Three Approaches to Word Embeddings

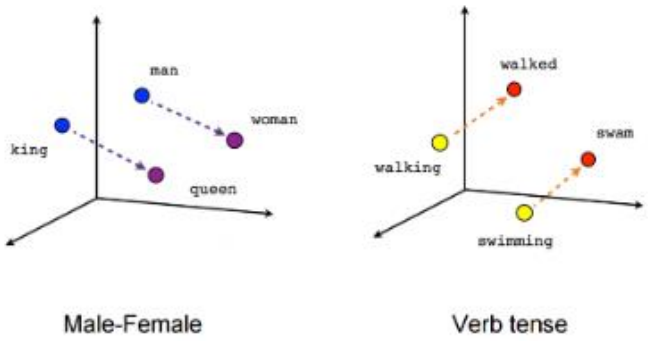
MICHAEL DOWD, EDINBURGH NAPIER NUMBER: 40451589

Word embeddings are a powerful tool in natural language processing. They are essentially a multi-dimensional numeric representation of a word which, once generated, encapsulates a trained model's entire understanding of a word in a machine-readable format, meaning they can easily be used as input into downstream models for more specific tasks such as classification or sentiment analysis.

Word2Vec

Mikolov et al. (January, 2013) proposed two techniques for producing word embeddings in an unsupervised fashion from a corpus of text. These models were simpler than existing NN and RNN based language models and achieved a far lower computational complexity without sacrificing performance.

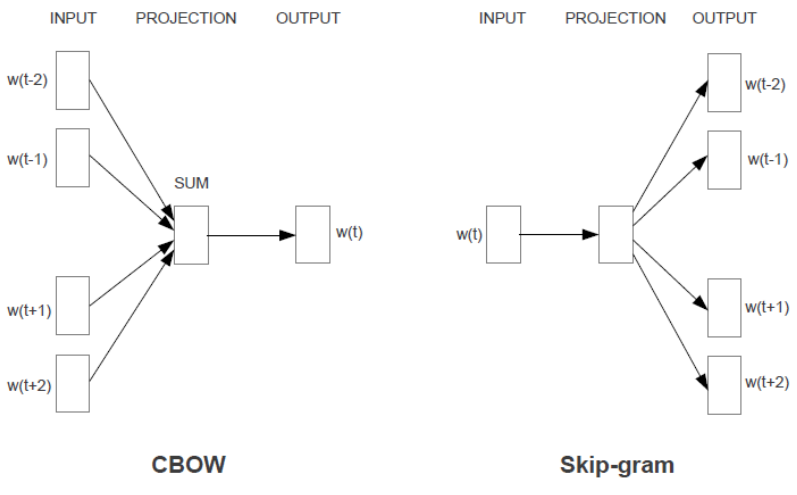
Interestingly, the generated word vectors also contained latent information about the relationships between words. This was recognised by the same authors in another paper (Mikolov et al. June, 2013) and the Word2Vec models generate vectors which aim to maximise this property. The relationships manifest 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 (Mikolov et al. June, 2013)

The proposed Word2Vec embeddings were generated with two different approaches. For the first, a shallow feedforward neural network was trained to predict a word based on the four preceding and subsequent words. The authors call this a continuous bag-of-words model (CBOW) as it's similar to a standard bag-of-words approach in that the order of words is not used. The second model trained by the authors, called skip-gram, is like CBOW in reverse, it uses the target word to try to predict preceding and subsequent words.

In both cases, the prediction output of these models is not the end-goal, rather, it's the activations of the trained model's projection layer that are extracted for each word in the vocabulary, these are the word embeddings.

