Doc2vec or n-grams is used as feature input to the machine learning algorithm.

# Doc2Vec

Doc2vec:

* Doc2vec (also known as: paragraph2vec or sentence embedding) is the modified version of word2vec. The main objective of doc2vec is to convert sentence or paragraph to vector (numeric) form.
* In Natural Language Processing Doc2Vec is used to find related sentences for a given sentence (instead of words in Word2Vec).
* Doc2vec is able to detect relationships among words and understands the semantics of the text.
* Doc2Vec is an unsupervised algorithm that learns fixed-length feature vectors for paragraphs/documents/texts.
* In **Python**, using **Doc2Vec** in **gensim** package.

Link:

1. Good Python example about Doc2vec: <https://medium.com/@mishra.thedeepak/doc2vec-simple-implementation-example-df2afbbfbad5> (if you are unable to open or read it, you can read the pdf file uploaded in the same folder of this file)
2. **Example of how to get the feature vector using doc2vec model and then how to apply it in classifier model**: <https://towardsdatascience.com/implementing-multi-class-text-classification-with-doc2vec-df7c3812824d> (if you are unable to open or read it, you can read the pdf file uploaded in the same folder of this file)
3. Simple Python example about Doc2vec: <https://thinkinfi.com/gensim-doc2vec-python-implementation/>
4. dm ({0,1}, optional) 1: PV-DM, 0: PV-DBOW

<https://medium.com/swlh/sentiment-classification-for-reviews-using-doc2vec-660ba594c336>

<https://joshuakyh.wordpress.com/2017/12/07/distributed-representation-of-anything/#:~:text=Differences%3A,word%20vectors%20in%20PV%2DDM>.

<https://stackoverflow.com/questions/44011706/what-is-different-between-doc2vec-models-when-the-dbow-words-is-set-to-1-or-0>

<https://stackoverflow.com/questions/56323377/which-method-dm-or-dbow-works-well-for-document-similarity-using-doc2vec>

<https://medium.com/swlh/sentiment-classification-for-reviews-using-doc2vec-660ba594c336>

1. Doc2vec library

<https://radimrehurek.com/gensim/models/doc2vec.html>

# N-grams

Consider to use ngrams with n=5 or more.

N-grams:

* N-grams of words/characters represented as a vector (N-grams are overlapping groups of multiple succeeding words/characters in the text)
* In **Python**, either of the two options below is used:
  + **ngrams()** from **NLTK** package
  + **TfidfVectorizer** from **sklearn** with ngram\_range=(2,2). Example: **TfidfVectorizer(ngram\_range=(2,2)**

Link:

1. Example of using function **ngrams** from **NLTK** package: <https://albertauyeung.github.io/2018/06/03/generating-ngrams.html>
2. Simple example of how to use function **ngrams** from **NLTK** package: <https://stackoverflow.com/questions/17531684/n-grams-in-python-four-five-six-grams>
3. **Example of how to use n-grams and then how to apply it to classifier model**: <https://stackoverflow.com/questions/48003907/how-to-train-naive-bayes-classifier-for-n-gram-movie-reviews>
4. Self-bult n-grams: <https://stackoverflow.com/questions/13423919/computing-n-grams-using-python>

Other notes:

* In February 2019, OpenAI started quite a storm through its release of a new transformer-based language model called GPT-2. GPT-2 is a transformer-based generative language model that was trained on 40GB of curated text from the internet. <https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-language-model-nlp-python-code/>