Question 1

Word2Vec:

It is an algorithm that translates a corpus of words from a written/spoken language to a vector space, where each word is assigned a unique vector. The vector space is constructed such that the geometric/euclidean properties of vectors are still valid on the embeddings. Simply put, similar words appear closer to each other. Sometimes, euclidean operations such as addition, subtraction may also yield meaningful results. For instance, King-Man+Woman may yield Queen on a well-trained embedding. We use the skip-gram model.

Skip-gram Model:

It is a shallow 2 layer neural network. It takes as input a target word and predicts surrounding context words, i.e. words closest to the target word in the vector space. The embeddings of the words are actually the weights to the only hidden layer in the network. After obtaining this layer, we may discard the rest of the model.

During the training process, the words would settle down into clusters. Similar words would be around each other. With more epochs, the clusters would become more separable.

Question 2

Vector adjustment:

We find the top 10 documents for each query using cosine similarity. We divide these into relevant and non-relevant using the ground truth and then apply the adjustment formula to the queries.

Query expansion:

After performing vector adjustment, we find all relevant documents to the query from the ground truth. Then, we select top 10 words from each document based on their TFIDF. Out of this pool of words, we again take the top 10. These words are used to update the query.

Instead of this approach, we could have compiled all words from all docs and then taken the top 10, however, that would have lead to significantly higher computation time.

Results:

Model/Iterations	3	5	10
Baseline	51.83		
Vector adjustment	75.84	75.98	75.99
Query expansion	79.01	79.09	79.14

As expected, query expansion outperforms vector adjustment, which outperforms the baseline cosine similarity result.

Query expansion works the best since we are adding terms corresponding to documents taken from the ground truth, thus it leads to higher similarity between the query and documents in the ground truth, and thus higher MAP.

Vector adjustment outperforms the baseline as we promote(add) terms corresponding to relevant documents and penalize(subtract) terms corresponding to irrelevant documents.