AMS 691.02: Natural Language Processing – Fall 2024 Assignment 1: Distributional Word Vectors

REPORT

1)Distributional Counting

1.1) I have written a code to count the number of times a word y appears in a context window with size w centered at the word x, using the provided **wiki-1percent.txt** corpus. A context window contains up to w words to either side of the center word, so it contains up to 2w + 1 words in total (including the center word).

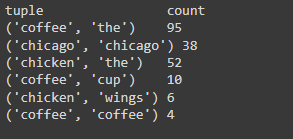
I have counted each occurrence of y in a single window and used the notation #(x, y) to denote the count of the tuple ⟨x, y⟩, i.e., the number of times that word y appeared within w words to the left or right of x. Here tuples are ordered: the first item in the tuple ⟨x, y⟩ is the center word and the second item is the context word.

Implementation:

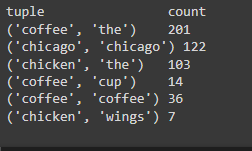
Iterated through all valid windows in each line (a valid window is one that is centered on a token in a line). For each valid window centered on word x, iterate through the w words to the left of x and the w words to the right of x. Done this by finding the index of each word x and according to the window size w calculated ind\_l and ind\_L for the left window and similarly ind\_r and ind\_R for the right window. Then, for each such word y, incremented the count for the tuple ⟨x, y⟩. Used V (for vocabulary) and VC (context vocabulary) when computing these counts.

1.2) Using vocab-15kws.txt to populate V and vocab-5k.txt to populate VC, use your code to report #(x, y) for the pairs in the following table for both w = 3 and w = 6. Some counts have already been filled in for you which you can use to check your code:

For w = 3:



For w = 6:



1.3) Using w = 3 (and again using vocab-15kws.txt for V and vocab-5k.txt for VC), evaluate your count-based word vectors using EVALWS and report your results on MEN and SimLex-999.

For w = 3:

For MEN dataset:



For the simlex-999 dataset:



Here for both the datasets, the correlation is low but still, the MEN dataset has less noise and more informative words than the simlex-999 dataset. That is why Spearman’s correlation is almost 0 for the second dataset.

2) Combining Counts with Inverse Document Frequency (IDF)

2.1) Extend your implementation to be able to compute IDF-based word vectors using Eq. 1. Using w = 3, vocab-15kws.txt to populate V, and vocab-5k.txt to populate VC, evaluate (EVALWS) your IDF-based word vectors and report your results.

Here I have used sentence retrieval, rather than documents as we are working with sentences. Let S denote the set of sentences in the corpus. Then, instead of defining word vector entries using counts #(x, y), I have defined them as follows. The word vector for a word x ∈ V has an entry for each word y ∈ VC with a value given by:

#(x, y) × (|S| / |{s ∈ S: s contains y}|)

The first term above is the “term frequency” (TF) and the second is the inverse of the “sentence frequency” for the context word.

For w = 3:

For MEN dataset:



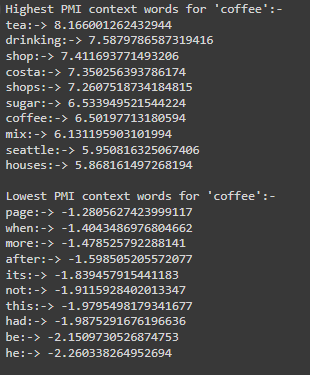
For the simlex-999 dataset:



Here the MEN dataset shows a moderate positive correlation and it is greater than the correlation we got from the raw count method that I have used in (1.3). It is because of discounting the most frequent words like is, etc. But still, the second dataset has a low correlation. It might be because of the quality of words in the dataset or it might contain frequently used words more.

3) Pointwise Mutual Information (PMI)

3.1) Implement the capability of computing PMIs. Use your code to calculate PMIs for w = 3 when using vocab-15kws.txt to populate V and vocab-5k.txt to populate VC. Note that since we are using different vocabularies for center words and context words, pmi(a, b) will not necessarily equal pmi(b, a) (though they will be similar). (If there is a word in V that has no counts, the numerator and denominator for all of its PMI values will be zero, so you can just define all such PMIs to be zero.) For center word x = “coffee”, print the 10 context words with the largest PMIs and the 10 context words with the smallest PMIs. Print both the words and the PMI values.



