**Machine Learning**

**Natural Language Processing**

**Topic Modeling**

**(Topic Name Extraction)**

**Present by**

**Gena Magdy Ibrahim**

**Topic modeling** is an unsupervised machine learning technique that’s capable of scanning a set of documents, detecting word and phrase patterns within them, and automatically clustering word groups and similar expressions that best characterize a set of documents.

**The Objective of task**

You are provided with a medical related dataset, containing 5k articles. You are required to demonstrate your NLP, data-mining & topic extraction skills to extract the topics that are included within the dataset along with mapping each article to the corresponding topic names. Each document should be matched to the top three most relevant topic names including the matching probability score. Kindly, mention how you came up with the optimum number of topics extracted and a final evaluation statement of your work.

**Model Implementation Steps**

1. Loading Data
2. Data Cleaning
3. Phrase Modeling: Bi-grams
4. Data Transformation: Corpus and Dictionary
5. Base Model: Latent Dirichlet Allocation (LDA) Model
6. Hyper-parameter Tuning
7. Final model
8. Visualize Results

**Step 1 Loading Data**

Dataset related to a medical dataset, containing 5k articles. It has a three columns (ArticleID, Title and Abstract) and don’t have any null values.

**Step 2 Data Cleaning**

1. Drop ArticleID and Title because our task is the objective to extract name topic so they don’t important.
2. Remove punctuation.
3. Convert the sentence to lowercase
4. Tokenize each sentence into a list of words, removing punctuations and unnecessary characters altogether.

**Step 3 Phrase Modeling: Bigrams**

1. Remove Stop Words
2. Form Bigrams that are two words frequently occurring together in the document.
3. Do lemmatization keeping only noun, adj, vb, adv

# Step 4 Data transformation: Corpus and Dictionary

# Latent Dirichlet allocation (LDA) is a particularly popular method for fitting a topic model, and it needs two inputs that are the dictionary and the corpus.

# Step 5 Base Model: Latent Dirichlet allocation (LDA)

# The latent Dirichlet allocation (LDA) is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar.

# C:\Users\original\Downloads\Schematic-of-LDA-algorithm.png

* The task is developed by***Latent Dirichlet Allocation (LDA)***method in the ***python*** using ***gensim*** implementation **(**gensim.models.LdaMulticore) because it uses all CPU cores to parallelize and speed up model training.
* **Compute Topic Coherence Score** as evaluation for the task using ***C\_v***measurethatis based on a sliding window, one-set segmentation of the top words and an indirect confirmation measure that uses normalized point wise mutual information (NPMI) and the cosine similarity.

**When using num\_topics=3, random\_state=100, chunksize=100, passes=10**

**The Coherence Score = 0.3771**

**The topics are ….**

[(0,'0.014\*"use" + 0.008\*"base" + 0.008\*"model" + 0.007\*"method" + 0.005\*"system" + 0.004\*"result" + 0.004\*"provide" + 0.004\*"study" + 0.004\*"process" + 0.004\*"specie"'),

(1,'0.012\*"cell" + 0.008\*"effect" + 0.008\*"study" + 0.007\*"protein" + 0.007\*"increase" + 0.006\*"show" + 0.006\*"high" + 0.006\*"gene" + 0.006\*"use" + 0.005\*"level"'),

(2,'0.018\*"patient" + 0.016\*"study" + 0.011\*"use" + 0.006\*"risk" + 0.006\*"health" + 0.006\*"include" + 0.006\*"high" + 0.005\*"group" + 0.005\*"year" + 0.004\*"treatment"')]

**Step 6 Hyper-parameter Tuning**

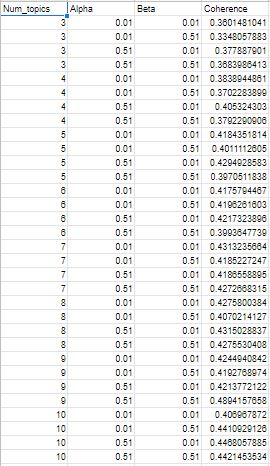
**Model hyperparameters** can be thought of as settings for a machine learning algorithm that are tuned by the data scientist before training, such number of topics, alpha and beta.

