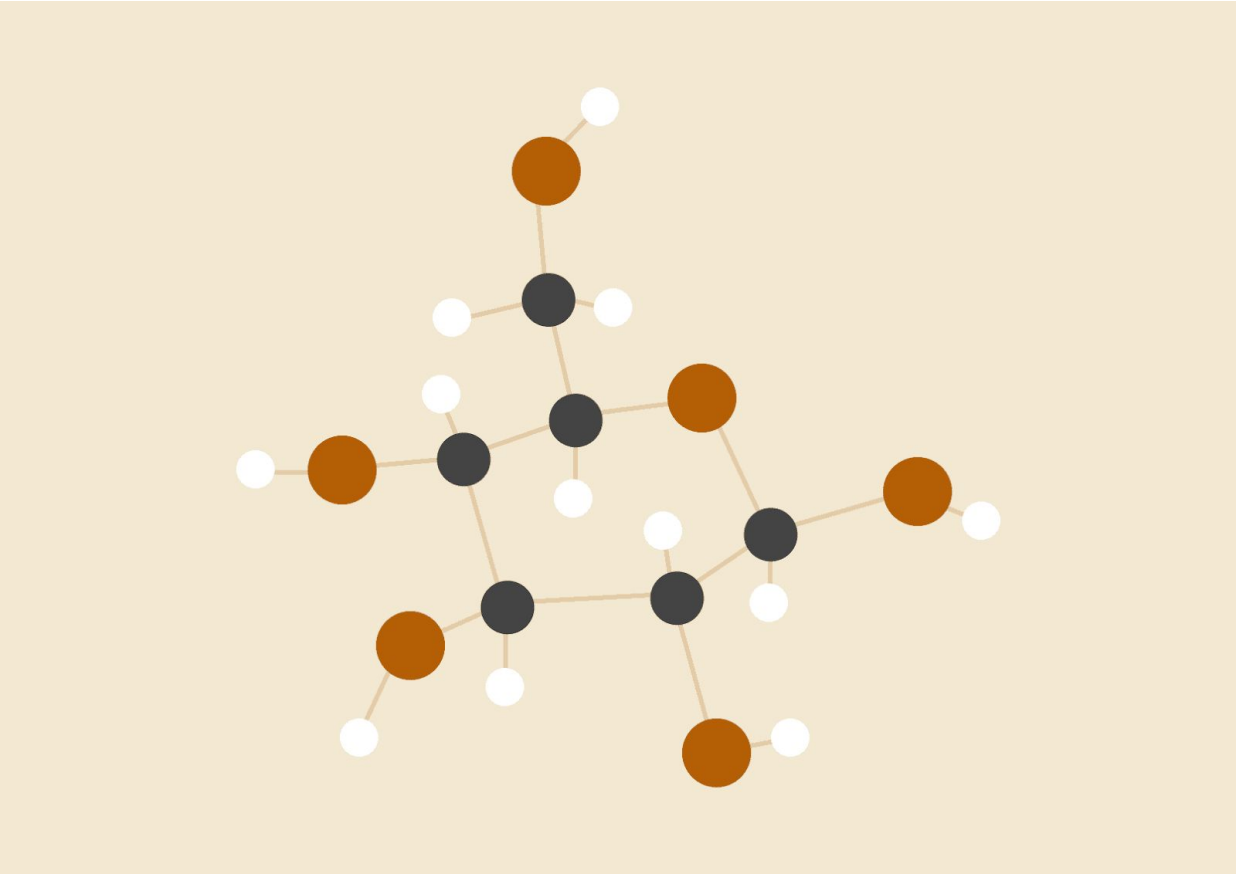
Wikivoyage Data Analysis



Kenneth Ho

04-01-2018

INTRODUCTION

This analysis studied the pages from the Wikivoyage by downloading the dataset from https://dumps.wikimedia.org/enwikivoyage. Three analysis results will be presented in this report: 1) Statistical tests on the links per articles(pages), 2) Centrality of all articles using Eigenvector approach; 3) Text analysis using Word2vec running in Tensorflow.

PROCEDURE

All the source code is located in https://github.com/kenneththo/wikimediaAnalysis.git including this report. There is one script (setup.sh) for setting up the system and environment and another one (run.sh) for performing the three analysis.