Here I have the PMI method to calculate the count which is using joint probabilities and partial probabilities of x and y. From the snippet above, we can see the highest PMI context words and lowest PMI context words that are related to coffee using this method.

3.2) Now, define word vectors using PMI. That is, the word vector for a word x ∈ V has an entry for each word y ∈ VC with a value given by PMI (x, y). As above, use w = 3, vocab-15kws.txt to populate V, and vocab-5k.txt to populate VC. Evaluate (EVALWS) your PMI-based word vectors and report your results.

For w = 3:

For MEN dataset:



For the simlex-999 dataset:

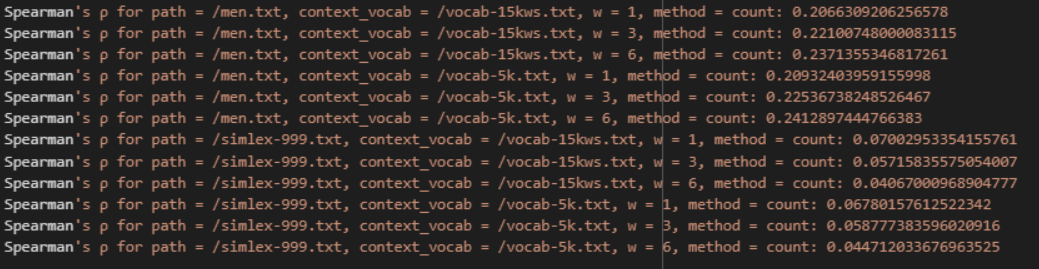


Here the MEN dataset shows a moderate positive correlation and it is almost equal to the correlation we got from the TF-IDF method that I have used in (2.1). There is a slight improvement in the second dataset when compared to the TF-IDF method.

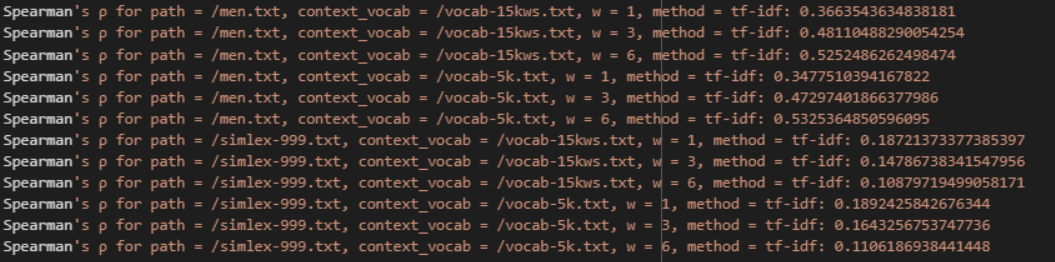
4) Quantitative Comparisons

4.1) Evaluate the word vectors (EVALWS) corresponding to the three ways of computing vectors (counts, IDF, and PMI), three values of w (1, 3, and 6), and two context vocabularies (vocab-15kws.txt and vocab-5k.txt). For all cases, use vocab-15kws.txt for V. Report the results (there should be 36 correlations in all) and describe your findings.

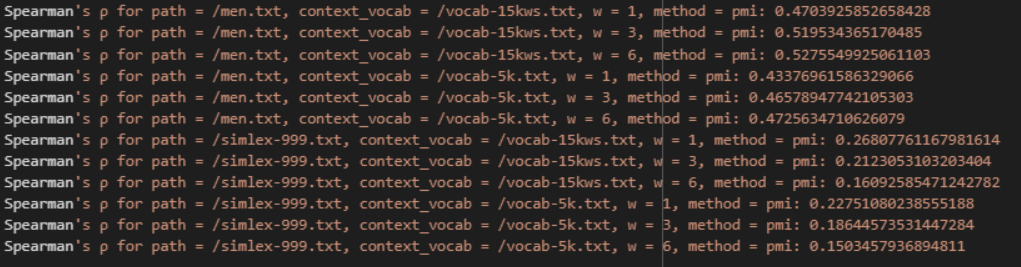
For counts:



For IDF:



For PMI:



Observations:

1. As window size increases (w = 1 to w = 6), Spearman correlations improve across all methods (count, TF-IDF, PMI) for the MEN dataset. The larger the windows better the semantic relations. The count-based method in MEN improves from 0.206 (w = 1) to 0.237 (w = 6). The TF-IDF-based method in MEN improves from 0.366 (w = 1) to 0.525 (w = 6). The PMI-based method in MEN improves from 0.470 (w = 1) to 0.527 (w = 6). But as window size increases (w = 1 to w = 6), Spearman correlations do not improve across all methods (count, TF-IDF, PMI) for the SIMLEX-999 dataset. The larger the windows worse the semantic relations. The count-based method in the SIMLEX-999 degrades from 0.07 (w = 1) to 0.04 (w = 6). The TF-IDF-based method in the SIMLEX-999 degrades from 0.187 (w = 1) to 0.108 (w = 6). The PMI-based method in the SIMLEX-999 degrades from 0.26 (w = 1) to 0.167 (w = 6). This is when I am using the vocabulary and context vocabulary the same i.e. vocab-15kws.txt. A similar trend is shown when context vocabulary is vocab-5kws.txt. So for semantic relationship MEN dataset is better than the SIMLEX-999 dataset.
2. A larger vocabulary (vocab-15k) results in higher correlations in comparison to a smaller one (vocab-5k) across all methods except the counts method which is almost the same. For ex: the PMI method with vocab-15kws.txt for MEN (w = 6) has a correlation of 0.527, compared to 0.472 with vocab-5kws.txt.
3. MEN consistently depicts higher correlations than SimLex-999 across all three methods.
4. **Count-based** method shows the lowest performance, limited by raw counts. The **TF-IDF-based** method shows better performance by reducing the impact of frequent words. The **PMI-based** method shows the best performance overall, giving weightage to valuable co-occurrences.

4.2) You should observe systematic trends in terms of correlation as window size changes which should differ for MEN and SimLex-999. Look at some of the manually annotated similarities in the MEN and SimLex-999 datasets and describe why you think the two datasets show the trends they do. Are these two datasets encoding the same type of similarity? How does the notion of similarity differ between them?

Observations:

1. Inthe **MEN** dataset as the window size increases, correlations in the MEN dataset improve across all methods, indicating it captures broader semantic contexts. This is because MEN depicts general word similarity and semantic relations. Larger window sizes provide richer co-occurrence data, enhancing the model's ability to reflect human judgments of similarity. In the **SimLex-999** dataset, shows lower correlations, even with larger window sizes. It strictly focuses on semantic similarity (e.g., "truck" and "car"), excluding functional similarity (e.g., "truck" and "road"). While larger windows improve correlation slightly, they also introduce noise by capturing functionally related but semantically unrelated words.
2. The **MEN** dataset captures a broad concept of similarity, encompassing both functional and semantic relationships. This means that words frequently co-occurring, even if not semantically related, can still be deemed similar. **The simLex-999** dataset measures strict semantic similarity, assessing words based solely on their meanings rather than functional relationships.

