# What is a Word Embedding

A word embedding is a numeric array representation of an arbitrary collection of words. This representation is required 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.

The first architecture, Continuous Bag of Words

Word2Vec is a tool designed by Google to take in a text corpus and produce word vectors. The word vectors can be used as features in downstream machine learning models.

* “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
* Does not take into account the position of words in a sentence. Word2Vec
* Can just use the vector for downstream processes

## ELMo

* Takes account of position of word in sentence
* 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]

BERT Models are first pre-trained in an unsupervised fashion by randomly masking around 15% of the words and tasking the model to predict the hidden words based on context. Following this 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 | Order Aware | Trained Model Required | Representations |  |  |
| Word2Vec | No | No | Words |  |  |
| ELMo | Yes | Yes | Words |  |  |
| BERT | Yes | 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>. Towards Data Science