

## Lesson 39 Summary: Word Embeddings

Data Science with Python – Key Concepts

Data Science Program

## Word Embeddings: Dense Representations

### Why Embeddings?

Semantic meaning  
Dense vectors  
Similar = close

### Word2Vec

CBOW / Skip-gram  
Context windows  
Google 2013

### GloVe

Co-occurrence  
Matrix factorization  
Stanford 2014

### Properties

king - man + woman  
= queen (analogy)  
Cosine similarity

### Pre-trained

Wikipedia corpus  
Transfer learning  
Fine-tune domain

### Finance Apps

FinBERT  
Sentiment analysis  
Document similarity

---

Dense vectors capture semantic meaning

## Limitations of BOW/TF-IDF:

- **Sparse:** High-dimensional, mostly zeros
- **No semantics:** “good” and “great” unrelated
- **Solution:** Dense, low-dimensional vectors

---

Similar words have similar vectors

## Learn from context:

- **CBOW:** Predict word from context
- **Skip-gram:** Predict context from word
- **Result:** 100-300 dim vectors

---

Google, 2013 - still widely used

## Famous analogy:

- **king - man + woman = queen**
- **Paris - France + Germany = Berlin**
- **Similarity:** Cosine distance

---

Embeddings encode relationships

## Transfer learning for NLP:

- **GloVe**: Wikipedia, Common Crawl
- **Word2Vec**: Google News
- **FastText**: Subword information

---

Fine-tune on domain-specific data

### Essential Commands:

Task	Code
Load GloVe	<code>gensim.downloader.load('glove-wiki')</code>
Get vector	<code>model['word']</code>
Similarity	<code>model.most_similar('word')</code>
Analogy	<code>model.most_similar(pos=[a,b], neg=[c])</code>

---

Embeddings are foundation for modern NLP