Goldman Sachs ESG Data Challenge

Team NFT

Introduction

- Common ESG Reporting Frameworks
  - Carbon Disclosure Project (CDP)
  - Climate Disclosure Standards Board (CDSB)
  - Global Reporting Initiative (GRI)
  - International Integrated Reporting Council (IIRC)
  - Sustainability Accounting Standards Board (SASB)

- Challenge
  - Despite the availability of ESG Reporting Frameworks, there are no specific format for representing the information.
  - Based on the availability of pdf documents, we would attempt to discover the “hidden” topics from the sentences and words. With the topics discovery, we would also seek to associate ESG contexts to these topics with the help of machine learning model and/or domain knowledge.
  - Infer topics discovery and Multi-class classification on unseen documents

Motivations

Extract Environmental, Social, and (Corporate) Governance (ESG) data from public sources and transform unstructured data to well-curated data with a generic ESG data model.

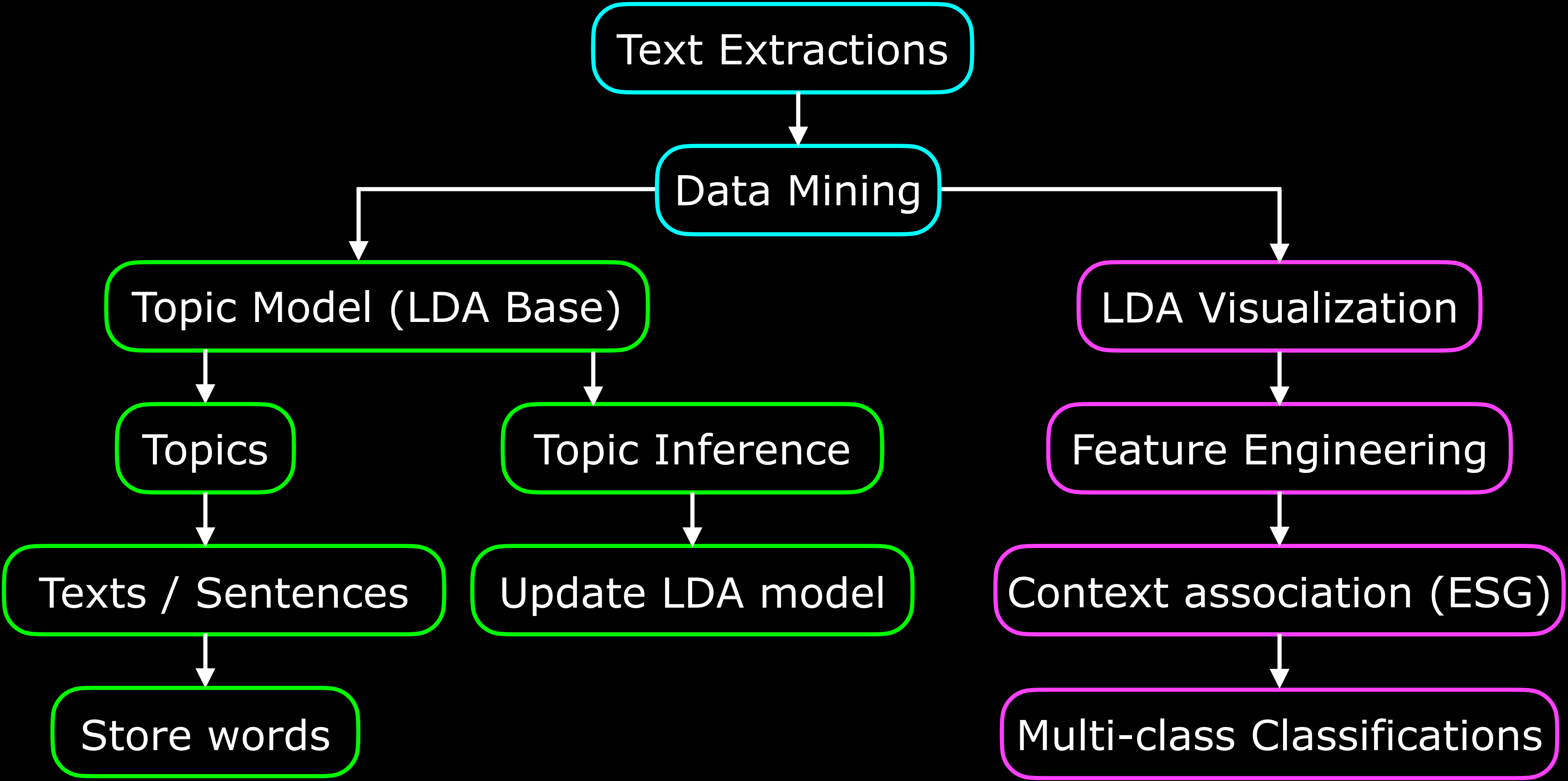
High-level solution approach :