There are several steps for executing the entire process:

1. The data will be downloaded and uncompressed from wikimedia site as an XML file.
2. Article, title, http url & redirect links will be extracted from each pages and stored in the MongoDB. (This will greatly increase the performance when the analysis needs to be rerun)
3. The data stored in the MongoDB has a simple json format:

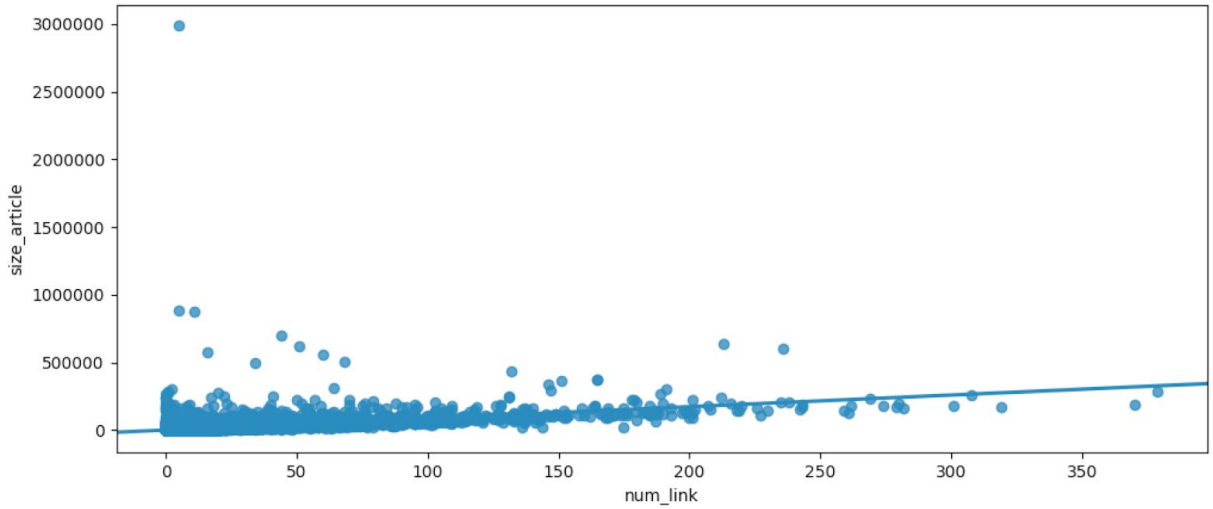
4. Data will be directly retrieved from MongoDB for each analysis
5. The setup.sh should be run first for downloading the mongodb tarball and installing. Mongod will be run in the background (not a init.d service)
6. Run.sh will execute all the analysis and the results will be shown on console as well as matlab plot.

ANALYSIS

I. SIZE OF ARTICLES VS NUMBER OF LINKS HYPOTHESIS TEST

There are 2 types of links existing in the articles 1) http link 2) reference / page redirecting. The http links embedded in the article in the format of the http[s], but page redirect is in the format of [[xxx]] where xxx is the title of a different article. The code extracted both and performed an OLS linear regression (article size Vs number of links) for both cases.

1) Statistical Test for links only:



	Title	**Num of Links** **Average = 4.16** **Median = 0**	**Article Size**
Article (Minimum Links)	El Valle de Antón	0	44
Article (Maximum Links)	London	379	289294

```
                            OLS Regression Results
==============================================================================
Dep. Variable:           size_article   R-squared:                       0.329
Model:                            OLS   Adj. R-squared:                  0.329
Method:                 Least Squares   F-statistic:                 2.939e+04
Date:                Sun, 01 Apr 2018   Prob (F-statistic):               0.00
Time:                        09:34:29   Log-Likelihood:            -6.6917e+05
No. Observations:               59890   AIC:                         1.338e+06
Df Residuals:                   59888   BIC:                         1.338e+06
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
Intercept   1810.9359     73.440     24.659      0.000      1666.993  1954.879
num_link     861.0942      5.023    171.423      0.000       851.249   870.940
==============================================================================
Omnibus:                   235635.343   Durbin-Watson:                   2.003
Prob(Omnibus):                  0.000   Jarque-Bera (JB):     596134476069.921
Skew:                          96.798   Prob(JB):                         0.00
Kurtosis:                   15457.911   Cond. No.                         15.3
==============================================================================
```

