Semantic distances and path entropy

bo LIU

June 2016

The main idea of the work is measuring the semantic distance between different blocks in the book. For simplicity we use the word page for block of words.

Each page is a combination of words. So without sequencing information, you can regard it as a distribution of words, also a distribution of meanings. So with the work from Gerlach [2] and Masucci [3], we can compute the distances between distributions.

With the distances between each pair of two pages, we can calculate the path entropy. The path entropy can distinguish the real route (p1 to p2 to p3...) from random routes like p1 to p117 to p36 to p88.... One application maybe re-ordering the randomised pages from a book.

The path entropy between node is defined on complex networks. With a network adjacency matrix A, from which the entry a_{ij} is the distances between node i and j. So the path entropy between node i and j, which is the amount of information needed given you are on node i and want to go to node j (among the first neighbors of node i), is,

$$I_{ik} = \frac{a_{ik} * \log a_{ik}}{\Sigma_{m \subset su_i} a_{im} * \log a_{im}}$$

su_i is the successors of node i in directed network. For undirected network, this is the first neighbors of node i. Then the path entropy for the path $P = [i, j, k, l, ...]$ is,

$$I_P = \Sigma_{i,j} I_{i,j},$$

i and j is the successive pair on path P.

I used 34 books from Charles Dickens. Fig. 1 section shows that this method distinguishes the real path from random paths well.

Calculate the semantic distances between two pages

Get the meaning of words

Co-occurrence matrix and Singular Value Decomposition (SVD) are the cores of this procedure.

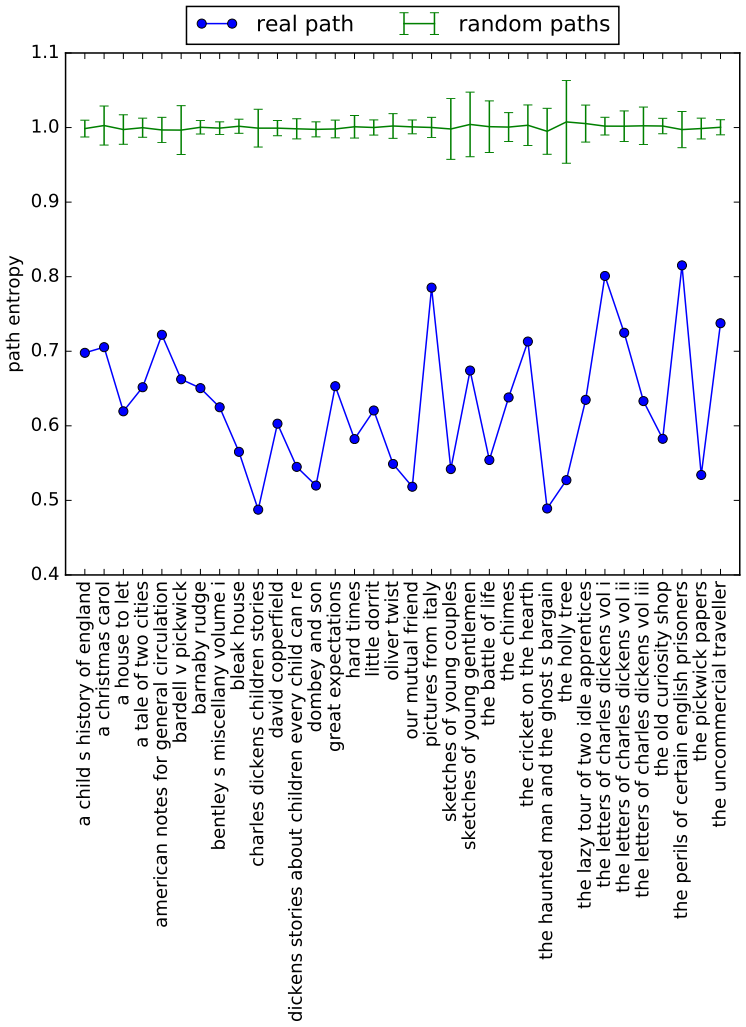

Figure 1: The path entropy for real path and 50 random paths from Charles Dickens' book.