We have the coherence score for the LDA model; perform a series of sensitivity tests to help determine the following model hyperparameters

\* Number of Topics

\* Hyperparameter alpha (Document Density)

\* Hyperparameter beta (Word Density)



**Step 7 Final Model**

Depend on below results, we see the best result with optimum number of topics extracted (Num\_topics=9, Alpha= 0.51, Beta=0.51)

**The coherence score = 0.4894**

**The Topics are …**

[(0,'0.005\*"channel" + 0.005\*"formulation" + 0.004\*"oxide" + 0.004\*"iron" + 0.004\*"nanoparticle" + 0.003\*"charge" + 0.003\*"compression" + 0.003\*"release" + 0.002\*"ingredient" + 0.002\*"polyphenol"'),

(1,'0.020\*"cell" + 0.012\*"protein" + 0.008\*"effect" + 0.006\*"expression" + 0.006\*"increase" + 0.006\*"show" + 0.006\*"activity" + 0.006\*"induce" + 0.006\*"study" + 0.005\*"mechanism"'),

(2,'0.005\*"smoking" + 0.005\*"smoker" + 0.003\*"smoke" + 0.003\*"tobacco" + 0.002\*"cessation" + 0.002\*"cigarette" + 0.001\*"nicotine" + 0.001\*"exudative" + 0.001\*"fry" + 0.001\*"apoa"'),

(3,'0.029\*"patient" + 0.019\*"study" + 0.013\*"use" + 0.010\*"group" + 0.009\*"high" + 0.009\*"risk" + 0.008\*"treatment" + 0.007\*"include" + 0.007\*"year" + 0.007\*"age"'),

(4, '0.015\*"use" + 0.008\*"model" + 0.007\*"base" + 0.007\*"study" + 0.007\*"method" + 0.006\*"result" + 0.005\*"high" + 0.005\*"provide" + 0.005\*"specie" + 0.004\*"different"'),

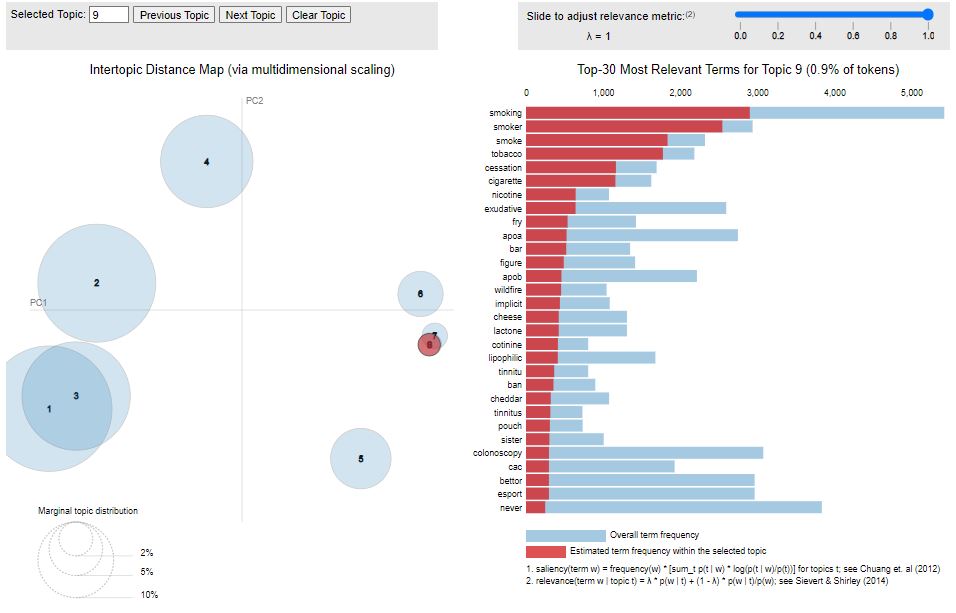
(5, '0.012\*"health" + 0.012\*"study" + 0.009\*"use" + 0.006\*"care" + 0.005\*"intervention" + 0.005\*"research" + 0.005\*"experience" + 0.004\*"participant" + 0.004\*"impact" + 0.004\*"conduct"'),

(6, '0.011\*"case" + 0.011\*"surgery" + 0.008\*"patient" + 0.008\*"surgical" + 0.008\*"implant" + 0.006\*"bone" + 0.006\*"present" + 0.005\*"perform" + 0.005\*"complication" + 0.005\*"follow"'),

(7, '0.018\*"infection" + 0.015\*"virus" + 0.012\*"genetic" + 0.012\*"gene" + 0.008\*"sequence" + 0.008\*"viral" + 0.007\*"antibody" + 0.007\*"genome" + 0.006\*"host" + 0.006\*"isolate"'),

(8, '0.005\*"music" + 0.004\*"facial" + 0.003\*"pedestrian" + 0.003\*"vehicle" + 0.002\*"musical" + 0.002\*"listen" + 0.002\*"crosswalk" + 0.002\*"postural" + 0.001\*"ssp" + 0.001\*"sit"')]

**Step 8 Visualize Result**

****

**Tools**

* Programming Language: **Python.**
* IDE: **jupyter-lab and Google Colab.**
* Libraries: **pandas, nltk, gensim, spacy and pyLDAvis.**