Gensim: A free python library for topic modelling

Gensim is an open-source library created by Radim Řehůřek for unsupervised topic modeling and natural language processing implemented in Python and Cython, a superset of Python that gives C-like performance written mostly in Python with C-inspired syntax. Some of its main features are streamed parallelized implementations of word2vec(created by Google), fastText(created by Facebook), LSA(latent semantic analysis), LDA(Latent Dirichlet Allocation), tf-idf, etc. Because of python implementation, its features, and well-organized documentation, genism has been one of the most popular choice among data scientists.

With 1 million downloads per week[1], gensim popularity is growing. One of the main reasons for its wide use is that it is easy to implement with python while python being the leading programming language in the world of data science. It is the fastest library for training of vector embeddings – Python or otherwise where the core algorithms in gensim use battle-hardened, highly optimized & parallelized C routines[1]. C-inspired syntax used in gensim make it up for the slower speed, the only weakness of Python compared to other languages. Therefore, gensim has been favorite among data scientists.

Gensim works well with any size of data by leveraging its data-streamed algorithms and making it memory independent. With the data size growing rapidly, the importance of distributed computing system has risen. It is impossible to store a corpus with millions of documents into RAM for processing so data streaming is a must. Gensim is able to extract one document a time from large corpus, process it as needed and corpus doesn't need to be a list, NumPy array, or pandas dataframe as it accepts any objects that, when iterated over, yield documents[2]. Gensim, by allowing to work with larger dataset helps to increase the accuracy of topic modeling and improve NLP models.

Gensim are wildly use for its different vector space model algorithms for topic modeling and other natural language processing task. The documents need to be converted into numerical values to run different NLP algorithms. First, with the use of doc2bow of Dictionary class from corpora library, the documents are vectorize to create bag-of-word representation. The popular algorithm , TF-IDF is implemented using TfidfModel class from models library of gensim on bag-of-words to transform into a different but more meaningful vector where the frequency counts are weighted according to relative rarity of each word in corpus. With the use of data streaming feature, gensim allows to save the trained model to disk, and later load them weather to continue the training on new training documents or to transform new documents based on trained models. These doc2bow class, TfidfModel can be implemented with just a few lines of clean codes.

The output vectors from TfidfModel can be used for different NLP task such as topic modeling. Gensim makes it easier to implement latent semantic analysis (LSA), a NLP technique to analyze relationship between terms by producing a set of concepts related to the documents and terms. LSA, which is more often called as LSI (Latent Semantic Indexing) transforms bag-of-words or TfIdf weighted vector space into a latent space of a lower dimensionality using LsiModel class from Models library. This module of gensim contains several algorithms to build LSI models for

(i) corpora much larger than RAM using distributed computing by creating clusters of machines, (ii) streamed corpora with sequential access, (iii) corpora that cannot be stored temporarily and need one-pass algorithm[3]. With the use of "decay" parameters of LsiModel, gensim allows to gradually "forget" old observations (documents) giving more preference to new ones. This LsiModel also have some parameters that one can check that affect the speed, memory footprint and numerical precision. In addition to this, gensim has a "similarities" module that helps to determine the similarities between different queries which works very well after implementing Lsimodel[4].

Gensim also support another popular transformation for topic modeling called LDA which can use bag-of-words to transform it into topic space of lower dimensionality. LDA is like LSA but topics are interpreted as probability distributions over words. The implementation of LDA including for the distributed system are very similar to LSI model except few notable changes which are very straightforward. However, gensim LDA uses a variational Bayes sampling method which tends to be faster but are less precise as compared to Gibbs sampling which is implement in Mallet (**MA**chine **L**earning for **L**anguag**E** **T**oolkit), a Java-based package put out by UMASS Amherst[5]. However, gensim have a wrapper for Mallet and one can chose to use either one of these with the trade off between speed vs precision.

Some of the popular models algorithm supported by gensim are Random Projections(RP) and Hierarchical Dirichlet Process (HDP). RP is used to reduce vector space dimensionality by approximating Tfidf distances between documents by throwing little randomness and is very memory and CPU friendly. While HDP is a non-parametric Bayesian method to clustering grouped data. However, creator warns that HDP is new addition to gensim and suggests user to use it with caution[6]. Additionally, the integration of pyLDAvis, an interactive web-based visualization tools for topic modeling, with gensim have attracted quite number of data scientists towards gensim.

However, gensim is still adding modules to its library and the creator of gensim has already identified certain areas of the models that need to be improved which are clearly marked in the documentation as TODO list. Gensim creator, Radim has been actively working on gensim and providing continuous improvement along with encouraging other scientists to join the list of contributors and requesting users to give feedback in some of the class implementation.

When compared with other toolkit for topic modeling such as MeTA, gensim seems to be better choice. MeTA is C++ based toolkit and has python binding called metapy but are not updated as regularly as gensim. Metapy is still very much under construction where the last update seems to be on Aug 2018. Data scientists coming from python programming language will find gensim much easier to implement and will have smaller learning curve as compared with MeTA. The documentation of gensim are well documented. There are numerous articles written by data scientists, and other authors talking about the use cases and implementation of gensim compared to any other toolkit for topic modeling. Browsing through several discussion board, reddit post, articles, it is easy to infer that gensim has been quite popular among those professionals and researchers in the field of NLP especially topic modeling.

To put in a nutshell, gensim is one of the most popular python libraries for those who are interested in NLP task such as topic modeling. However, the creator says that he doesn't have ambition to make gensim an all-encompassing framework for NLP but to help NLP practitioners to try out popular topic modelling algorithms on large dataset easily with distributed system and data streaming features and also to facilitate prototyping of new algorithms for researchers. With its well-versed documentation along with a variety of corpora and pretrained models[7] and several learning-oriented lessons, gensim is providing enough resources for anyone interested in the field of topic modeling to learn easily and quickly.

Reference:

[1]https://radimrehurek.com/gensim/index.html

[2] https://radimrehurek.com/gensim/auto_examples/core/run_corpora_and_vector_spaces.html

[3] https://radimrehurek.com/gensim/models/lsimodel.html#module-gensim.models.lsimodel

[4] https://radimrehurek.com/gensim/auto_examples/core/run_topics_and_transformations.html

[5] https://towardsdatascience.com/basic-nlp-on-the-texts-of-harry-potter-topic-modeling-with-latent-dirichlet-allocation-f3c00f77b0f5

[6]https://radimrehurek.com/gensim/auto_examples/core/run_topics_and_transformations.html#sphx-glr-auto-examples-core-run-topics-and-transformations-py

[7]https://github.com/RaRe-Technologies/gensim-data