1. Extract texts from documents or corpus repository.
2. Categorise words into numerous topics via unsupervised learning.
3. Find topic model with highest coherence.
4. Inference on unseen documents and improve trained model.
5. Store words from derived topics for analysis and computations.
6. Using topic vectors and feature engineering from topic model.
7. Using visualisation and domain knowledge for context association.
8. Perform Multi-class Classifications.

Proof-Of-Concept Environment

- Kaggle Notebook
- Python 3.7.x
- Python Libraries
  - pdfplumber / pdftotext
  - WordCloud
- Machine Learning Libraries
  - gensim
  - nltk
  - pyLDAvis
  - spacy
  - sklearn
  - lightgbm
- Sample Dataset
  Neural Information Processing Systems (NIPS) is one of the top machine learning conferences in the world.   Extracted text for all NIPS papers to date (ranging from the first 1987 conference to the current 2016 conference).

Proposed Solution Functionalities



Text Extractions with Preprocessing

Process flow :

1. Extract document texts from the raw PDF files deposited in a pre-defined directory (eg. '/kaggle/input/nips-papers/')

2. Store the extracted texts in CSV formatted file together with other meta data information in an extracted document repository for detailed analysis and processing

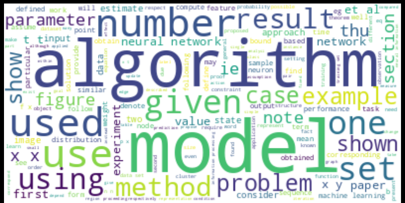
3. Obtain words of each text with gensim simple_preprocess library