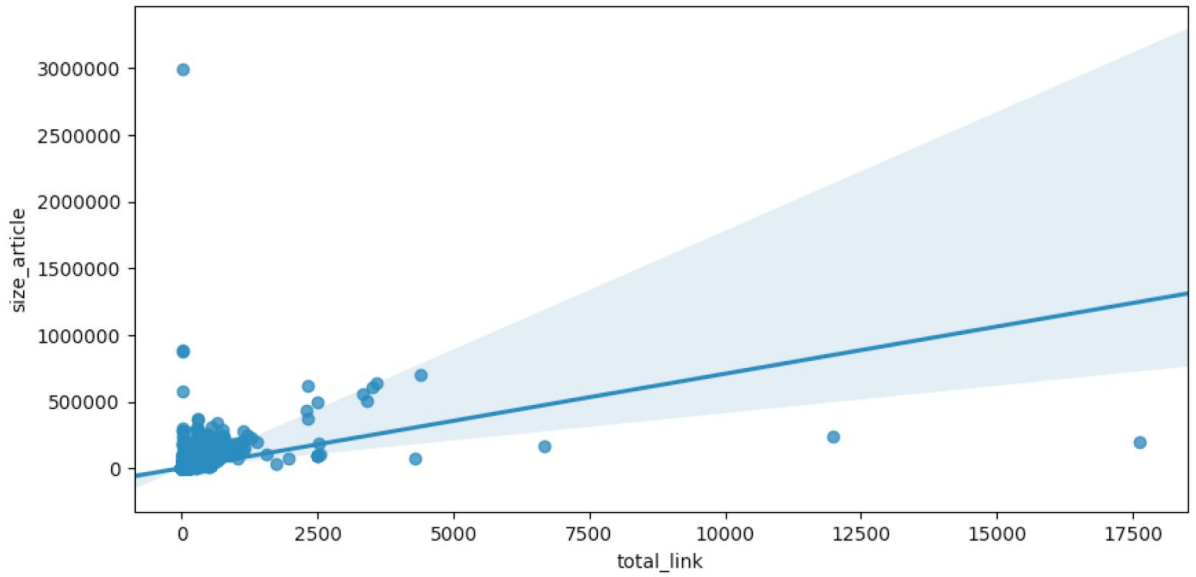
T-Test can be used to conduct the hypothesis of the following:

$H_o : num\ link_{coef} = 0$

In this case t = 171.423 and p_value = 0, hence the hypothesis is rejected. It can be concluded that the relationship between num link and article size is linear.

Another metric to look at is the R^2 which can be used to measure whether all the data points are fitting along the regression line appropriately. In this case, R^2 = 32.9% which indicated the regression is not very well fit to the model.

2) **Statistical Test for links + Redirect:**



	Title	**Num of Links** **Average = 16.5** **Median = 2**	**Article Size**
Article (Minimum Links)	Similkameen	0	66
Article (Maximum Links)	Wikivoyage:Airport Expedition/TLAs	17632	198220

```
                            OLS Regression Results
==============================================================================
Dep. Variable:           size_article   R-squared:                       0.144
Model:                            OLS   Adj. R-squared:                  0.144
Method:                 Least Squares   F-statistic:                 1.006e+04
Date:                Sun, 01 Apr 2018   Prob (F-statistic):               0.00
Time:                        10:25:40   Log-Likelihood:            -6.7648e+05
No. Observations:               59890   AIC:                         1.353e+06
Df Residuals:                   59888   BIC:                         1.353e+06
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
Intercept   4223.7529     80.389     52.542      0.000      4066.191  4381.315
total_link    70.6239      0.704    100.303      0.000        69.244    72.004
==============================================================================
Omnibus:                   205734.687   Durbin-Watson:                   1.994
Prob(Omnibus):                  0.000   Jarque-Bera (JB):     231544765732.835
Skew:                          63.680   Prob(JB):                         0.00
Kurtosis:                    9634.812   Cond. No.                         115.
==============================================================================
```

T-Test can be used to conduct the hypothesis of the following:

$H_o : num\ link_{coef} = 0$

In this case t = 100.3 and p_value = 0, hence the hypothesis is rejected. It can be concluded that the relationship between num link and article size is linear.

