Text Summarization and Paraphrase Generation

Team members

Sanika Katekar
Praveen Reddy
Soham Sajekar

Dataset- CNN-Daily Mail

Training pairs

Dataset consists of
287,226 training pairs

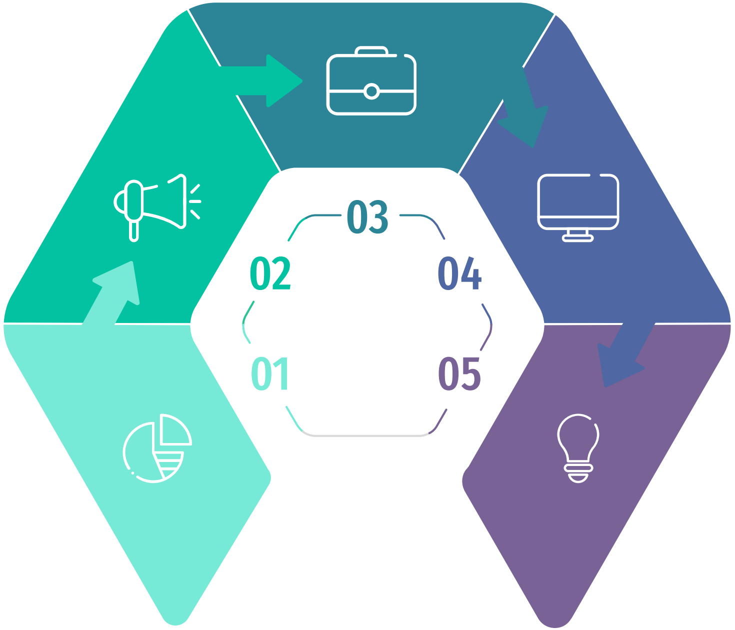
In use: 25000 training articles

Highlights

A separate column with summaries for every article.

News Articles

300,000 unique news articles from CNN and Daily mail.



