Effect of Minor Model Parameters on Graphical Model

Corpus pre-processing

1. Period: whether there is a period at the end of a sentence.
2. Sentence boundary: whether windows are bounded by sentence boundaries.

During corpus-pre-processing, we varied whether we included a period at the end of a sentence, and whether to count co-occurrences across sentence boundaries. WE explored the latter variable because words within a sentence may bear stronger syntactic and semantic relations compared to words across sentences. When applied, sentences are considered as independent units, and co-occurrence counting only happens within a sentence. In the 'no boundary' condition, we take the more common approach of counting co-occurrence across sentence boundaries. For example, in Figure C.1, the verb 'search' and 'go_to' are not linked when keeping track of sentence boundaries, because they never occur in the same sentence. However, in a 'no boundary' model, 'go_to' is the fourth word coming after 'search', and thus there can be a co-occurrence count between the words if the window size is at least 4. Whether or not to count the period at the end of a sentence, at first, appears to be a minor issue, yet there is work showing that this choice does influence results. Furthermore, although it is not very clear how the period influences the spatial models, it makes a discernible difference in the co-occurrence graphical representation.

Co-occurrence Matrix Formation

1. Window type: forward, backward, sum
2. Window size: 1,2,7
3. Window weight: flat, inverse
4. Normalization: no normalization, row-log, PPMI

In this work, we follow the procedure for encoding word co-occurrence used to construct HAL (Lund & Burgess, 1996). For each word type in the corpus (referred to as target), we count the occurrence of words within the sequential neighbor (window) of the word type. The window type determines whether co-occurrences are counted forward, backward, or whether the two types of occurrences are summed (both forward and backward occurrences are counted). Next, the size of the window may vary (e.g. size 1 only counts the immediate neighbors, etc.). Next, the co-occurrence count is influenced by the variable 'window weight': In a 'flat' window, a count is incremented by exactly 1 no matter how far away two words are within a window, while in the 'inverse window', the count value is inversely proportional to the distance separating the word. For example, in Figure C.1, 'go_to' and 'search' are 4 words apart, thus the 'inverse' window will make the count between them 1/4, while a flat window results in a count of 1. Lastly, normalization is a transformation of the co-occurrence matrix commonly used in distributional modeling. We include a null condition without any normalization, row-log normalization, and normalization using Positive Pointwise Mutual Information (PPMI). In row-log, we add 1 to the co-occurrence count, take the log of each entry, and then normalize the entries by their corresponding row sums. PPMI is obtained by taking the positive part of PMI transformation, which is a way to rule out the effect of absolute word frequency.

$$PPMI(x,y) = max(log\frac{p(x,y)}{p(x)p(y)}, 0)$$

The six minor variables determine the shape of the co-occurrence graph and the weight of the connections in the network. The form of the graph and the weight of the edges are critical determinants of how activation spreads, and therefore influences our semantic relatedness score. The minor variables are grouped into two classes: Presence or absence of a period, sentence boundary, and window size determine the shape of a graph, while window type, window weight

and normalization have effects on the edge weights. A detailed discussion of such effects follows.

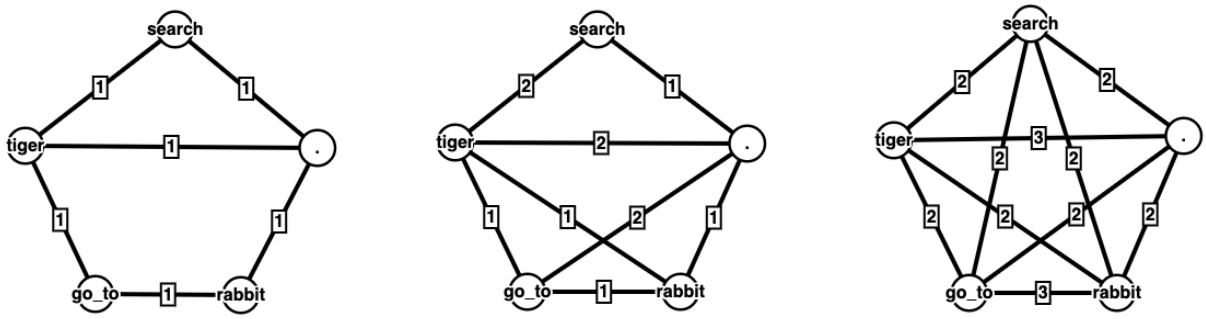
Effects of Minor Variables on Co-occurrence Graph

Window Size

The choice of window size may affect the co-occurrence graph. To illustrate, we plot the co-occurrence graph based on the mini corpus shown above. The three co-occurrence graphs shown in Figure 1 were created with a window size of 1, 2, and 7, respectively. When only adjacent co-occurrence is encoded (Figure 1a), the graph is relatively simple. Two sentences are chained together, with ‘tiger search.’ in the upper arch and the sentence ‘tiger go_to rabbit.’ in the lower arch. Both sentence-final words are linked to the period, resulting in a ring. Since in this case we include periods and ignore sentence boundaries, the second occurrence of ‘tiger’ follows the first period, and this means their nodes must be connected in the network

Figure 1:

An Illustration of Co-occurrence Graphs Varied by Window Size



Note. The graphs are built from the mini corpus 'tiger search. tiger go_to rabbit.' with window size 1 (C1a), 2 (C1b), or 7 (C1c) (from left to right). The edges are undirected and self-loops (a word co-occurs with itself) are not included.

