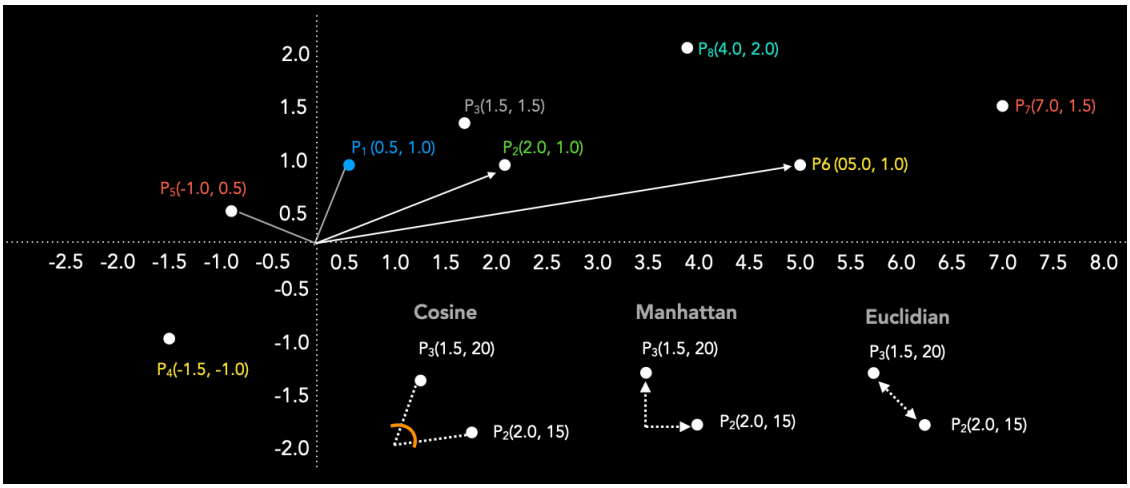
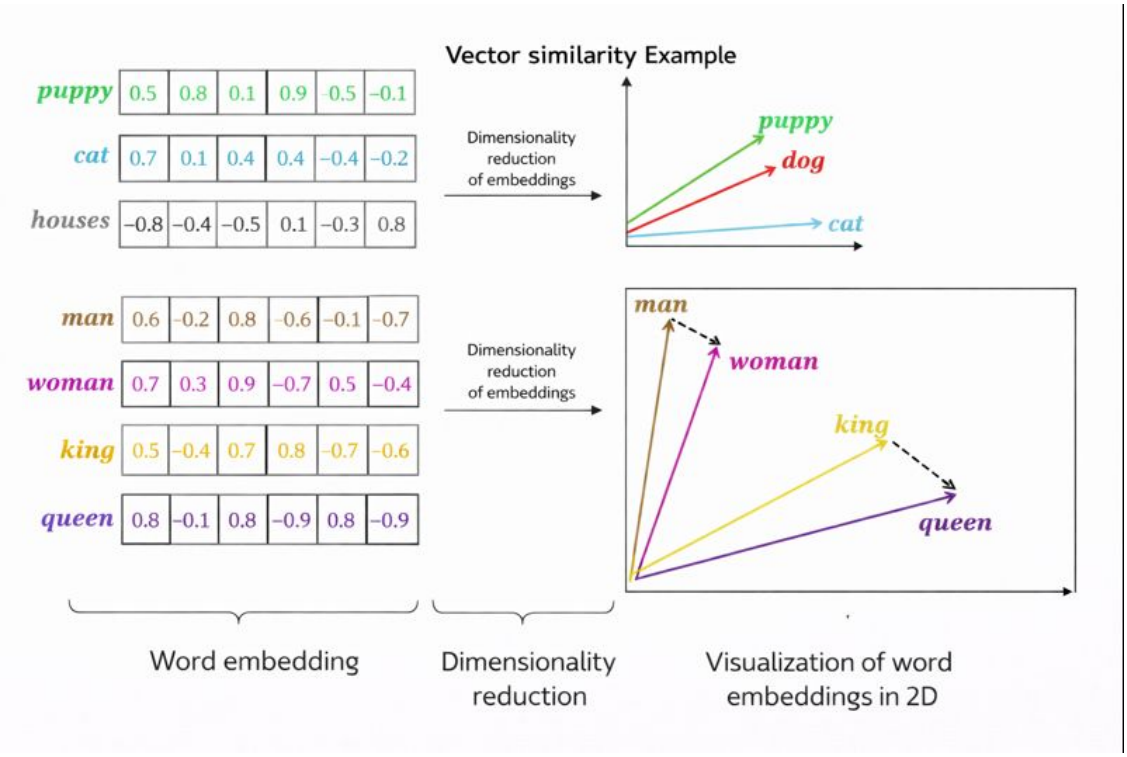
05 Vector Similarity

