CS224N Lecture-2

In the first half, techniques to improve word2vec were discussed like negative sampling etc.

The second half discussed the GLoVe word vectors. The main problem with word2vec is that it ignores the use of global statistics and only considers the local context. Hence if "the" and "cat" frequently occur in each other's context, then it won't know whether it is because "the" and "cat" are highly related in terms of each other's meaning (like "New" "Delhi") or is it because "the" appears quite frequently but has nothing to do with the sense of "cat".

Only using the global statistical approach also doesn't seem to work that well when evaluated on tasks like the analogy.
GLoVe incorporates both these techniques for learning the word vectors. The key idea is that the ratio of probabilities of co-occurrence could be a strong basis of learning word meanings, which was demonstrated by the example of "steam" and "ice".

For learning GLoVe representation, first, we construct the co-occurrence matrix. This can be computationally expensive, but it only needs to be done only once. Next, we build the function, which we would like to minimize.

The authors argued that their function, in essence, is very similar to the service used in word2vec. In the paper, it is shown that GLoVe outperforms word2vec. Still, the papers after GLoVe gave the empirical conclusion that there is no real difference in performance when using GLoVe or word2vec for downstream tasks like Question Answering.

The lecture finally discusses evaluations techniques which could be used to quantify the effectiveness of word vectors.

There are two kinds: intrinsic evaluation which means evaluating on sub-task like the analogy. These tasks are relatively less costly and typically fast to compute. One essential requirement, for it to be successful, is that performance on subtask should have a positive correlation with the final task.

Extrinsic evaluation, on the other hand, does the evaluation on the final tasks like question answering. This is generally expensive, and typically we do not always know whether the increase or decrease in performance is due to quality of word vectors.