COMP3220 — Document Processing and Semantic Technologies

Week 05 Lecture 1: Word Embeddings

Diego Mollá

Department of Computer Science
Macquarie University

COMP3220 2022H1

Programme

1. Challenges of Text for Machine Learning
2. Word Embeddings

Reading

- Deep Learning book (2nd edition), Chapter 11.
- Jurafsky & Martin, Chapter 6.

Programme

Words as Arbitrary Symbols

- Words are encoded as arbitrary symbols.
- Within one language there is no clear correspondence between a word symbol and its meaning.
  - "dig" vs. "dog"
  - "car" vs. "automobile"
- Different languages may use different representations of the same word.

https://en.wikipedia.org/wiki/File:Hello_in_different_languages_word_cloud.jpeg

Ambiguities Everywhere

Language features ambiguity at multiple levels.

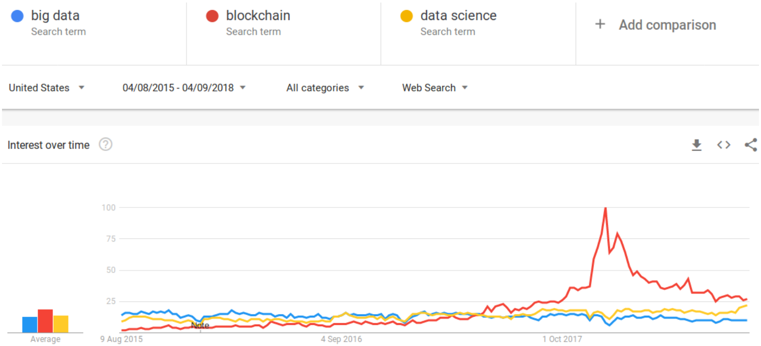
Lexical Ambiguity

Example from Google's dictionary:

- bank (n): the land alongside or sloping down a river or lake.
- bank (n): financial establishment that uses money deposited by customers for investment, ...
- bank (v): form in to a mass or mound.
- bank (v): build (a road, railway, or sports track) higher at the outer edge of a bend to facilitate fast cornering.
- ...

So many words!

- Any language features a large number of distinct words.
- New words are coined.
- Words change their use in time.
- There are also names, numbers, dates... an infinite number.



https://trends.google.com

Long-distance Dependencies

- Sentences are sequences of words.
- Words close in the sentence are often related.
- But, sometimes, there are relations between words far apart.

grammatical: "The man living upstairs ... very cheerful"
"The people living upstairs ... very cheerful"

knowledge: "I was born in France and I speak fluent ..."

reference: "I bought a book from the shopkeeper and I liked it"

Long-distance Dependencies

- Sentences are sequences of words.
- Words close in the sentence are often related.
- But, sometimes, there are relations between words far apart.

grammatical:	"The man living upstairs is very cheerful"
	"The people living upstairs are very cheerful"
knowledge:	"I was born in France and I speak fluent ..."
reference:	"I bought a book from the shopkeeper and I liked it"

Long-distance Dependencies

- Sentences are sequences of words.
- Words close in the sentence are often related.
- But, sometimes, there are relations between words far apart.

grammatical:	"The man living upstairs is very cheerful"
	"The people living upstairs are very cheerful"
knowledge:	"I was born in France and I speak fluent French"
reference:	"I bought a book from the shopkeeper and I liked it"

Long-distance Dependencies

- Sentences are sequences of words.
- Words close in the sentence are often related.
- But, sometimes, there are relations between words far apart.

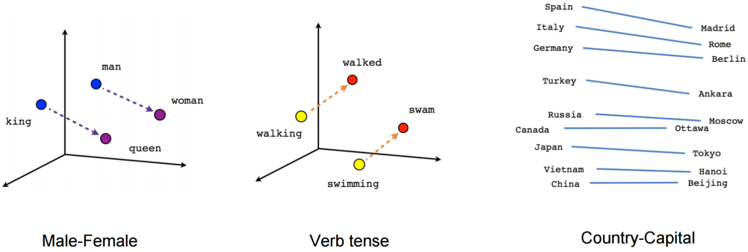
grammatical:	"The man living upstairs is very cheerful"
	"The people living upstairs are very cheerful"
knowledge:	"I was born in France and I speak fluent French"
reference:	"I bought a book from the shopkeeper and I liked it"

Programme

1. Challenges of Text for Machine Learning
2. Word Embeddings

Word Embeddings

- First introduced in 2013, nowadays is one of the most common ingredients in text processing systems.
- Word embeddings squarely aim at addressing the issue of representing words as continuous vectors of integers.
- Words with similar context are mapped to similar vectors.
- Embeddings are learnt using large, unlabelled training data.