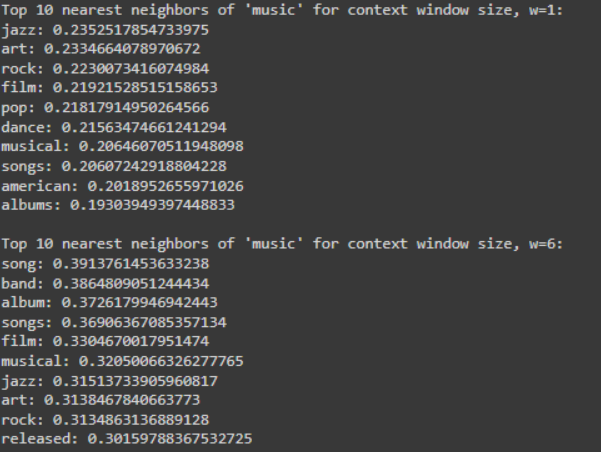
5) Qualitative Analysis

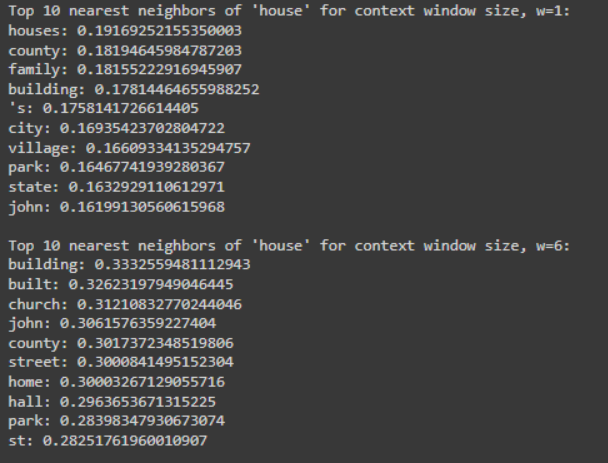
5.1) For the two window sizes w = 1 and w = 6, compute and print the 10 nearest neighbors for the query word judges. (Hint: using my implementation, the nearest neighbor for both window sizes is judge, followed by justices for w = 1 and appeals with w = 6.



5.2) Discuss your findings, showing examples of nearest neighbors for particular words to support your claims.

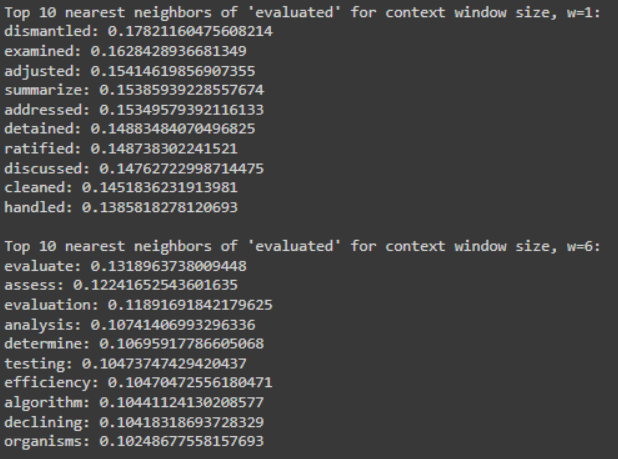
1) Nouns:

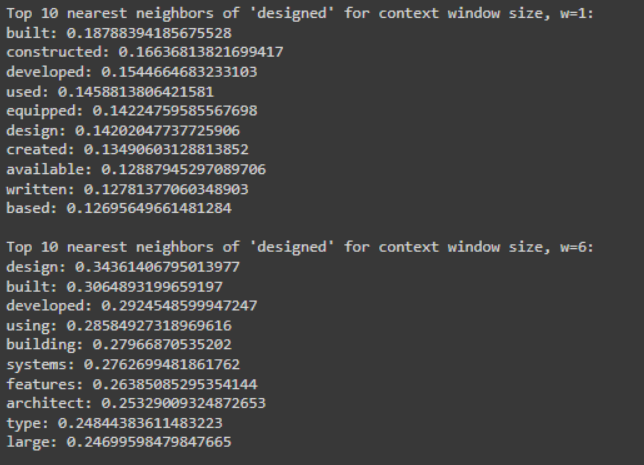




For window size w = 1, nearest neighbors for "music" include nouns like jazz, art, and film, which are related to genres or forms of media. For "house", neighbors are houses, county, and family—mostly concrete nouns and place names. For Window size w = 6, the neighbors of "music" shift to song, band, and album, which are semantically more similar and specific to the domain of music. For "house", neighbors like building, church, and home become more prominent, showcasing a tighter focus on physical structures.