4. Remove stop words with ntlk ‘stopwords’ package with own extension (eg. author names)

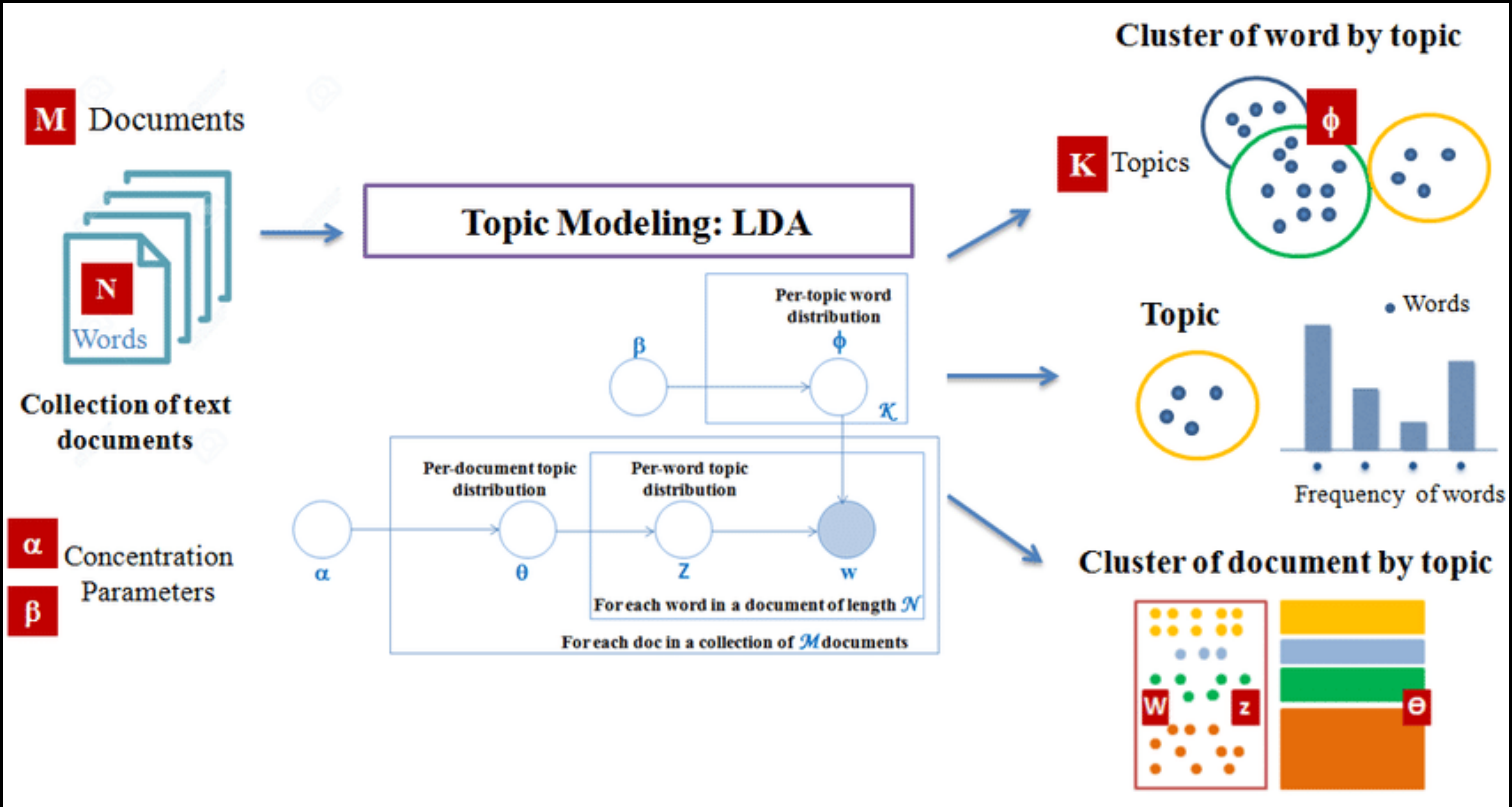
Text Mining

Process flow :

1. Use a regular expression to remove any punctuations and lowercase text.
2. Visualise with WordCloud
3. Transform the textual data in a format to serve as an input for training a Latent Dirichlet Allocation (LDA) model.
   - Tokenising the text and removing ‘stopwords’
   - Convert the tokenised object into a corpus and dictionary
4. Construct Bigram and Trigram Models
   - Bigrams and Trigrams with two important arguments to Phrases model (min_count and threshold)
5. Perform text lemmatisation keeping only nouns, adjectives, verbs, adverbs
6. Data transformation
   - Create Dictionary with gensim corpora library with optional filtering of tokens
   - Create Corpus with lemmatised text by converting document (a list of words) into the bag-of-words format = list of (token_id, token_count) 2-tuples. Each word is assumed to be a tokenised and normalised string



Latent Dirichlet Allocation (LDA)


M
Documents
N
Words
Collection of text documents
α
β
Concentration Parameters
Topic Modeling: LDA
Per-topic word distribution
β
ϕ
K
Per-document topic distribution
Per-word topic distribution
α
θ
z
w
For each word in a document of length N
For each doc in a collection of M documents
Cluster of word by topic
K
Topics
ϕ
Topic
Words
Frequency of words
Cluster of document by topic
W
z
θ

