On the Relation between the NTT Kanji data dictionary and Wikipedia

Shin Asakawa
(Tokyo Womens’ Christina University)

Key words word embeddings, vector space models, word2vec, t-SNE, visualization

The NTT Kanji database ([天野・近藤 1999]) is one of the most popular kanji databases. However, it has been long ever since it published. The word2vec (Mikolov, Yih, & Zweig, 2013; Mikolov, Sutskever, Chen, Corrado, & Dean, 2013; Mikolov, Chen, Corrado, & Dean, 2013; Mikolov, Le, & Sutskever, 2013) was proposed based upon large vocabulary. The word2vec was insisted that it could be dealt with semantics as well as syntax, got popularity among machine learning scientists. In spite of these popularities, no trials have been attempted to compare them so far. Here, we tried to compare them in term of a modern visualization, t-SNE (Maaten & Hinton, 2008).

Since the NTT Japanese lexical database consisted of multi sources, we must take averages among same indexes. For example, the ‘psylex7.txt’, ‘psylex71.txt’ and ‘psylex72.txt’ give frequencies sources originated from articles or documents. Therefore, we did not get unid key to integrate between the NTT database and wikipedia. In order to get unid index keys in the NTT dataset, we averaged over all the indexes, then we tried to merge with wikipedia. We used the ‘mecab’ to separate all the Japanese documents in the wikipedia.ja into particles.

Wikipedia downloaded all the wikipedia documents in https://dumps.wikimedia.org/jawiki/latest/. We could get the data time stamped 12/Sep/2016. We made all the wikipedia documents divided with the ‘mecab’ into Japanese particles with option ‘-Owakati’. We got 1,265,697,011 particles in the wikipedia.ja, that were 1,140,357 vocabularies as tokens when we processed low frequency words (less than 5) as unkown token. When we set minimum count to 1, we got 5,774,139 tokens.

The data was processed by word2vec (cbow and skip-gram), then we got 128 dimensional vector for each word.

A popular method for exploring high-dimensional data is t-SNE, introduced by van der Maaten and Hinton. The technique has become widespread in the field of machine learning, since it has an almost magical ability to create compelling two-dimensional “maps” from data with hundreds or even thousands of dimensions.

t-SNE is the high dimensional data $p_{ji}$ to map onto a low dimensional map $q_{ji}$. Here the cost function to be minized can be written as:

$$C = \sum_i KL(P_i || Q_j) = \sum_i \sum_j p_{ji} \log \frac{p_{ji}}{q_{ji}},$$

in which $P_i$ represents the conditional probability distribution over all other datapoints given datapoint $x_i$, and $Q_j$ represents the conditional probability distribution over all other map points given map point $y_j$. $KL$ denote the Kullback-Leibler divergence. The SNE cost function focuses on retaining the local structure of the data in the map.

References