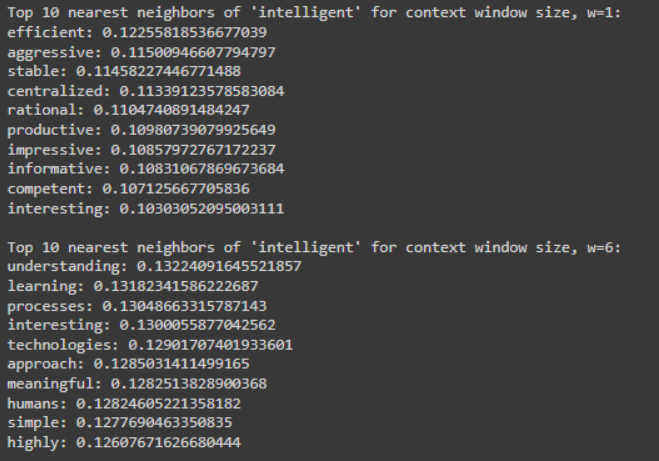
2) Verbs:

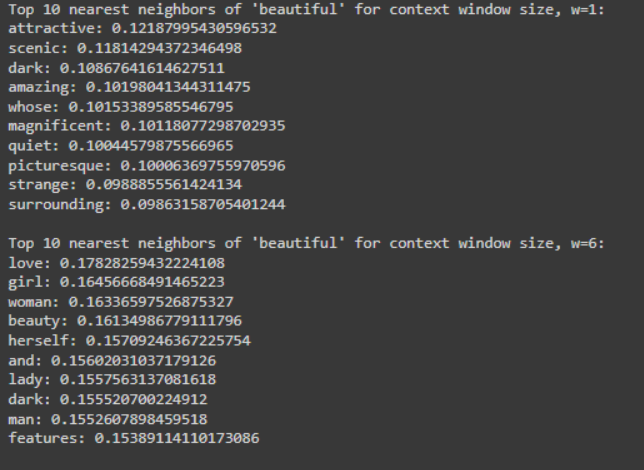




For window size w = 1, the "evaluated" verb has neighbors like examined, adjusted, and dismantled, all verbs with related meanings in the context of assessment. For "designed", neighbors like built, constructed, and developed share the same part of speech(POS) (verbs). For Window size w = 6, neighbors for "evaluated" shift to evaluate, assess, and determine which are synonyms that still maintain the same POS but focus on assessment tasks. "Designed" has neighbors like design, built, and developed, still verbs but now leaning towards architectural engineering terms.

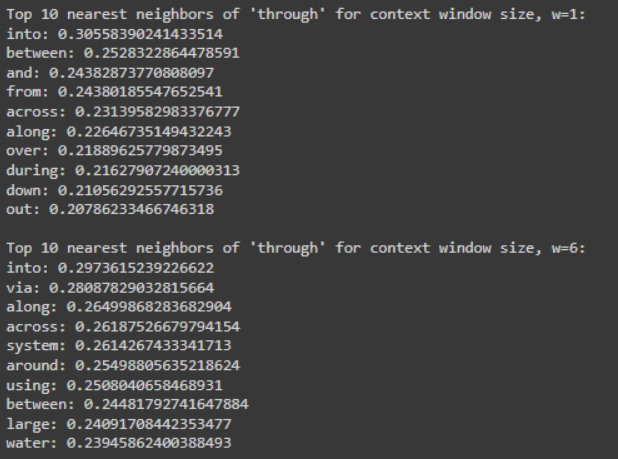
3) Adjectives:

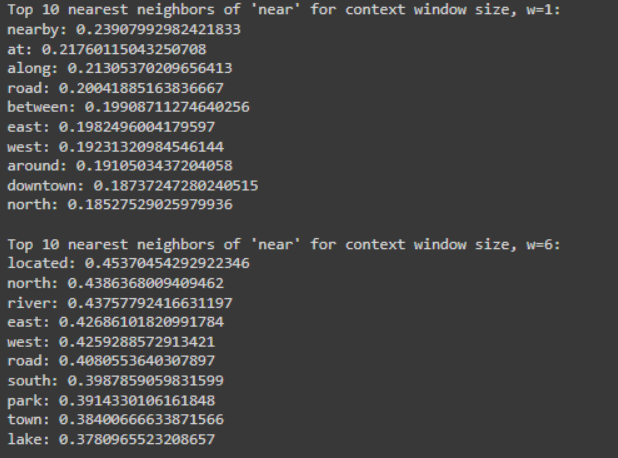




For window size w = 1, neighbors for "intelligent" include adjectives like efficient, stable, and productive, indicating functional or related characterstics. For "beautiful", neighbors like attractive, scenic, and amazing reflect appearance. For window size w = 6, neighbors for "intelligent" become more abstract, like understanding, learning, and processe*s*—still conceptually related but now including more nouns. For "beautiful", neighbors like love, woman, and herself shift to more sentimental meaning.

4) Prepositions:





For window size w = 1, neighbors for "through" include into, between, and from, all prepositions. For "near", neighbors are nearby, at, and along—also prepositions indicating location. For window size w = 6, neighbors of "through" like into, via, and along remain prepositions but with broader contextual relevance. For "near", neighbors include located, river, and town, indicating nouns related to positioning.

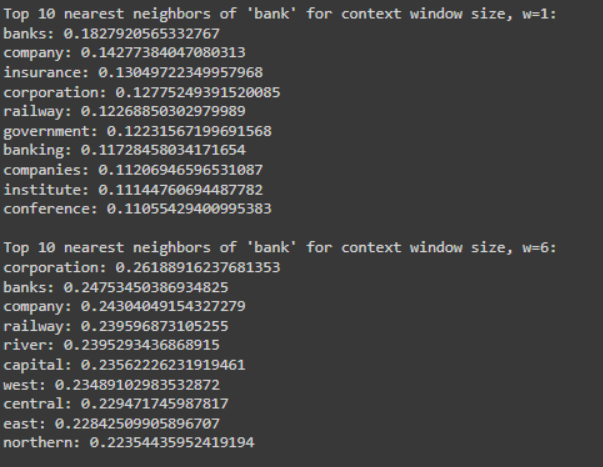
**Overall:**

The nearest neighbors of a word tend to match its part of speech tags, especially with smaller window sizes. Larger windows introduce more context-driven relationships, particularly for adjectives and prepositions. Nouns and verbs exhibit more consistent nearest neighbors across different window sizes.

5.3) Now try choosing words with multiple senses (e.g., bank, cell, apple, apples, axes, frame, light, well, etc.) as query words. What appears to be happening with multisense words based on the nearest neighbors that you observe? What happens when you compare the neighbors with different window 6 sizes (w = 1 vs. w = 6)? Discuss your findings, showing examples of nearest neighbors for particular words to support your claims.

For different multisense words:

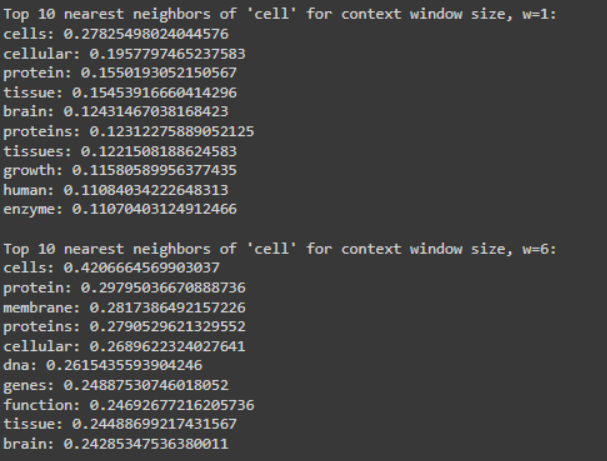
1)bank



**Window size w = 1**, neighbors like "banks", "company", and "insurance", relate mainly to the financial sense of the word. The small window highlights the local context, emphasizing the narrower sense of "bank" as a financial institution.