Topic Modelling

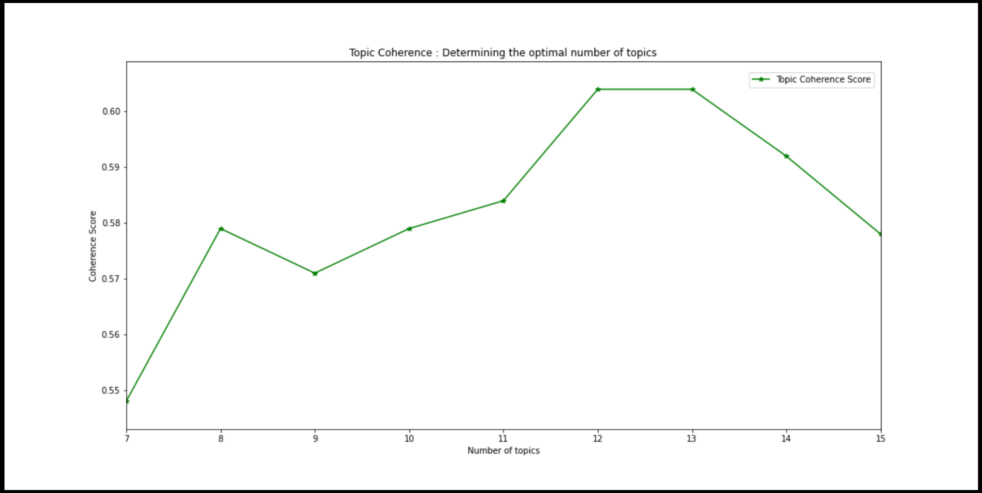
- Latent Dirichlet Allocation (LDA)
  - Dimensionality Reduction
  - Unsupervised Learning
  - Tagging (Topics)
  - LdaMulticore (Use all CPU cores to parallelise and speed up model training)
  - Suggested Topics (11-13) with High Coherence (0.568 - 0.612)

- Hierarchical Dirichlet Process (HDP)
  - Unsupervised Learning infers topics from the data via Online Variational Bayes Inference
  - Suggested 20 topics with multiple trainings (over 100 or 6,560 documents)
  - Low Coherence Score (approx. 0.218)

- Hyperparameters Tuning (num_topics, chunksize, passes)
  - LDA model training with range of topics (7-15)
  - Best Coherence Score (approx. 0.608) with 11-13 topics (LDA is a probabilistic model so runs may have slightly different results)

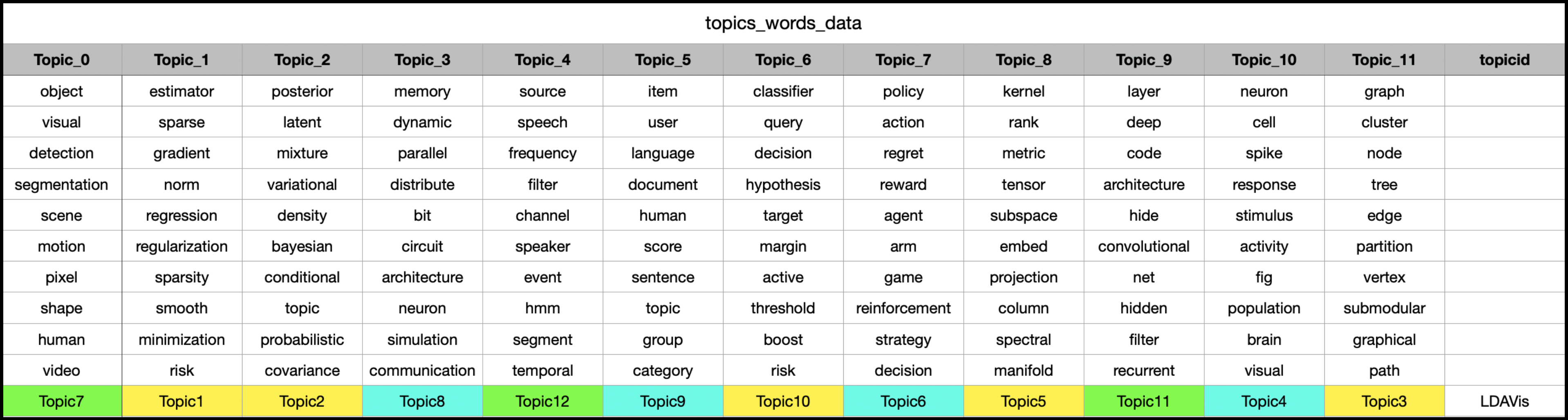
Topic Modelling

- Generalisation
  - LDA is based on a Bayesian statistics which uses dirichlet priors for the document-topic and word-topic distributions, giving a better generalisation without overfitting the data especially for small datasets
  - Inference on unseen sampled texts and document
- Model Improvement
  - Continual model update to avoid data drifts over time, maintain relevancy



