Word2Vec

Contents

What is Word2Vec?

Word2Vec is a two-layer neural network applied to text. The input of Word2Vec is a text corpus and the output is a set of vectors. **The input corpus must be in the English language.** Word2Vec turns text into a numerical form, with groups the vectors of similar words clustered together in vector space. The vectors used to represent words are called **neural word embeddings**. So a neural word embedding represents a word with numbers. By "vectorizing" words, Word2Vec makes natural language computer-readable; on vectors we can perform powerful mathematical operations in order to detect similarities among words mathematically, **without human intervention**. Word vectors will place similar words close to each other in space. Thus, the words cat, dog and chicken would most likely cluster in one corner, while car, road and toll cluster in another.

Similar things and ideas are shown to be "close". Their relative meanings have been translated to measurable distances. Qualities become quantities, and algorithms can do their work. But similarity is just the basis of many associations that Word2Vec can learn. For example, it can gauge relations between words of one language, and map them to another.

Given enough data, usage and contexts, Word2vec can make highly accurate guesses about a word's meaning based on past occurrences. Those guesses can be used to establish a word's association with other words (e.g. "man" is to "boy" what "woman" is to "girl"), or cluster documents and classify them by topic. Those clusters can form the basis of search, sentiment analysis and recommendations in such diverse fields as scientific research, legal discovery, e-commerce and customer relationship management.

PC-ACE Implementation of Word2Vec

The PC-ACE application of Word2Vec is based on DL4J (Deeplearning4j, Deep Learning 4 Java) Word2Vec (http://deeplearning4j.org/), modified by Alberto Purpura. English is the only language supported by PC-ACE Word2Vec. After computing the Word2Vec model, PC-ACE displays word similarities in a network graph in Gephi or in a scatterplot in Excel.

THE WORD2VEC ROUTINE HAS AL ELEMENT OF RANDOMNESS (LIKE MALLET). RESULTS FROM SUCCESSIVE RUNS ON THE SAME DATA WITH THE SAME OPTIONS MAY NOT YIELD THE EXACT SAME RESULTS.

An Example

Suppose that you want to analyze the following two newspaper articles that come with PC-

ACE sample database with file names: Atlanta Constitution_2-9-1888_2.docx and Atlanta Constitution_2-10-1888_2.docx. Both documents have been merged into a single txt document, containing the following information.

THE LYNCHING IN LIBERTY.
Fuller Particulars about the Killing of an Incendiary.

Savannah, Ga., February 9-[Special]-Very little information can be obtained from Liberty County about the lynching of the negro incendiary Tuesday night. About a fortnight ago The Constitution published an account of a fire at Johnson's station, on the Savannah, Florida and Western railway. Mr. Chapman lost a store and the railway company's warehouse was burned, along with several other buildings. It was suspected that the fire was started by an incendiary, and on Tuesday a negro was arrested on suspicion. He was given a preliminary hearing, and confessed that he was one of a party of five who broke into Chapman's store. After stealing all they could carry off, the burglars sprinkled kerosene about the building and set fire to it. The magistrate committed the negro to jail. While the deputy sheriff was on his way to the Hinesville jail, he was surprised by a crowd of fifteen men, who took the prisoner away from him. The negro was carried in to woods, and it is supposed he was hung or burned. The officer never saw him more. It is expected that the other incendiaries will share the same fate.