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

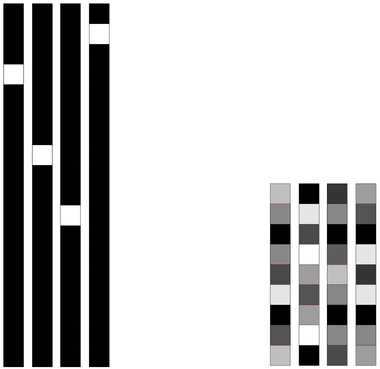
One-hot vs. word embeddings

One-hot

- Sparse
- Binary values (typically)
- High-dimensional
- Hard-coded

Word embeddings

- Dense
- Continous values
- Lower-dimensional
- Learned from data



Two Ways to Obtain Word Embeddings

1. Learn the word embeddings jointly with the task you care about (e.g. document classification).
2. Use pre-trained word embeddings.

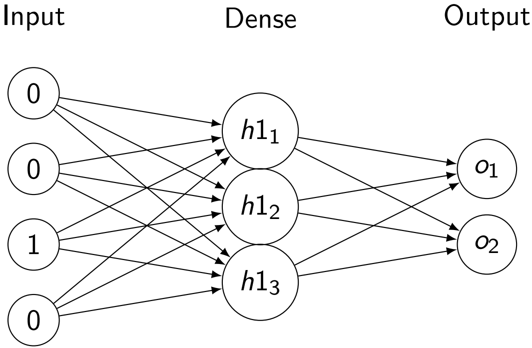
Learning Word Embeddings

- You can add a dense layer as the first layer of your network and let the system learn the optimal weights.
- This approach is so useful and common that many deep learning frameworks define an “embedding” layer that facilitates this.
- The input to the “embedding” layer is the word index.
- The output is the word embedding.

Embedding Layer as a Dense Layer

The input of the dense layer is the one-hot encoding of the word

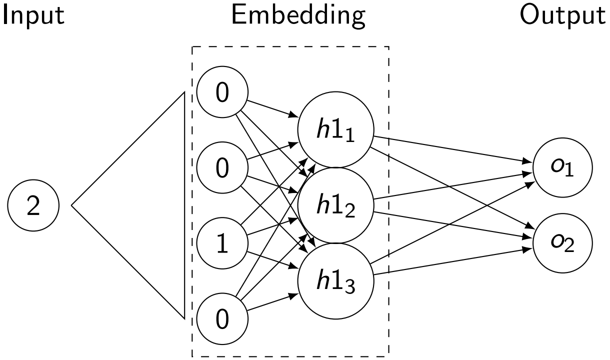
A Dense Layer



Embedding Layer in Keras

The input of a Keras embedding layer is a sequence of word indices which will be internally converted into their one-hot representations, followed by the dense layer.

A Keras Embedding Layer (for one word)



Processing Sequences of Words in Keras

- The input of a Keras embedding layers is a sequence of words.
- The output is a sequence of word embeddings.
- Since the layer will process a batch of samples at a time, each batch must have sequences with the same numbers of words.
- Keras provides a way to trim sequences of words or pad them to adjust the sequence length: pad_sequences.

Using pre-trained word embeddings

The Problem: Data Sparsity

- Sometimes we have so little training data that many words are poorly represented.
- Often, words in the test data do not occur in the training data.
- For these unseen words we would not be able to learn the embeddings.

A Solution: Pre-training

- Several people have computed word embeddings for large vocabularies using large data sets.
- We can then use these pre-trained embeddings to map from the word index to the word embedding.

Using Word Embeddings in Keras

- The following notebook is based on the jupyter notebooks provided by the Deep Learning book: https://github.com/fchollet/deep-learning-with-python-notebooks
  - Using word embeddings.
- The notebook illustrates how you can use an embeddings layer for text classification, and how to load pre-trained word embeddings.
- This notebook is important because it also illustrates Keras' text tokenisation techniques.

Final Note: Contextualised Word Embeddings!

Recent research devised a way to produce context-dependent word embeddings. The resulting systems are beating state of the art in many applications!

http://jalammar.github.io/illustrated-bert/

Take-home Messages

1. Explain some of the fundamental challenges that plain text represents to machine learning.
2. Apply word embeddings in deep learning.

What's Next

Week 6

- Processing text sequences.
- Reading: Deep learning book (2nd edition), Chapter 11.
- Reading: Jurafsky & Martin, Chapter 9.