# What is a Word Embedding

A word embedding is a numeric array representation of an arbitrary collection of words. This representation is can then be used by downstream machine learning models, whose inputs are typically numeric and of a fixed size.

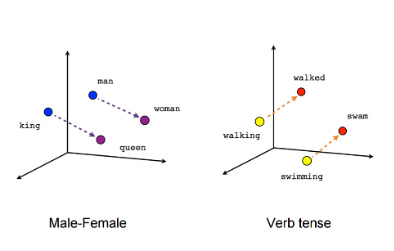
# Different Types of Word Embeddings

Generated in an unsupervised fashion by feeding a model a large quantity of text input and allowing it to create a reusable mapping function

## Word2Vec

In [2], Mikolov et al. propose two model architectures for producing continuous vector representations of words from a corpus of text. These models are simpler than the existing NN and RNN based language models of the time and achieved a far lower computational complexity without sacrificing performance.

Interestingly, the generated word vectors contain latent information about the relationships between words. This was recognised by the same authors in an earlier paper [8] and the Word2Vec models generate vectors which aim to maximise this property. The relationships manifest as linear regularities between word vectors and can be explored by simple vector arithmetic. The classic example being vector(King) – vector(male) + vector(female) = vector(Queen)



Vector arithmetic examples [7]

The first architecture, Continuous Bag of Words

Word2Vec is essentially a shallow feedforward model, this improves training time, but also limits the amount of information the model can capture. Another limitation is that it isn’t contextually aware. This means a word2vec model will generate the same vector for the same word, regardless of whether it has different meanings in different sentences. [7]

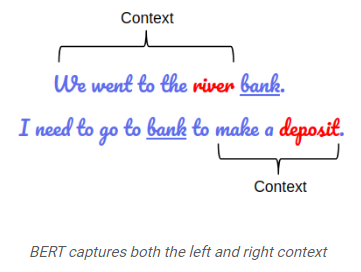
* “Bag of words”
  + Order of context words does not influence vector prediction
* Model Architectures
  + Continuous Bag of Words (CBOW)
  + Continuous Skip-Gram (skip grams)
    - Uses current word to predict surrounding words
    - Weighs nearby words more heavily than distant words
    - Slower but better job on infrequent words

## ELMo

* Developed to address problem of Polysemy – one word having multiple meanings based on context
* Takes account of position of word in sentence
* One of the first to use: *Pretraining + Fine Tuning = Transfer Learning*
* LSTM Architecture
* Attention Mechanism
* Need to load and use the model for downstream processes, as this is what captures context
* Word based – takes words as input and outputs word embeddings

## BERT

BERT stands for Bi-directional Encoder Representations from Transformers. It improves on previous work where representations are built using trained transformers by introducing the concept of bi-directionality. This essentially allows the BERT model to incorporate context from both directions [3]. Why is bi-directional context important? Here’s an image from [7] showing how the meaning of the word bank can be affected by context to the left or right of the word



BERT captures both the left and right context [7]

BERT Models achieves bi-directionality by unsupervised pre-training using two different approaches. First it randomly masks 15% of the words and tasks the model to predict the hidden words based on context. Second it trains the model to predict the next sentence.

Following the pre-training phase the model will be fine-tuned for a specific by taking the pre-trained model and applying further training, this time supervised, with task specific data and labels.

* Context aware
* Different layers generate different vectors
* Sequence model
* Need to load and use the model for downstream processes, as this is what captures context
* Represents words as subwords, vocabulary is region of 30000 for millions of unique words. This is smaller than ELMo.
* Balance between character based and word based representation
* Avoidance of Out of Vocabulary cases (other models suffer from this)

## Summary table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Context Aware | Order Aware | Trained Model Required | Representations |  |  |
| Word2Vec | No | No | No | Words |  |  |
| ELMo | Yes | Uni-directional | Yes | Words |  |  |
| BERT | Yes | Bi-directional | Yes | Sub-words |  |  |

# References

1. Brief review of word embedding families, Article, 23 March 2019, <https://mc.ai/brief-review-of-word-embedding-families-2019/>
2. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. ICLR Workshop,
3. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding, *where*
4. Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report, OpenAI.
5. Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Settlemoyer. 2018. Deep Contextualized Word Representations, *where?*
6. Anirudh S. 2019. Word Embedding Using BERT In Python, <https://towardsdatascience.com/word-embedding-using-bert-in-python-dd5a86c00342>. TowardsDataScience.com
7. Mohid Sanad Zaki Rizvi. 2019. <https://www.analyticsvidhya.com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework/>. AnalyticsVidhya.com
8. T. Mikolov, W.T. Yih, G. Zweig. Linguistic Regularities in Continuous Space Word Representations. NAACL HLT 2013