Store Topic words

- Store the top 10 (user-define) topic words
  - Use the baselined LDA model to get all the top 10 words for each of the topics (optimal with highest coherence score LDA model).
  - Save the retrieved topics words in a CSV formatted file for future analysis and comparisons.



topics_words_data

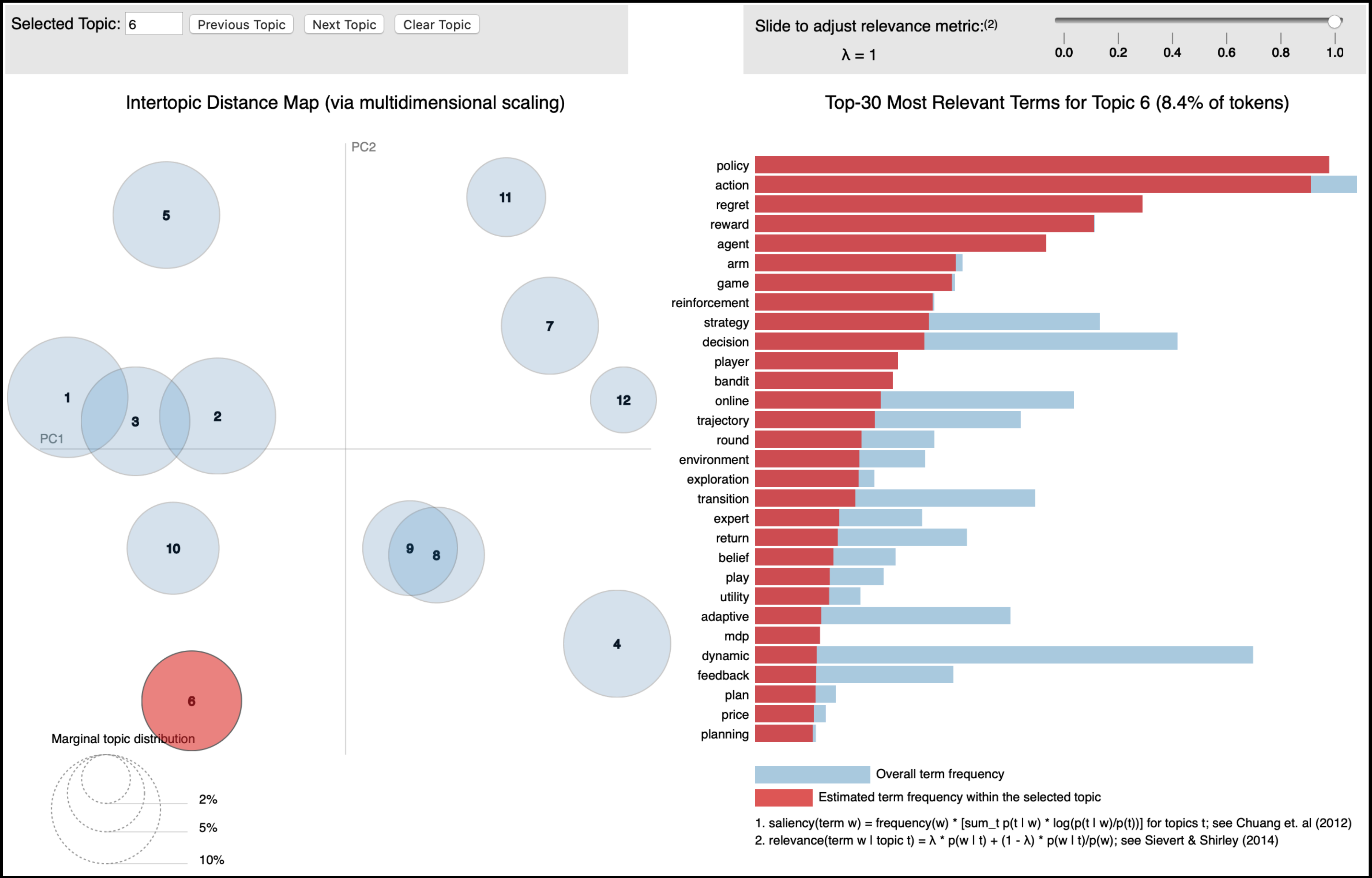
Topic_0	Topic_1	Topic_2	Topic_3	Topic_4	Topic_5	Topic_6	Topic_7	Topic_8	Topic_9	Topic_10	Topic_11	topicid
object	estimator	posterior	memory	source	item	classifier	policy	kernel	layer	neuron	graph	
visual	sparse	latent	dynamic	speech	user	query	action	rank	deep	cell	cluster	
detection	gradient	mixture	parallel	frequency	language	decision	regret	metric	code	spike	node	
segmentation	norm	variational	distribute	filter	document	hypothesis	reward	tensor	architecture	response	tree	
scene	regression	density	bit	channel	human	target	agent	subspace	hide	stimulus	edge	
motion	regularization	bayesian	circuit	speaker	score	margin	arm	embed	convolutional	activity	partition	
pixel	sparsity	conditional	architecture	event	sentence	active	game	projection	net	fig	vertex	
shape	smooth	topic	neuron	hmm	topic	threshold	reinforcement	column	hidden	population	submodular	
human	minimization	probabilistic	simulation	segment	group	boost	strategy	spectral	filter	brain	graphical	
video	risk	covariance	communication	temporal	category	risk	decision	manifold	recurrent	visual	path	
Topic7	Topic1	Topic2	Topic8	Topic12	Topic9	Topic10	Topic6	Topic5	Topic11	Topic4	Topic3	LDAVis