In this case, R^2 = 14.4% which indicated only 14% of the data points fitted to the regression line which means much worse than the previous analysis.

2. CENTRALITY ANALYSIS

The opensource python library networkx is used to compute eigenvector centrality. In order to obtain a subset of the graph, depth-limited DFS algorithm is used for specifying a root and the depth of the subgraph. For demo purposes, the code is set to retrieve a graph starting from “California” as root, and a depth of 2.

The result is as follow:

Top 10 most central articles:

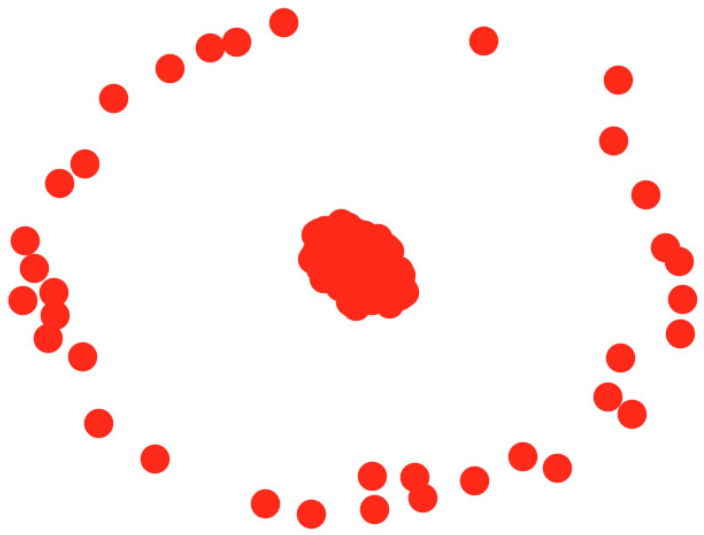
Article	Centrality Score
UNESCO World Heritage List	0.26488556225673177
Turkey	0.13632379457670354
Canada	0.1341159450974553
Europe	0.1269035963495885
United Kingdom	0.1235240993031823
Russia	0.11762085514398726
LGBT travel	0.11698487993196494
Australia	0.11477476567261292
Singapore	0.1131672015534571
Argentina	0.11187756108754959

By Setting root to “los Angeles” with a depth of 1:

Article	Centrality Score
California	0.304830069461745
San Francisco	0.1967665574050977

Interstate 5	0.19437269438108118
Southern California	0.19409421522912138
San Diego	0.185717063779078
Sacramento	0.15426670280723723
United States of America	0.14585963958423798
Chicago	0.1362535089894396
Las Vegas	0.1329077040659322
Mexico	0.13274639375358227

The following is a simple plot of Los Angeles demonstrated the 1 level of connectivity. (Node names are omitted)



3. TEXT ANALYSIS - WORD2VEC

There are many approaches to this analysis, I have initially looked into using a simple 2-layer network (linear activation + softmax), but because of the bad computation performance in softmax, Hierarchy Softmax may be able to resolve the problem by creating multiple softmax layers into a tree structure. But in this exercise, Negative Sampling is used because of its popularity and effectiveness. There are several open source python library available to allow quick implementation such as gensim, and SparkML library. In this exercise, I have used the keras library to tokenize the documents into words, and keras skipgram and sequences classes to generate the word sequences and negative samples. The network takes 2 inputs (target words and content words) and transformed through an embedded layer into vectors of size 100 (default) , and then merged into a dense network using a matrix dot as the similarity operation. Negative samples are marked with label = 0 by randomly generated matching between target and content pairs from outside the window. For experimental purposes, 3 epoch is

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 1)	0	
input_2 (InputLayer)	(None, 1)	0	
embedding (Embedding)	(None, 1, 100)	3091100	input_1[0][0] input_2[0][0]
reshape_1 (Reshape)	(None, 100, 1)	0	embedding[0][0]
reshape_2 (Reshape)	(None, 100, 1)	0	embedding[1][0]
merge_2 (Merge)	(None, 1, 1)	0	reshape_1[0][0] reshape_2[0][0]
reshape_3 (Reshape)	(None, 1)	0	merge_2[0][0]
dense_1 (Dense)	(None, 1)	2	reshape_3[0][0]

