Technology Review – Gensim Library

Gensim, an abbreviation for generate similarities, is a very popular natural language package for topic modelling in Python. It is proven to be working as one of the most mature NLP libraries through thousands of academic and corporate usages. This review is going to discuss the common use cases, advantages, and potential improvements of Gensim.

Usage of Gensim:

There are plenty of built-in NLP model algorithms in the Gensim package that include but is not limited to API functions for building vectors from documents and words, identifying topics from documents in the corpus, analyzing semantics structures, etc. Some of the most common usage of Gensim functions are described below:

Bag of Words

Bag of words model is a popular algorithm in NLP that transforms document to a vector of integers that represent the number of times a specific word occurred in a document. This is easy to do in Gensim with the help of doc2bow function in the "gensim.corpora.dictionary" class. The function takes a document string and returns a sparse vector that contains (word-id, frequency-count) pairs.

```
genism.corpora.dictionary.doc2bow(new_doc.lower().split())
```

Word2Vec

Word2Vec algorithm computes word embeddings from raw texts to reconstruct linguistic contexts of words. Gensim can help us here to efficiently represent documents as semantic vectors using its Word2Vec function, which takes one of these two model architectures: the Continuous Skip-Gram model and Continous Bag of Words (CBOW) model. The difference is that the SG model uses current word to predict its surrounding words, and the CBOW model predicts the current word from its surrounding words.

```
genism.models.Word2Vec(sentences, min_count=10, vector_size=200)
```

Doc2Vec

Doc2Vec Algorithm is a variation from the Word2Vec algorithm described above, instead of producing semantic vectors for words, it produces a vector representation for each document in the corpus.

```
genism.models Doc2Vec(vector_size=40, min_count=2, epochs=30)
```

TF-IDF (term frequency-inverse document frequency)

The TF-IDF algorithm gives weight to words by TF-IDF importance, as opposed to frequency only. It takes a vector representation with integer values, like the Bag of Words output. It then takes the local component (TF) and global component (IDF) to increase the value of rare features then normalize to unit length. Gensim function TfidfModel helps us to perform this transformation for documents in the corpus easily.

```
genism.models TfidfModel(corpus, normalize=True))
```

LDA (Latent Dirichlet Allocation)
LDA model algorithm is an unsupervised probabilistic topic modelling algorithm that transforms document from Bag-of-Words model space into a topic space, number of topics can be customized using parameters in the function. Gensim function LdaModel helps us to initiate LDA models.

```
genism.models LdaModel(corpus, id2word=dictionary, num_topics=100)
```

LSI (Latent Semantic Indexing)
LSI model algorithm is a topic modelling algorithm takes the same input as LDA model function, which is a vector representation with integer values from the Bag of Words output or Tf-Idf weighted space into latent space. It mainly focuses on dimension reduction to retain the most useful information in lower dimensionality. It is easy to use Gensim function LsiModel helps us to set up LSI models.

```
genism.models LsiModel(tfidf_corpus, id2word=dictionary, num_topics=300)
```

Advantages of Gensim:
Comparing to other libraries of similar functionalities, Gensim stands out with its own advantages that engage users to actively develop NLP applications with Gensim:

1. **Performance**. Gensim has built-in multi-core support for large load of code executions. It utilizes the multiple cores in the CPU system to run code in parallel for optimized performance. Gensim also does not need to reside the whole input corpus fully in Random Access Memory (RAM) which makes it scalable to large scale corpora.

2. **Simplicity**. Gensim is built with unsupervised learning modelling, and therefore does not require costly annotations or hand tagging of documents to save users' effort. It is also platform independent, which means that users can install and run Gensim codes from their preferred platform and possibly leverage the pre-trained models/corpus via the Gensim-data project.

3. **Agility**. All Gensim source code is hosted on Github and maintained by its open-source community under GNU LGPL license , which allows personal and commercial uses to constantly contribute back to the Gensim package and extend the usability of Gensim as new algorithms are developed.

As introduced above, to conclude, Gensim is an extremely powerful library that makes complex NLP algorithms accessible and easy to use for developers. It also handles large-scale text processing more efficiently with the help of incremental RAM usage and parallel computing leveraging multi-cores in your machine. It is continuously evolving with the help of Gensim's user community

References:

Gensim. (2021). Gensim: Topic Modelling for Humans. https://radimrehurek.com/gensim/

Tutorialspoint. (2021). Learn Gensim.

https://www.tutorialspoint.com/gensim/gensim_doc2vec_model.html