December 20, 2025

1 Vector Similarity

- **Measures Textual Relatedness**: Similarity in NLP quantifies how closely related two pieces of text are, based on factors like words choice, context, and meaning.
- **Used in Various Applications**: It's Crucial for tasks such as document clustering, recommendations systems, and paraphrase detection, where understanding the degree of similarity between texts is essential.
- **Different Methods**: Similarity can be measured using various approaches, from simple techniques like cosine similarity on word vectors to more advanced methods involving contextual embeddings from models like BERT.





[1]:
```
import spacy
from prettytable import PrettyTable
from spacy import displacy
```

[2]:
```
nlp = spacy.load('en_core_web_lg')
```

[3]:
```
doc = nlp('cat')
vec1 = doc.vector
```

[4]:
```
doc.vector.shape
```

[4]: (300,)

[5]:
```
# doc.vector
```

[6]:
```
doc = nlp('dog')
vec2 = doc.vector
```

[7]:
```
doc.vector.shape
```

[7]: (300,)

[8]:
```
from sklearn.metrics.pairwise import cosine_similarity

sim = cosine_similarity([vec1], [vec2])
print(sim)
```

```
[[0.80168545]]
```

[9]:
```
words = ['dog', 'cat', 'man', 'woman', 'king', 'queen', 'houses']
table = PrettyTable(field_names=['word 1', 'word 2', 'score'])
```

[10]:
```
from itertools import combinations

list(combinations(words, 2))
```

[10]:
```
[('dog', 'cat'),
 ('dog', 'man'),
 ('dog', 'woman'),
 ('dog', 'king'),
 ('dog', 'queen'),
 ('dog', 'houses'),
 ('cat', 'man'),
 ('cat', 'woman'),
 ('cat', 'king'),
 ('cat', 'queen'),
 ('cat', 'houses'),
 ('man', 'woman'),
 ('man', 'king'),
 ('man', 'queen'),
 ('man', 'houses'),
 ('woman', 'king'),
 ('woman', 'queen'),
 ('woman', 'houses'),
 ('king', 'queen'),
 ('king', 'houses'),
 ('queen', 'houses')]
```

[11]:
```
words = ['dog', 'cat', 'man', 'woman', 'king', 'queen', 'houses']
table = PrettyTable(field_names=['word 1', 'word 2', 'score'])
for word1, word2 in combinations(words, 2):
        token1 = nlp(word1)
        token2 = nlp(word2)

        sim = token1.similarity(token2)
        table.add_row([word1, word2, f"{sim:.2f}"])

print(table)
```

```
+--------+--------+-------+
| word 1 | word 2 | score |
+--------+--------+-------+
|  dog   |  cat   |  0.80 |
|  dog   |  man   |  0.42 |
|  dog   | woman  |  0.39 |
```

dog	king	0.25
dog	queen	0.23
dog	houses	0.23
cat	man	0.36
cat	woman	0.36
cat	king	0.28
cat	queen	0.31
cat	houses	0.20
man	woman	0.74
man	king	0.41
man	queen	0.27
man	houses	0.24
woman	king	0.27
woman	queen	0.41
woman	houses	0.21
king	queen	0.73
king	houses	0.21
queen	houses	0.21