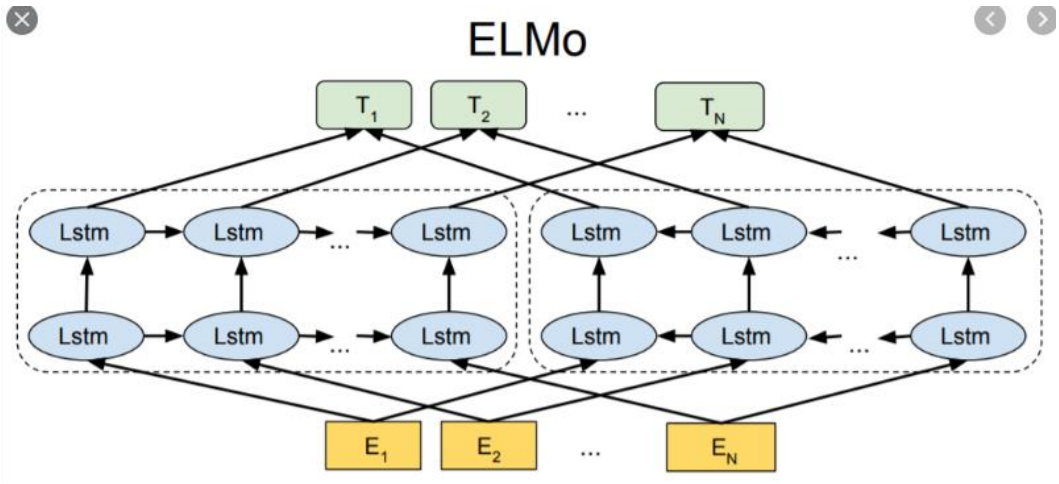


CBOW and Skip-gram model architecture summary (Mikolov et al, January 2013)

ELMo

ELMo (Embeddings from Language Model), aims to capture not only the relationships between words (as per word2vec), but also how word use varies across contexts (Peters et al, 2018). Therefore, the generated word embeddings are functions of an entire sentence.

They achieve this by using stacked LSTM layers. Long Short Term Memory nodes are used in neural networks to boost retention of information over longer input sequences. This is done by maintaining a hidden state, which is activated and added to their output by a logic gate. In the case of a language model trying to predict the next word, the model will have better knowledge about the entire sentence up to that point and may make better predictions.



ELMo architecture summary (Devlin et al. 2019)

ELMo uses a pair of stacked LSTM layers going in different directions to achieve a measure of bi-directionality and capture the context of a word within a specific sentence. The table below clearly illustrates why this is an important step:

	Source	Nearest Neighbors
GloVe	play	playing, game, games, played, players, plays, player, Play, football, multiplayer
biLM	Chico Ruiz made a spectacular play on Alusik 's grounder {...}	Kieffer , the only junior in the group , was commended for his ability to hit in the clutch , as well as his all-round excellent play .
	Olivia De Havilland signed to do a Broadway play for Garson {...}	{...} they were actors who had been handed fat roles in a successful play , and had talent enough to fill the roles competently , with nice understatement .

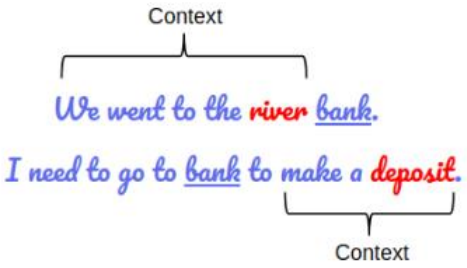
ELMo (or biLM) compared to GloVe, an earlier language model (Peters et al, 2018)

Here we see that a prior language model, GloVe, had no concept of the context of a word in a sentence – the word 'play' had a single vector, whose nearest neighbours were words with similar vectors. ELMo (or biLM), on the other hand, must include surrounding words in the input and output as the meaning of the word 'play' can only really be determined by the sentence it's used in. What's more the two different meanings are clearly identified by the model as shown by the nearest neighbour sentences.

Where word2vec embeddings are static and easily downloaded as a dictionary, ELMo comes as a pre-trained model, which generates embeddings based on a full sentences. Users are expected to download the weights and use the model output as a feature in their own downstream models. This approach is called feature-based transfer learning.