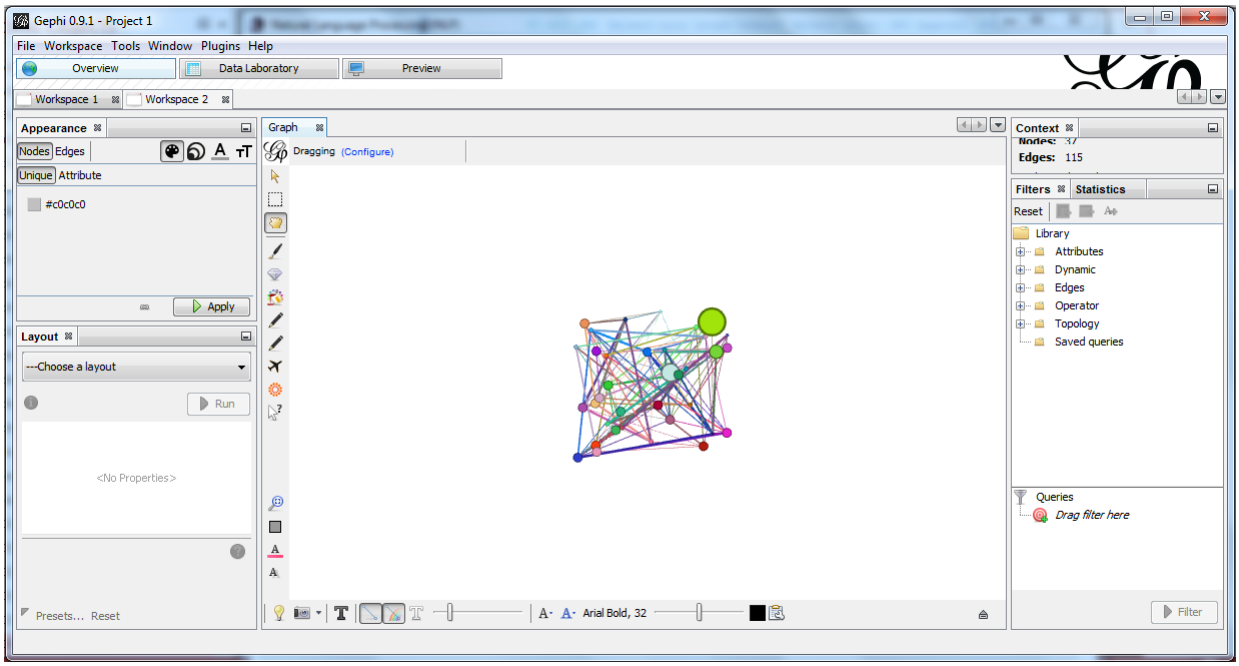
SHOT TO DEATH.
An Incendiary Put Out of the Way.
ARMED MEN VISIT HINESVILLE JAIL,

Overpower the Jailer, Seize a Negro Prisoner and Riddle Him With Bullets-Great Excitement Prevails.

Savannah, February 8.-[Special.] -A few weeks ago a house and a warehouse were destroyed by fire in Hinesville, and all the circumstances pointed to its being the work of an incendiary. The people have been greatly wrought up in consequence. Intelligence received here tonight states that a negro was arrested there yesterday on the charge of burning the houses aforesaid. He is said to have confessed the deed, and implicated several in the crime. After a preliminary investigation, he was committed to jail in Hinesville. Last night a band of armed men overpowered the deputy sheriff, who had the prisoner in charge, and carrying him off the woods shot him to death. Great excitement prevails in that section.

Visualizing Results as Gephi Network Graph

The user can set the maximum number of nodes to be displayed in the graph and its density. The following symbols are excluded from the PC-ACE Word2Vec computations: punctuation symbols, have, has, had, been, was, were, is, are, not, then, coordinating conjunction, cardinal number, determiner, existential there, preposition or subordinating conjunction, list item marker, modal, comparative and superlative adverb, particle, symbol, to, interjection, wh-determiner, wh-pronoun, possessive wh-pronoun, wh-adverb, personal pronoun.



The Word2Vec output graph visualized in Gephi may be quite hard to read.

Needless to say, there are a number of different ways to improve the graph readability; please read the TIPS file **Gephi Displaying network graphs.**

The Java Word2Vec routine also has options that would improve the readability of the graph prior to the visualization of the graph in Gephi, by changing some of the **parameters** for:

1. the maximum number of nodes visualized in the Gephi graph (default=100)
2. the number of near words to be considered in the Gephi graph (Graph density; default=5)

Thus, if you want greater detail, you can increase the default graph density value. If the graph then becomes too "busy", with too many nodes, you can reduce the maximum number of nodes to be visualized at the expense of precision.

Note!
Number of nodes. You can set the number of nodes to any value. If the value entered (e.g., 200) is greater than the actual number of nodes in the text (e.g., 50), the value entered will be ignored and the computations and display will be based on the actual value.
Graph density. It is unlikely that a value for graph density greater than 10 or 15 may produce meaningful results.

If you set the number of nodes to N, the routine will visualize only the nodes in the model that correspond to the words with the N most frequent words present in the text; in other words, the routine sorts words by their decreasing frequency and considers the set of first N. Remember that, even with a very large number of nodes, you can always focus on specific nodes by hovering over a specific node or by changing the layout options of the graph (see the TIPS file Gephi Displaying network graphs).

How do you read nodes and edges in the Word2Vec Gephi graph?

In the Gephi graph, each **node** represents a different word of the text, with the size of the node

proportional to the frequency of the word in the text. For large texts, with many words characterized by large frequencies, visual differences in node size will be negligible. The **edge** thickness between any two nodes (i.e., between any pair of words) will measure the estimated similarity between the pair of words, according to the Word2Vec model.

Visualizing Results as Excel Scatterplot

In an Excel scatterplot, a handful of words will be displayed in a 2x2 Cartesian space, where words selected by Word2Vec for their proximity will be displayed. Look carefully to see whether the spatial distribution of words suggests connections you had not thought about.