Topic Inference

- Sample unseen texts
  - Verifications of model with crafted sampled texts
  - Use the created corpora dictionary (with doc2bow) to convert sample texts into the bag-of-words (BoW) format = list of (token_id, token_count) tuples
  - Baselined LDA model able to infer topic with high probability (approx 59.32% with 4 crafted matching words out of the top 10 topic words)

- Unseen pdf document (content is related to Reinforcement learning)
  - Text preprocessing by removing punctuations and convert to lowercase
  - Generate words remove stop words, create bigrams and lemmatised the words
  - Use the created corpora dictionary (with doc2bow) to convert sample texts into the bag-of-words (BoW) format = list of (token_id, token_count) tuple
  - Baselined LDA model able to infer topic with high probability (approx 76.12%)

LDA Visualisations (Interactive)



Selected Topic: 6

Slide to adjust relevance metric:(2) λ = 1

Intertopic Distance Map (via multidimensional scaling)

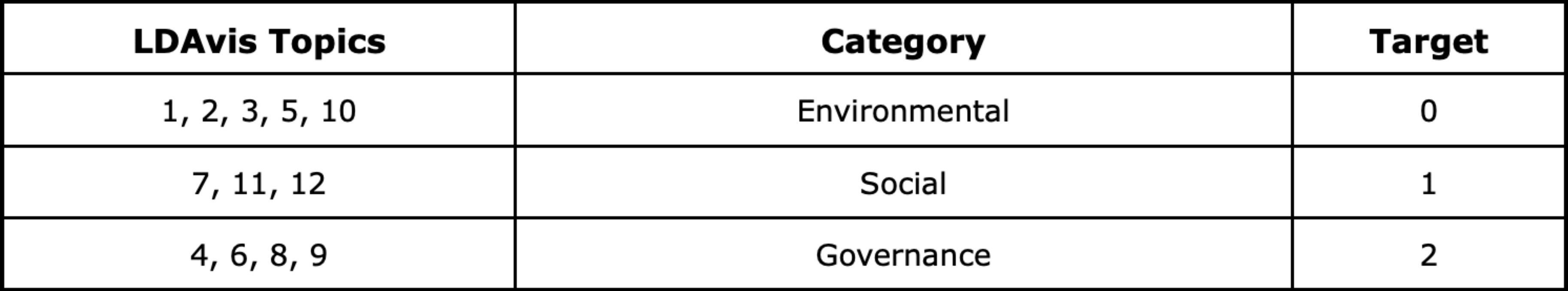
Top-30 Most Relevant Terms for Topic 6 (8.4% of tokens)

