Natural language processing

Mias Ghantous – 213461692

Faisal Omari –

Section 1)

Part 1:

we implemented a function called which takes a sentence and make the tokens as required, how did we do it?

1. If we have at the end of the token then this a regular word if the first character is a Hebrew latter and the second to last.

2. if we have Hebrew latter at the end of the token and also at the beginning then this is also a regular word

3. anything else is not a Hebrew word that we should include.

Questions:

1. Min\_count: is the minimum count that would be included in the training, if it was big then we are going to learn on the most frequent words in the language, which leads to overfitting and if it were too small then we would learn on words that we don’t use them very often like names of countries which will include noise, we defined it to be in the 1 that means that every word in the corpus will be included but we would have put it bigger.

Window: is the max distance between the current word and the predict word in a sentence, small value leads to over fitting as we are not going to see a lot of repeated combinations, and not enough data on medium frequent combinations of words. And big value leads to a lot of noise as the not frequent combinations would appear in the training, we choose which we think is enough not too big and not too small.

Vector\_size: is the embedding vector size which tells how much does words are similar, too big of a number leads to overfitting because of the number of features, and too small would lead to underfitting because the feature would not be representative enough, we choose 100 which is okay.

1. The problem is that we work with Hebrew and in Hebrew we have a lot of combined words that make one meaning like "בית ספר" and our corpus is not big enough to cope with this and also in Hebrew there is tokens that have many words like "וכשמהבית" which means 'and from the hose' and we created the corpus in a way that all of these words are just one token also we have words like "בצל" which can mean "in the shadow" or "onion" 2 words that have totally different meanings and in the corpus we don’t attempt to fix this.

Section 2)

Part 1. Implementation in function , we put all the words in a list and iterate throw the list.

Part 2. Implementation in which takes the model also the sentences and return the a dictionary from the index of the corpus to the average embedding of each sentence

Part 3. We search the corpus for sentences that potentially we can use we choose them to be not that long has very common words in the language also sentences that can be said a lot in protocols.

Part 4. Our way is to first try the same word if any of the first 3 is ok then take it

Else try a similar word or a word that fits the sentence if any of the first 3 are ok then take it.

Else, try more than one word that are related.

For each word we have tried a lot and some of them got nice results and some of them we didn’t get useful results.