1. **Specific problem definition** :

In this homework, the task is to find couples of words in the form (D,E) where D is an intensifier adverb while E is an adjective and the function of D is to increase the intensity the expression of E. One more criteria is that D is not general-purpose and only modifies to a restricted group of adjectives E.

1. **Discuss the result:**
2. **Idea for the code to prove the reasonability of my result:**

**I**. There are 2 main issues to deal with: finding adverbs that used as intensifier and among these adverbs, choosing ones that just modify to a restrictive group. My solution for the first issue: in the wordnet corpus in nltk, each synset contains a group of words with the same meaning and type, and the common meaning for these words can be gotten through synset.definition(). I create a list of about 20 common intensifiers that I know and collect the definitions of them into a list of strings, and then find the most common words in these strings. And the result : word ‘intensifier’ , ’intensifiers’, ‘degree’ are 3 most common words in definition strings, for example in wordnet, definition for the Synset(‘very.r.01’) is “used as intensifiers; …”. However, one important thing that i realize: if the definition string contains ‘intensifier’ or ‘intensifiers’, then it must be an intensifier, but if the definition string contains ‘degree’ then it can be an intensifier or a manner adverb. To collect intensifiers, I create a list choose\_intensifer=[ ] and then go through all synsets in the wordnet, if the definition part of that synset contains ‘degree’ or ‘intensifier’ or ‘intensifiers’ then append all synonyms in that synset to the list choose\_intensifier. As mentioned above, this list contains intensifiers but also maybe some manner adverbs, so I have to score these adverbs. The score is as follow: with each adverb K, initializing score\_k=0, I access all adverb synset in the list nltk.Synsets(K) and if there is at least one synset that its definition contains ‘intensifier’ or ‘intensifiers’ then return score 1 , otherwise if definition string does not contain “intensifier” and “intensifiers” but it contains “degree” then i give another formula to compute score\_k, and the result will range from 0 to less than 1, this formula guarantee that it gives higher score for adverbs that more likely to be intensifier, the more detail for this formula is given in comment of code in file .py

**II**. Secondly, after collect possible intensifiers, I continue to deal with the second issue as follow: using some text corpus to create a list of words as l1 , then creating a list l1\_bigrams containing bigrams for l1, and use filter function to get bigrams of the form (adverb, adjective) that occur at least twice in l1\_bigrams. Next, I create an empty list A and among bigrams of the form (adv,adj) in l1\_bigrams, if adv is contained in choose\_intensifier list ( defined above) then the bigram (adv,adj) would be appended to A. By this way, I have A as a list of (adv,adj) bigrams that adv is an intensifier. Next, for the issue of restrictiveness, I set a number limit = 8 to be a threshold for restrictiveness, I create an empty list chosen\_couple, and among bigrams in list A, if the adverb in that bigram pairs with less than limit distinct adjectives (limit=8) in A, then that bigram would be appended to the list chosen\_couple. So, finally I have chosen\_couple is a list of bigrams of (adv,adj) that adv is an intensifier and it modifies for just a restrictive number of adjectives( less than 8 distinct adjectives).

**III.** Formula for scoring a couple (D,E):

Total\_score = 2\*intensifier\_score + restrictive\_score + 0.2\*occurence\_score **(F)**

As mentioned in graph I, i gives score for each adverb. In this formula, intensifier\_score measures the possibility that D can be an intensifier and it ranges from 0 to 1. I give it weight 2 to avoid case that couples of manner adverb and adjective can have high score. As mentioned in graph II, in list chosen\_couple, one adverb D can couple with several adjectives, so in the formula (F), restrictive\_score is the score measuring restrictiveness of D, and it ranges from 0 to 1, if D occurs in less number of bigram in chosen\_couple, then it gain higher score for its restrictiveness, and because the restrictiveness is our main focus, I give it weight 1. Finally, occurrence\_score is the number of times a couple (D,E) appear in the original text, and it ranges from 0 to 1, if the couple occur more, then it gain higher score. (more detail about the score formula (F) is given in comment of code in file .py )

1. **Evaluating result**: The result seems to be good because I see many intensifiers with a restricted group of adjectives that these intensifiers modify. And the ranking also looks quite reasonable. However, I still see some good couples having quite low score and ranking, so the formula (F) to give score still need more improvement.
2. **For improvement:** In this code, I choose wordnet with its feature “definition” to find out all possible intensifier, but the result still contains some manner adverb, but with more work, I think we can eliminate these manner adverbs. Furthermore, when I collect bigrams from the corpuses, I use total 5 corpus brown, Gutenberg, reuters, inaugural, webtext but only about 62 adverbs from these bigrams appear in the list choose\_intensifier but in fact, choose\_intensifier list contains about 171 adverbs. And because of the expensiveness of running code, I just use 5 corpuses, but if I have a CPU with faster speed, I will use more than 5 corpuses and get more intensifiers. Finally, the formula for scoring couples seems good but still need improvement, if I have more experiment, I can give a better weight for each factor of score in the formula (F) and have a more reasonable ranking for couples.