Total params: 3,091,102
Trainable params: 3,091,102
Non-trainable params: 0

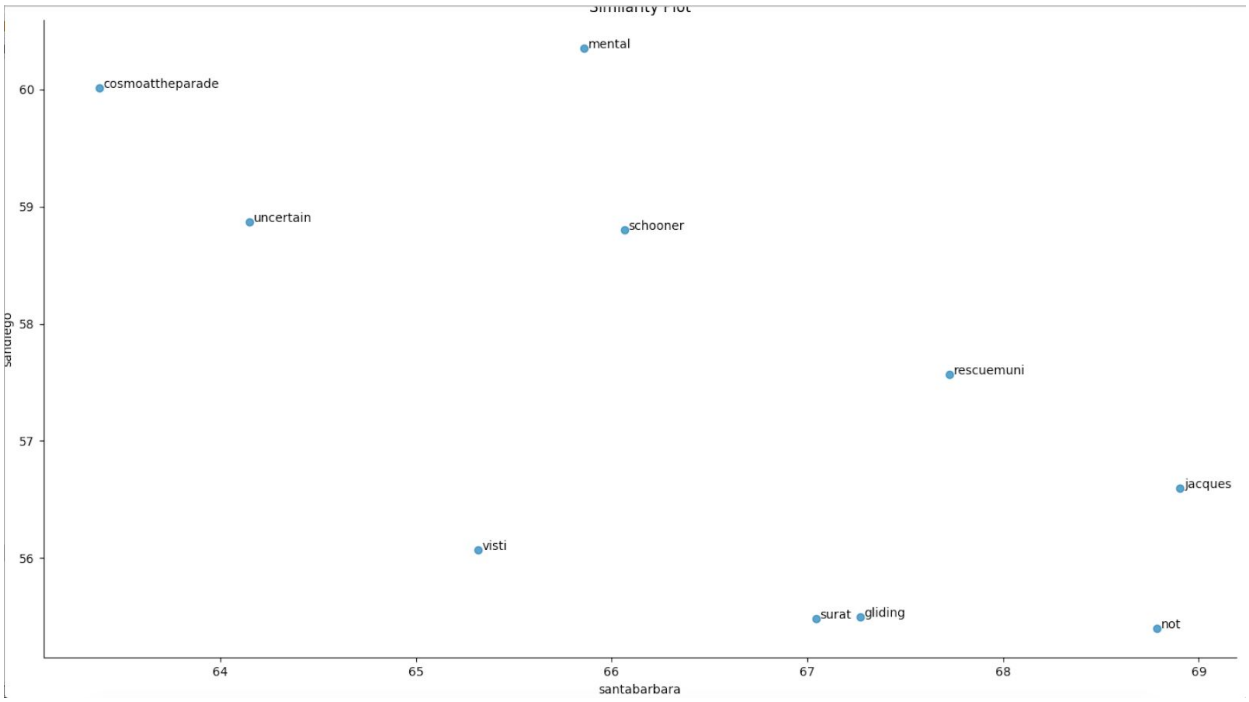
For demo purposes , the hyperparameters such as the window size is being set to 5,

embedding matrix size = 100, epoch = 1 for faster to get the model trained. With 1 epoch we are getting an accuracy of 87% from training data. (Not train/validation split and not considering overfitting in this exercise for simplicity)

```
Epoch 1/1
2018-04-01 13:17:42.830938: I tensorflow/core/platform/cpu_feature_guard.cc:140] Your CPU supports instructions that this TensorFlow binary was not compiled t
o use: AVX2 FMA
545530/545530 [==============================] - 442s - loss: 0.2873 - acc: 0.8726
```

During the prediction, two words (locations) will be inputed. Each location will get be a list of content words with its probabilities of similarity, and then both lists will be ranked together and the top K content words will be output as the result.

By providing “santa barbara” and “san diego” would come back with the top 10 content words.



PERFACE

I would like to express my thank you for offering the opportunity to interview at Credit Suisse Labs. I found the Case Study and the programming exercise are very interesting. Given that I am not a “hard-core” data scientist and still in the progress of learning

various data science techniques, I planned to use this opportunity to pick up some NLP (word2vec) and learned about eigenvector centrality, and I ended up having great time. I hope this coding exercise has demonstrated my ability and qualification to complement the role at your company. I also hope that I have the opportunity to leverage my experience in Big data to help build the infrastructure and the platform for data engineering team.