Lecture 3 | GloVe: Global Vectors for Word Representation

# Window based co-occurrence matrix (example)

# Window length 1 (more common: 5 – 10)

# Symmetric

# Example corpus:

# I like deep learning

# I like NLP

# I enjoy flying

# 

# Problems:

# The matrix increases in size with vocabulary (generally 10,000+)

# Very high dimensional, therefore face with sparsity issues and requires a lot of storage space

# Solutions:

# The idea is to store “most” of the important information in a fixed, small number of dimensions: a dense vector

# Usually 25 – 1000 dimensions, similar to word2vec

# How to reduce dimensionality?

# Use singular value decomposition (SVD) – factorisation of real/complex matrix

# 

# Problems with SVD:

# Computational cost scales quadratically for n x m matrix – O(mn^2) flops when n < m, therefore bad for millions of words or documents

# Hard to incorporate new words/documents

# 

# Combining the best of both techniques, count based and direct prediction, we have the GloVe model

# 

# Fast training

# Scalable to huge corpora

# Good performance with small corpus and vectors

# With the GloVe model, we end up with U and V from all the vectors u and v. They both capture similar co-occurrence information. The best solution is to sum them up, where X\_final = U + V

# Evaluate word vectors

# Intrinsic

# Evaluation on a specific/intermediate subtask

# Fast to compute

# Helps to understand the system

# Not clear if it’s really helpful unless correlation to real task is established

# Extrinsic

# Evaluation on a real task

# Can take a longer time to compute accuracy

# Unclear if the subsystem is the real problem of it’s the interaction of other subsystems

# If you replace one subsystem with another and it improves accuracy, great!

# Intrinsic word vector evaluation

# Word vector analogies

# Man:woman :: king:?

# Evaluate word vectors by how well their cosine distance, after addition, captures intuitive semantic and syntactic analogy questions

# What if the information is there but not linear?

# 

# The table above shows that:

# Increasing dimension doesn’t always lead to better results

# More data (size) and better-quality data will lead to better results (just like most deep learning models)

# The best dimensions seem to be around 300 and window size of 8 around each center word is good for Glove vectors. Asymmetric context (only words to the left) are not as good so it might be best to stick with symmetric context

# Extrinsic word vector evaluation

# Good word vectors should help directly with named entity recognition (NER), therefore, finding a person, organisation or location