First we build a co-occurrence matrix of the entire text. For a text $T = \{w_i | i \in 1...N, w_i \in typ\}$, in which i is the type index and N is the number of tokens, typ is the types set of the book, we define co-occurrence matrix c_{ij} as the times w_j appear within a certain distance (half of window length a) from w_i. When window length a is the smallest, this becomes the normal co-occurrence. We use the co-occurrence matrix to represent the meaning of words.

The idea behind this co-occurrence matrix is that words are semantically close when they appear close to each other in the text. And furthermore, we address the problem of polysemy and synonymy by connecting multiple co-occurrences to a word. One co-occurrence only addresses one part of the meaning of a word. So multiple co-occurrences make a complete combination of the meaning for a word in this specific text. The stronger co-occurrence, the closer semantic, which means the word happens in the text more with this meaning. By this two advantages, we use the co-occurrence matrix to represent the meaning of a word.

Then SVD method [1] is done on the co-occurrence matrix,

$$C = USV^T.$$

C is the co-occurrence matrix. U is the orthonormal singular vectors of CC^T, ordered with the corresponding singular values. V is the orthonormal singular vectors of C^TC, ordered with the corresponding singular values. S is a diagonal matrix containing the square roots of eigenvalues from U and V in descending order.

What SVD does is it takes the original co-occurrence matrix and then breaks it down into linear independent vectors. It actually linearly transforms the bases from the distinct words into one that represents the principal directions. For the majority of the vectors, the corresponding singular value/eigenvalue is very small, they can be ignored. So we can approximate the data with much smaller size than the original. U and V are actually the same since C is a symmetric matrix. They both represent the linear coefficient for the new bases. S represents the variance of the vectors along the new bases.

There are two reasons to use SVD here.

First, this is a the requirement of the computation feasibility. If we use distinct words as the domains of the distribution, data would be noisy and size too big. We can consider the distribution of the distinct words as a vector in the space with the distinct words as bases. Since SVD can identify the main directions of the vector (those with highest singular values/eigenvalues). Then we just need to choose the main directions as the result bases/domains of the vector/distribution. By doing this, we remove the noises from the co-occurrence matrix and removed the size of the data.

Also later, comparison of distributions with Jensen-Shannon Divergence prefers agreement of attribute domains. But for two pages, it is possible that they share no common words other than the function words. SVD convert the bases into a common set, so that all the resulting vectors have non-zero weight on all the new bases, enabling the comparison.

The resulting S matrix decays very fast. So we need a small part to approximate. Size of S is $N \times N$. Then assuming we take the first n dimensions of S, we get the final word representing matrix as

$$M_{N\times n} = U_{N\times n} S_{n\times n}.$$

In M, each row vector represents a word in the new bases.

Calculate the semantic distances between two pages

We assume the semantic of a page is the sum of the semantic of words on the page. For the text $T = \{p_j\}$, p_j is the page, we have $p_j = \{w_i\}$. Then the semantic of page j is,

$$S_i = \Sigma_j \vec{m_j}.$$

j is the index of the tokens on page i.

Since the vector can be regarded as a distribution, we can use the Jensen-Shannon Divergence to calculate the distance between two distribution [3].

For two distribution S_i and S_j, the Jensen-Shannon Divergence is,

$$JSD(S_i||S_j) = H(S_i + S_j) - H(S_i) - H(S_j).$$

$$H(S_i) = -\Sigma_i p_i \log p_i.$$

$H(S_i)$ is the Shannon entropy for the distribution S_i.

Then for each pair of two pages p_i and p_j, we can compute the Jensen-Shannon Divergence as the semantic closeness a_{ij}.

References

[1] Kirk Baker. Singular Value Decomposition Tutorial, jan 2005.

[2] Martin Gerlach, Francesc Font-Clos, and Eduardo G. Altmann. Similarity of Symbol Frequency Distributions with Heavy Tails. *Physical Review X*, 6(2):021009, apr 2016.

[3] A. P. Masucci, A. Kalampokis, V. M. Eguíluz, and E. Hernández-García. Extracting directed information flow networks: An application to genetics and semantics. *Physical Review E*, 83(2), 2011.