BERT

BERT (Bi-directional Encoder Representations from Transformers) improves on previous work by introducing the concept of *deep* bi-directionality. The model was created to address the fact that models up until that point where either contextless (word2vec), uni-directional or shallowly bi-directional, as per ELMo (Devlin et al, 2019). Deep bi-directionality allows the BERT model to incorporate context from both directions in its understanding of any given word. The image below shows how the meaning of the word bank can be affected by context to the left or right of the word

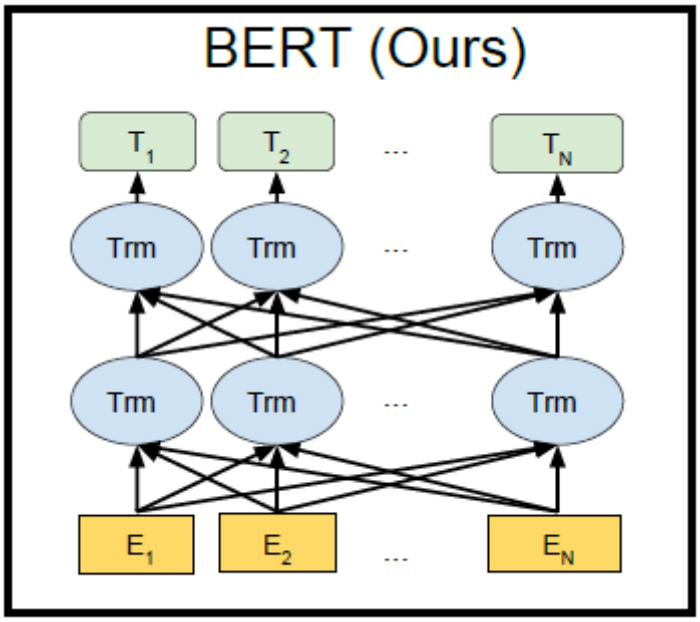


BERT captures both the left and right context (Sanad Zaki Rizvi, M., 2019)

The BERT model achieves bi-directional and cross-sentence context using two simple training methods.

1. It randomly masks 15% of the words and trains the model to predict the hidden words based on the other words in the sentence. This builds knowledge of word context within a sentence
2. It trains the model to predict whether sentence pairs belong together or not. This trains the model to understand context which spans sentences, which might be especially important for question answering tasks.

The concepts behind BERT are conceptually simple yet have yielded excellent empirical results, including new state of the art scores on eleven standard natural language processing tasks (Devlin et al, 2019).



BERT architecture summary (Devlin et al, 2019)

Due to its generic neural network architecture, BERT can be fine-tuned for a particular problem. This is where the entire model is retrained for a small number of iterations, thus allowing the weights to change slightly to better fit the task at hand. The authors also compare a feature-based approach and achieve a very close F1 score by concatenating the representations from the last four layers of BERT and feeding this into a two layer biLSTM model. Both approaches beat previous state of the art models such as ELMo.

References

Mikolov, T., Chen, K., Corrado, G., Dean, J. (January 2013). Efficient estimation of word representations in vector space. *Proceedings of the International Conference on Learning Representations (ICLR 2013)*

Devlin J., Chang, M., Lee, K. Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT 2019,* pages 4171–4186

Peters, M., Neumann, M., Iyver, M., Gardner, M., Clark C., Lee, K., Settlemoyer, L. (2018). Deep contextualized word representations. *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics,* pages 2227-2237

Mikolov, T., Yih, W.T., Zweig, G. (June, 2013). Linguistic regularities in continuous space word representations. *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* pages 746–751

Sanad Zaki Rizvi, M. (2019). *Demystifying BERT: A Comprehensive Guide to the Groundbreaking NLP Framework.* Retrieved from: https://www.analyticsvidhya.com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework/