**FAQs and Resources**

**Topics: Introduction to Natural Language Processing**

1. **Why do we need word embeddings when we can encode each word with one-hot encoding?**One-hot encoding is an orthogonal representation i.e. all the vectors have no relation with among themselves. Each vector is independent of the other. Machine learning translation and Question-Answering tasks require words representation that is aware of the semantics of the word, therefore, word embeddings is a better option.
2. **What is negative sampling and why is it required?**Word2Vec model uses the softmax layer coupled with cross-entropy loss as an output. Consider the example where the vocabulary size is 10000 and the embedding dimension is 300. The hidden layer in the model will have 300 X 10000 = 3M weights. Updating each weight is computationally expensive and requires a large number of training examples. Negative sampling considers very few negative words( words that do not occur in the context of input word) and updates weights corresponding to those words. Effectively negative sampling updates a small number of weights, thereby increasing the speed of training significantly.
3. **How do word embeddings encode semantic information?**Word embeddings are learnt from the co-occurrence of words in the corpus. Word embeddings are dense vectors in high dimensional space. The training objective of Word2Vec is such that similarity between related words is maximized and between unrelated words is minimized, hence word embeddings have semantic information encoded in them.
4. **What different algorithms are available to generate word embeddings?**Mainly there are two algorithms used to generate word embeddings, namely Word2Vec and Glove. Both algorithms do not have different representation for ambiguous words. For e.g. words like “book” which can be both a noun and a verb have a single representation vector rather than two. Recently, there have been works like BERT that try to address this problem.
5. **What is the difference between Glove and Word2Vec?**Word2Vec uses classification as a proxy task to generate word embeddings. The weights of the output classification layer are the word embeddings corresponding to each word in the vocabulary. On the other hand, Glove reduces the dimension of the co-occurrence matrix using a reconstruction loss. The reduced matrix is the word embedding matrix.
6. **How can we train our own Word2Vec models?**Gensim provides an API to train your own Word2Vec models as well as use pre-trained models to get word embeddings. Most of the deep learning frameworks like and TensorFlow have embedding layers that one can add in the network to train their own Word2Vec. Refer the practice exercise for more details on the programming part.
7. **Should we always use pre-trained Word2Vec models?**Always using pre-trained Word2Vec might not give good results every time. Depending on what amount of dataset you have you can train your own model. Generally, when there is enough data, using an embedding initialized with pre-trained weights is a good practice.
8. **What dimension of word embeddings should we use?**Larger dimensions of word embeddings add a number of weights to the model, which in turn requires more data to train. Depending on the amount of dataset and the number of unique tokens in the dataset one can choose what dimension of embeddings should be used.