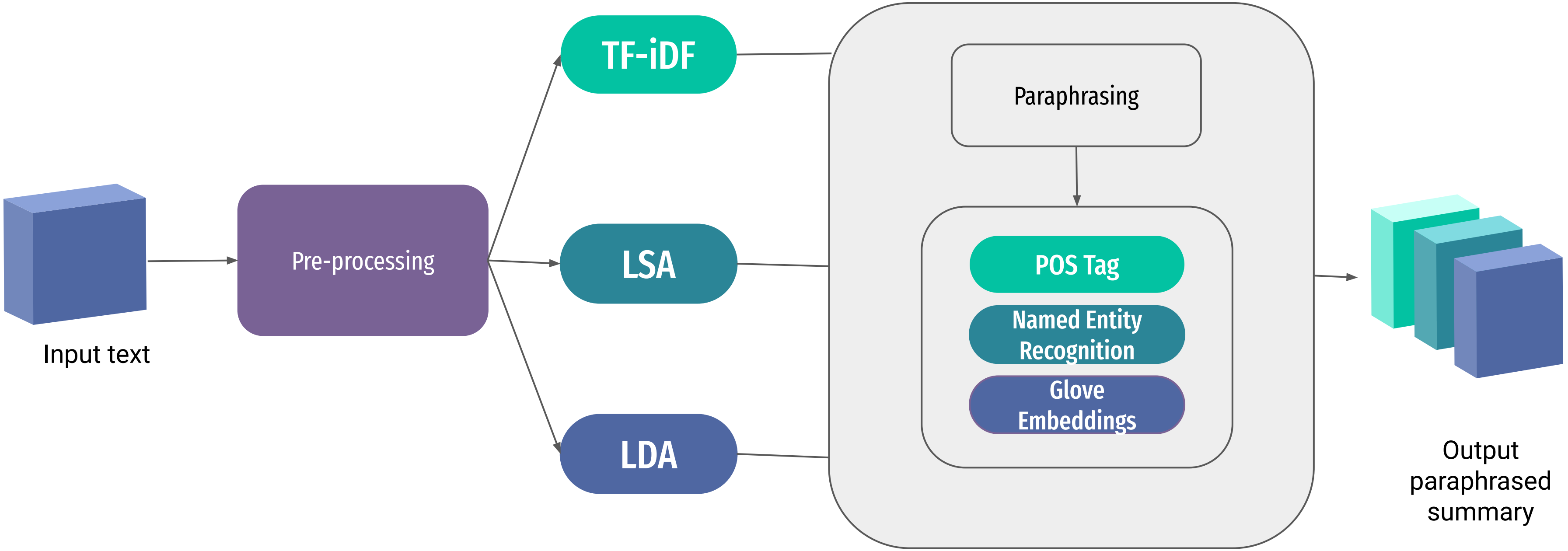
Validation and test sets

13,368 validation pairs and 11,490 test pairs

Extractive & Abstractive

Works for both extractive and abstractive type summarization

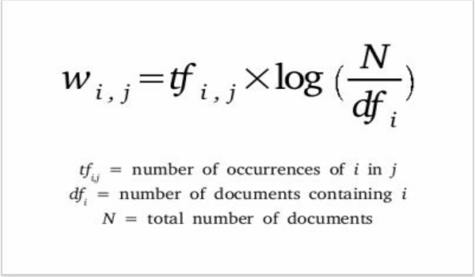
Project Pipeline



TF-IDF: Term Frequency-Inverse Document Frequency

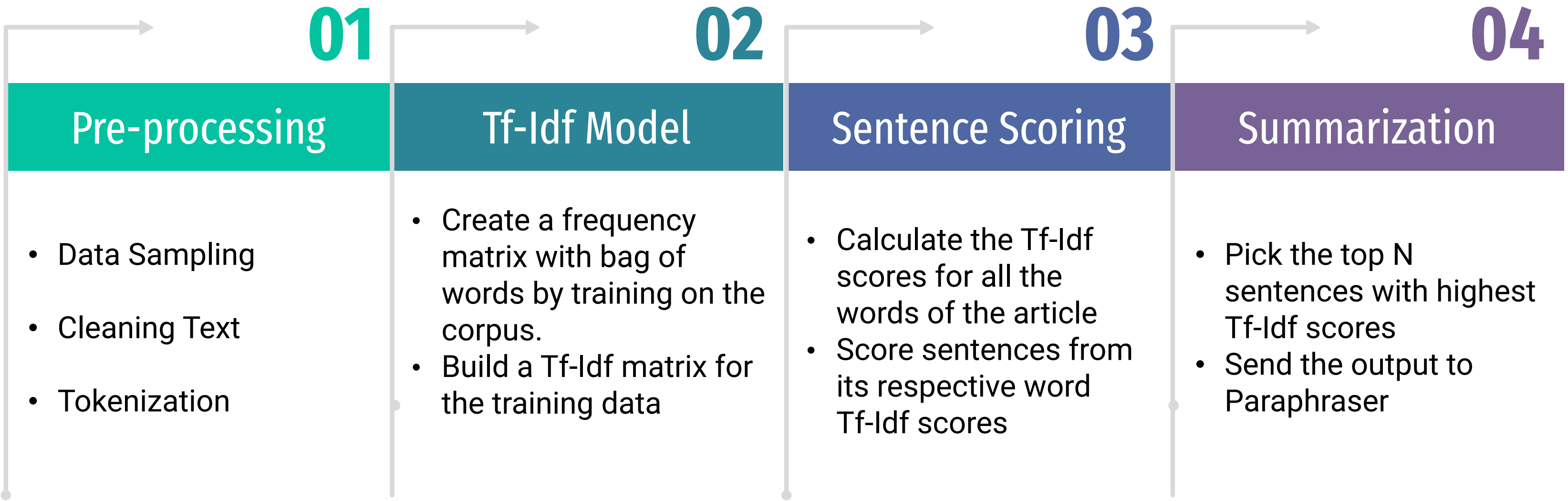
TF: Count of occurrence of the word in the document

DF: Count of occurrence of the word in the corpus



$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

tf_{ij} = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents



01

Pre-processing

- Data Sampling
- Cleaning Text
- Tokenization

02

Tf-Idf Model

- Create a frequency matrix with bag of words by training on the corpus.
- Build a Tf-Idf matrix for the training data