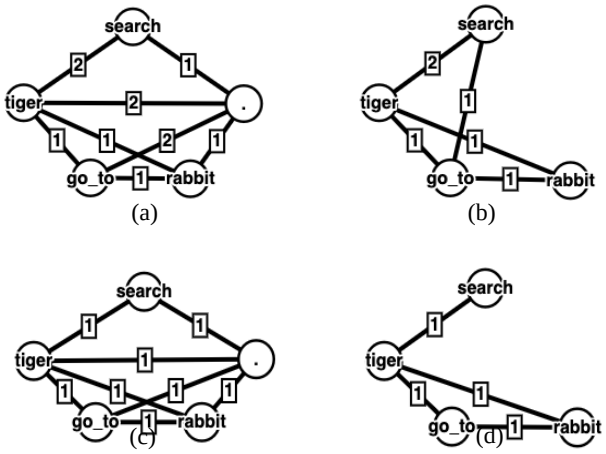
In Figure 1b, where the window size is 2, we have two more edges connecting the two animal words and 'go_to' with a period. In the extreme case where window size is 7, all words are connected to each other since the window size covers the whole corpus. The effect of window size is clear on Figure 1: The larger the window, the more co-occurrences are counted, and the denser the network becomes.

Sentence boundary

Let us further examine the network shown in Figure 1b with different sentence boundary conditions. The graph in Figure 2b is obtained by getting rid of the period in the corpus, while 2c and 2d are obtained by respecting sentence boundaries. When sentence boundaries are respected, words across sentences are never connected. For example, the two verbs 'go_to' and 'search' are connected in 2b, as the words are only two slots apart and the co-occurrence go across the sentence boundary. However, such connection is absent when there is a sentence boundary, as in 2d. Respecting sentence boundaries has both advantages and disadvantages. On the one hand, sentences or clauses next to each other are usually semantically related, and, consequently, words separated across a sentence boundary might also be semantically related. Therefore, the semantic information of inter-sentential semantic relations are lost if sentence boundaries are respected. On the other hand, the words within a sentence are more directly related compared to words across sentences. For example, 'rabbit' is the patient of 'go_to' in the corpus and is not directly related to the predicate in the former sentence. However, if the sentence boundary is ignored and the window size is large enough, 'rabbit' and 'search', would be connected in the graph.

Figure 2:

An Illustration of Co-occurrence Graphs Constructed Varied by Period and Sentence Boundaries



Note. Graphs constructed with inclusion of period in the corpus in (a) and (c), and without period in (b) and (d). Graphs constructed with sentence boundary in (c) and (d), without sentence boundary in (a) and (b).

Period

The inclusion of periods makes the sentence-initial and sentence-final words closer in the network. Since most of the sentence-initial and sentence-final words are not related to each other, the inclusion of periods may produce unwanted relations. This is related to the hub effect, which describes situations in which the inclusion of functional units like punctuation and function words, links otherwise unrelated content words via densely connected hub nodes. Nevertheless, removing the period may lead to alternative issues. For instance, when periods have been excluded, and sentence boundaries are not respected, the initial word of every sentence will be

directly linked to the final word of the previous sentence. Generally, a good compromise is to exclude periods, and to respect sentence boundaries. However, this strategy is not always ideal, and care must be taken to customize these settings.

Window Weight, Window Type, And Normalization

These three variables do not add or remove edges in a graph, but change the strength of its edges. Let us start with window weight. When the window size is greater than 1, it is an empirical question whether words with different sequential distances should have the same co-occurrence count in the corpus. Intuitively, words closer to each other should be more related than words that occur farther apart. Taking the inverse of the distance is a way to accomplish this. For example, we can assign words that occur at a distance of 10 a weight of 1/10. Window type and normalization methods work together to influence the weight of edges in the graphs. Briefly, normalization transforms co-occurrence values by considering the frequency co-occurrences in the same row and/or column of the co-occurrence matrix. Window type determines whether co-occurrences are counted in the forward, backward, or both directions.

Effects of Minor Variables on Similarity Graph

Since the similarity models are derived by computing the similarity score of rows in the co-occurrence matrix, the influence of the minor parameters are less transparent. Different choices on the preprocessing of the corpus, the window parameters and normalization methods will end up in different co-occurrence matrices, making some word vectors more similar in one model, and other word pairs more similar in other models. As a result, different co-occurrence matrices give rise to different similarity tables.