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Feature Engineering and Context Association

- Feature Engineering
  - Create topic vectors from inferred topics of all the documents
  - Create new feature based on the total number of characters in the extracted texts/sentences from each document

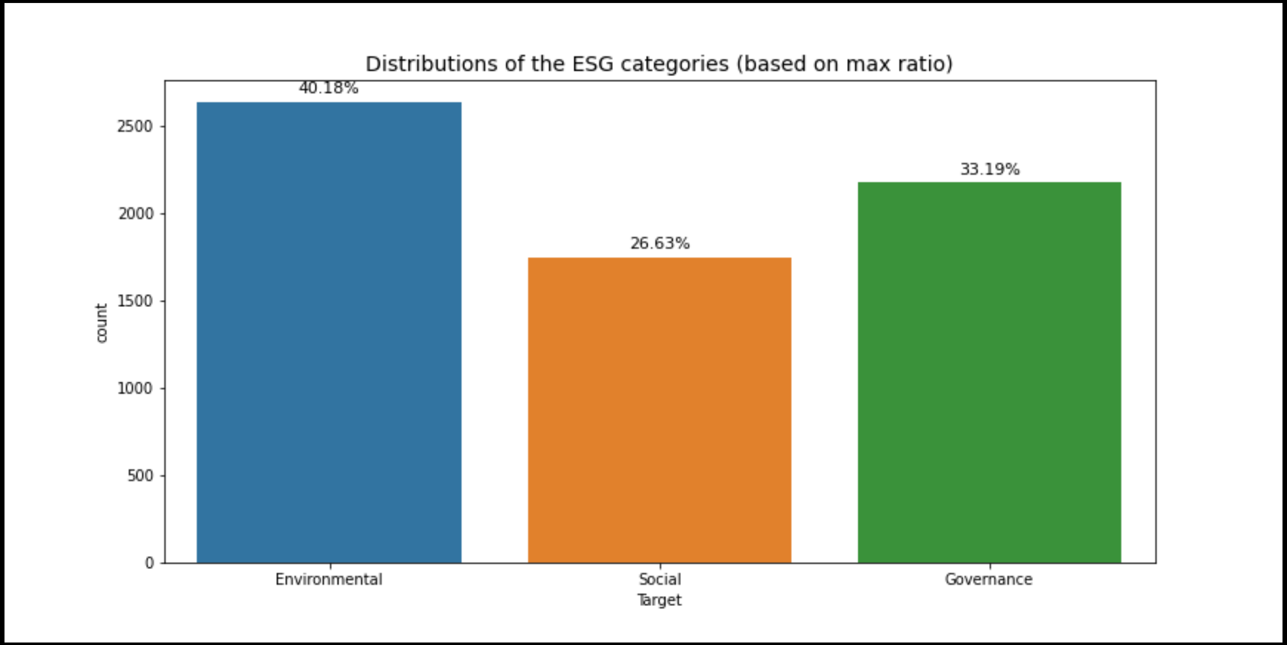
- Context association with Data Labelling (target)
  - Based on the LDA visualisation and domain knowledge, we can label the documents so that they can be used for training a document classifier
  - LDAvis Topics are different from ‘topicid’ from LDA ‘get_document_topics’
  - Group and save respective topic words into ESG categories and it can be used to overcome cold-start (without ESG categorical information)