03

Sentence Scoring

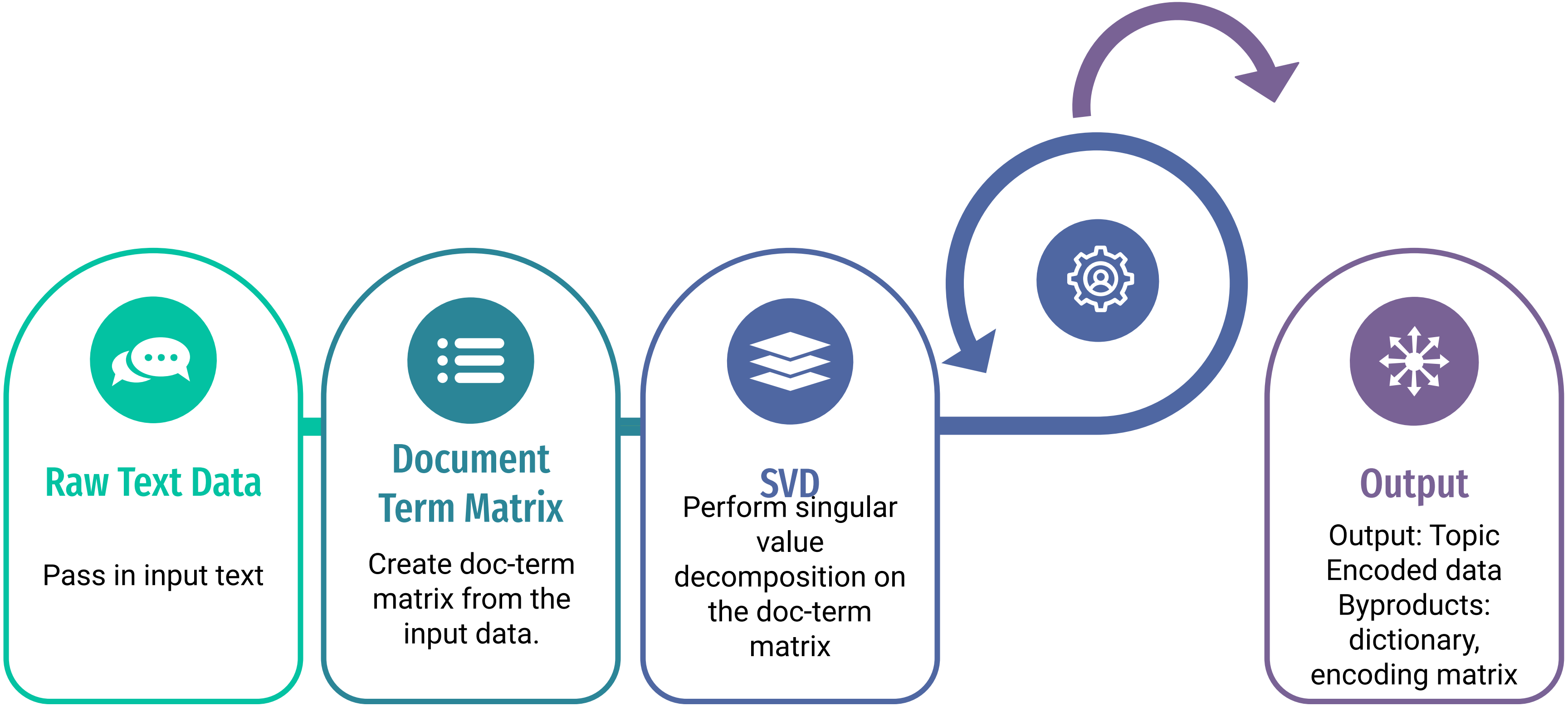
- Calculate the Tf-Idf scores for all the words of the article
- Score sentences from its respective word Tf-Idf scores

04

Summarization

- Pick the top N sentences with highest Tf-Idf scores
- Send the output to Paraphraser

Latent Semantic Analysis



Raw Text Data

Pass in input text

Document Term Matrix

Create doc-term matrix from the input data.

