The Illustrated Word2vec

Muhammad Hashim Javed 21100271

1 Paper trivia

This article, written by Jay Alammar, illustrates the embedding of words to the vector space. Author uses illustrations for readers to make it easy to understand the concept.

2 Problem description

Computer cannot understand natural languages, i.e, English, they only understand number and learn the patterns from numbers. That's why for many problems like text prediction, recognition, translation and etc, it is necessary to represent a word as combination of numbers, i:e, vectors.
In many applications like Siri, Google Assistant, Alexa, Google Translate, or even smartphone keyboard with next-word prediction, word embedding is central technique.

3 Main techniques of the paper

'You shall know a word by the company it keeps'.
In many next word prediction applications like google keyboard, word embedding is key part of program. A very simple prediction program would be to randomly initiate embeddings and train embedding of a word according to next word in train data, i.e: for 'You' output should be 'shall'. This model seems simple but have many problems, uneffeciency, huge complexity. Authur discusses the improvements on this models as follow.

- **Look one way :** Look one way is very similar to above model. In this we train embeddings such that in our window last word is output and other are input, i.e: For window of size three 'You' and 'shall' will be trained such that they output 'know'.
- **Look both ways :** In text, words also depend on context of coming words it is not effecient to only train on previous word. For example, For window size three, 'You' and 'know' will be input and 'shall' will be output.
- **Skipgram :** Above both techniques, given context predict the word, but we can also predict context given word, this method is called skipgram. e.g: For window size three 'shall' will be input and 'You' and 'know' will be output.

Training Process : In above method, we produce effecient embedding of words through training, and during prediction it computes similarity of word with all words in vocabulary and selects the word with highest rating. This is computationally very exhaustive, given the size of vocabulary in millions in daily life. For that we use another method of training, in which we learn embedding such that if two words are related they should output 1 and else 0. Here another problem arise as we only have data about similar

word.

Negative Sampling : If we only train on positive samples, our program will train in one side only, to counter that we introduce fake samples, with label 0, they are called negative sample. Negative sampling is vast field but author doesn't discuss that.

Window Size : Window is size is one of parameter in this process, which can vary from 5 ot 50, from prdicting similar words to related words. **Negative Samples :** For one positive there could be negative samples from 2 to 5, for large enough dataset.