**Window size w = 6**, neighbors here like "river" and "capital" appear, indicating a broader geographical sense of "bank" like riverbank. The larger window captures multiple meanings, with "river" suggesting the geographical interpretation, and "capital" connecting to both finance and geography.

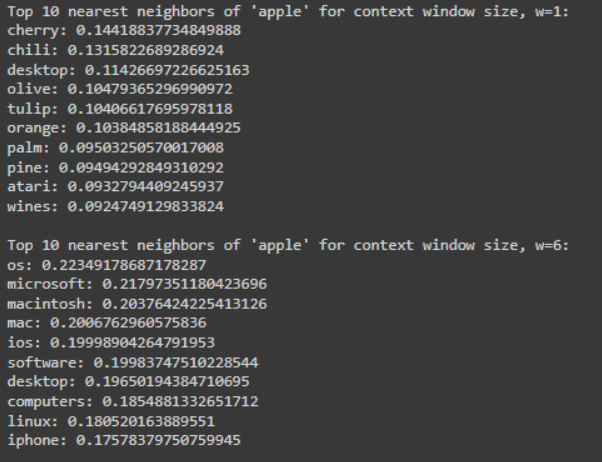
2)cell



**Window size w = 1**, neighbors like "cells", "cellular", "protein" etc. suggest a strong biological sense.

**Window size w = 6**, neighbors like "cells", "membrane", "proteins", "dna", "genes" etc. maintains the biological theme but introduces broader related concepts like "dna" and "genes," reflecting a more abstract understanding of cellular biology.

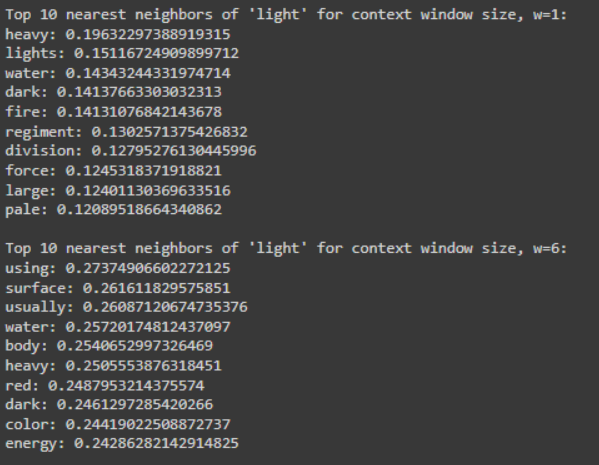
3)apple



**Window size w = 1**, neighbors like "cherry", "chili", "olive", “orange” etc.havefood-related senses of "apple", with neighbors like "cherry," "olive," and "orange." It also includes "desktop" which highlights at a technical meaning.

**Window size w = 6**, neighbors like "os", "microsoft", "macintosh", "mac" etc. clearly shifts to the technology-related sense of "apple" focusing on the technology domain.

4)light



**Window size w = 1**, neighbors like "heavy", "lights", "water", "dark"etc. focus on physical characteristics associated with light.

**Window size w = 6**, neighbors like "using", "surface", "water”, "body", "energy" etc. focus toward a broader scientific sense, highlighting a physics-related sense of "light".

**Overall:**

1. I noticed that window size impacts the nearest neighbors for words with multiple senses based on PMI vectors as shown in the examples above. Words with multiple senses have varied meanings, which makes them perfect candidates for studying the effect of different window sizes.
2. Smaller window sizes like w = 1 tend to emphasize literal meanings of words by focusing on immediate neighboring words. This is useful for capturing concrete senses, such as "bank" in a financial context or "apple" as a fruit.
3. Larger window sizes like w = 6 expand the scope to include words that co-occur in a broader context, which introduces general senses of words. For example, "apple" in the tech sense becomes dominant, and for "light" scientific meanings are captured.
4. For multisense words, smaller window sizes stick onto one meaning, while larger windows unveil different senses. This shift is visible across words like "bank," "apple," and "light," where the nearest neighbors differ significantly based on window size.