Keyword Extraction – Detailed Instructions

This program executes an unsupervised keyword extraction model that works on individual documents.

Things to have before running the code.

1) Pre-trained Google News Word2Vec word vectors. These can be downloaded from here. (https://github.com/3Top/word2vec-api)

2) Pre-trained Wikipedia TF-IDF scores. These are in the file pretrained_tfidf.txt.

3) A stop word list. These words are available in stopwords_en.txt

Packages to have before running.

1) Numpy and Scipy – for fast numerical computing

3) NLTK – for word lemmatization and frequency distribution calculation

4) Gensim – for loading and working with Word2Vec word embeddings.

All the installations can be done using pip.

The model has two clustering mechanisms and three different cluster-selection mechanisms. The formal syntax for running this model on command line is as follows.

>>> python KEX.py filename mode number-of-keywords[optional] topic-word[optional] distance-measure[optional]

Filename – the single document placed in a simple text document.

Mode – There are six modes in total. All of these are listed below. The details of these models will be described after discussing all the arguments.

1- Clustering with number of keywords relative to document length.
2- Skip-Agglomeration with number of keyword relative to document length.
3- Clustering with absolute number of keywords requested by the user.
4- Skip-Agglomeration with absolute number of keywords requested by the user.
5- Clustering with absolute number of keywords and a starting word/phrase given by the user.
6- Skip-Agglomeration with absolute number of keywords and a starting word/phrase given by the user.

Number of Keywords – Mode 1 and 2 do not take this argument into consideration. For all the others, this argument is the number of keywords to be retrieved from a document.

Topic-word – This argument works only for modes 5 and 6. If no word/phrase is given, the model automatically switches to mode 3 or 4.

Distance-measure – There are three different distance measures to choose the best cluster.

avg - Average Diameter
w2w – Word-to-word distance
skip – Skip-Agglomerative distance

If none are chosen, default for clustering is set to word-to-word and skip-agglomeration clustering is set of skip-agglomerative distance.

Execution Examples:

1 – Clustering with number of keywords relative to the document length. This mode takes a text file as input, and returns all the prominent keywords from that.

Execution: **python KEX.py clouds.txt 1** (If no distance measure is specified, default cluster selection method is used: word-to-word distance) (Recommended)

(Or)

python KEX.py clouds.txt 1 avg/w2w/skip

2 – Skip- Agglomeration with the number of keywords relative to the document length. This mode takes a text file as input, and returns all the prominent keywords from that.

Execution: **python KEX.py clouds.txt 2** (If no distance measure is specified, default cluster selection method is used: skip-agglomerative distance) (Recommended)

(Or)

python KEX.py clouds.txt 2 avg/w2w/skip

3 - Clustering with absolute number of keywords requested by the user. For retrieving 10 keywords, the execution instruction is given below

Execution: **python KEX.py clouds.txt 3 10** (If no distance measure is specified, default cluster selection method is used: word-to-word distance) (Recommended)

(Or)

python KEX.py clouds.txt 3 10 avg/w2w/skip

4 – Skip – Agglomeration with absolute number of keywords requested by the user. For retrieving 10 keywords, the execution instruction is given below

Execution: **python KEX.py clouds.txt 4 10** (If no distance measure is specified, default cluster selection method is used: skip-agglomerative distance) (Recommended)

(Or)

python KEX.py clouds.txt 4 10 avg/w2w/skip

5 - Clustering with absolute number of keywords and a starting word/phrase given by the user. (If a topic-name is not specified, the model switches to mode 3.)

Execution**: python KEX.py clouds.txt 5 10 clouds**

6 – Skip-Agglomeration with absolute number of keywords and a starting word/phrase given by the user. (If a topic-name is not specified, the model switches to mode 4.)

Execution**: python KEX.py clouds.txt 6 10 clouds**