LDAvis Topics	Category	Target
1, 2, 3, 5, 10	Environmental	0
7, 11, 12	Social	1
4, 6, 8, 9	Governance	2

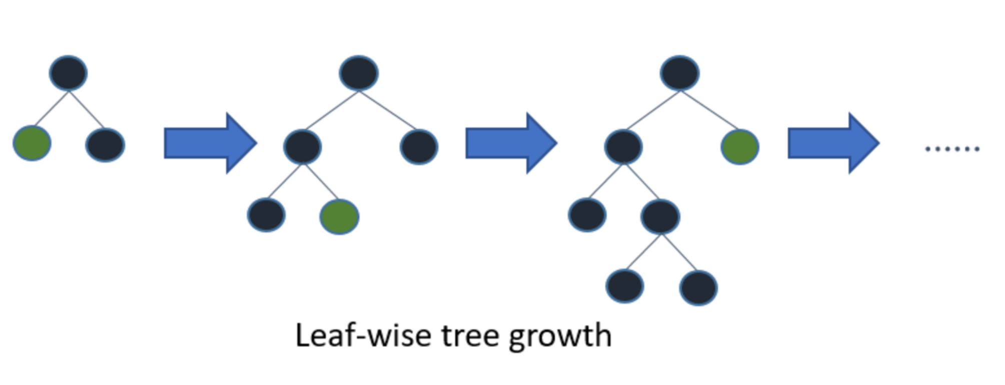
Feature Engineering and Context Association

- Context association with Data Labelling (target)
  - Based on the provided ESG Category words (acquired from domain expertise with provided ESG word data file or learnt from LDA model)
  - Assume top 30 (can be treated as hyper-parameter) lemmatised words from unseen document and used for comparison with ESG words
  - The inferred ESG category will be based on the highest matching ratio between the lemmatised words and provided ESG words



Multi-class Classifications

- Light Gradient Boosting Model
    - Faster training speed and higher efficiency
    - Lower memory usage
    - Better accuracy
    - Support of parallel, distributed, and GPU learning
    - Capable of handling large-scale data
    - Train-test-spilt
    - Hyperparameter tuning (learning rate, max_depth, epochs)
    - High Precision score (approx. 97.47%)



Future Improvements

1. Preprocessing texts by removing unnecessary characters, unicode characters, headers and footers (definable by user).
2. Perform structured analysis on the suggested words to extend stop words for removal (depending on the domain area).
3. Continuous improvement on the topic model with updates to minimise data drifts, maintain model relevancy.
4. Explore other models (Probabilistic Latent Semantic Analysis, Probabilistic Latent Semantic Analysis or lda2vec) and used them for comparisons with LDA via gensim libraries.
5. Analyse and filter words before storing from derived topics
6. Compute distances (Hellinger, Kullback-Leibler, Jaccard) between each topic clusters and group based on distances (centroids).
7. Improve on the ESG word data bank with more relevant documents.
8. Perform feature engineering to improve ESG predictions.
9. Optimal hyperparameters for the Multi-class Classifier.

References

[1] Topic Modelling : A Deep Dive Into LDA, Hybrid-LDA, And Non-LDA Approaches (https://lazarinastoy.com/topic-modelling-lda/)

[2] Gensim Topic Modelling : Parallelized Latent Dirichlet Allocation (https://radimrehurek.com/gensim/models/hdpmodel.html)

[3] Gensim Topic Modelling : Distance Metrics (https://radimrehurek.com/gensim_3.8.3/auto_examples/tutorials/run_distance_metrics.html)

[4] Feature Engineering for Machine Learning, by Alice Zheng, Amanda Casari

[5] LightGBM documentation (https://lightgbm.readthedocs.io/en/latest/)

[6] What is LightGBM, How to implement it? How to fine tune the parameters? (https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc)

[7] Introduction to Machine Learning (2nd Ed.), by Ethem Alpaydin, The MIT Press, 2010

[8] Introduction to Data Mining, by Pang-Ning Tan, Michael Steinbach, and Vipin Kumar, Addison Wesley, 2005.

Questions & Answers

Thank you