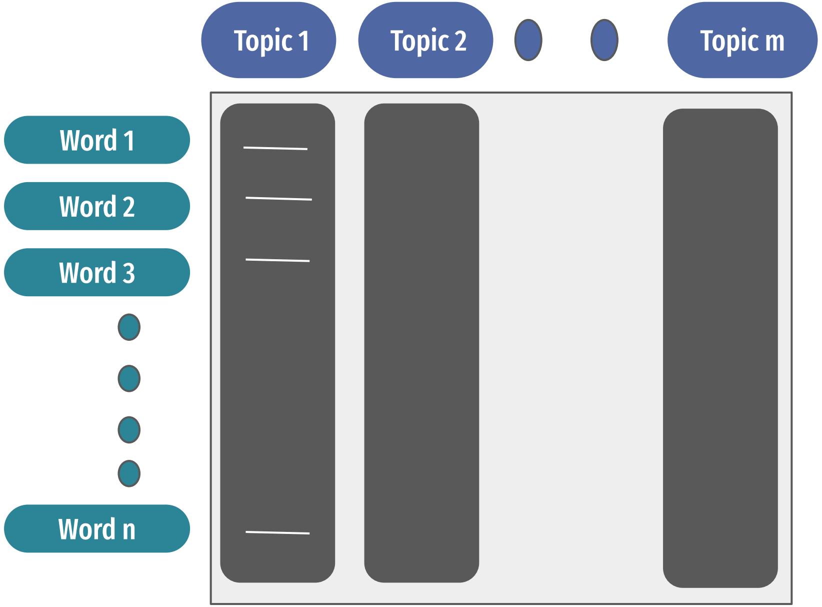
SVD

Perform singular value decomposition on the doc-term matrix

Output

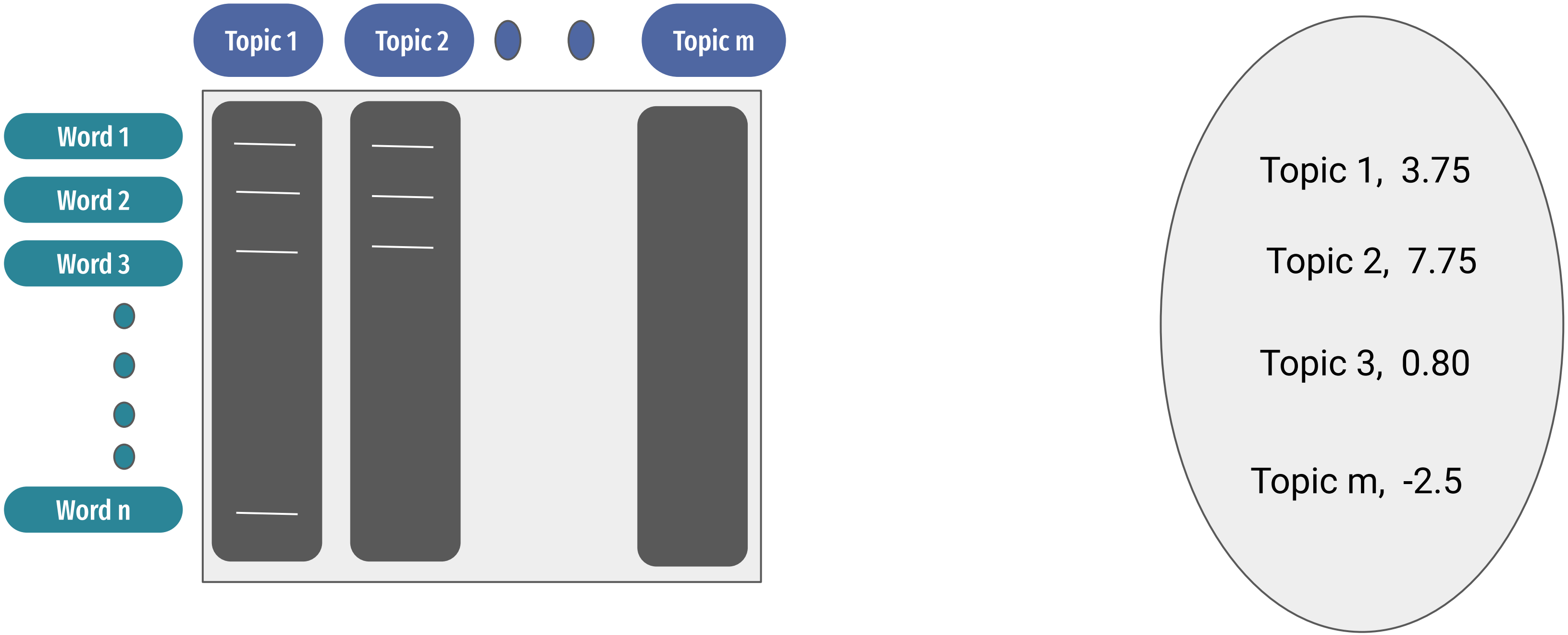
Output: Topic Encoded data
Byproducts: dictionary, encoding matrix

Sentence Ranking



Encoding Matrix

Sentence Ranking



Topic 1

Topic 2

Topic m

Word 1

Word 2

Word 3

Word n

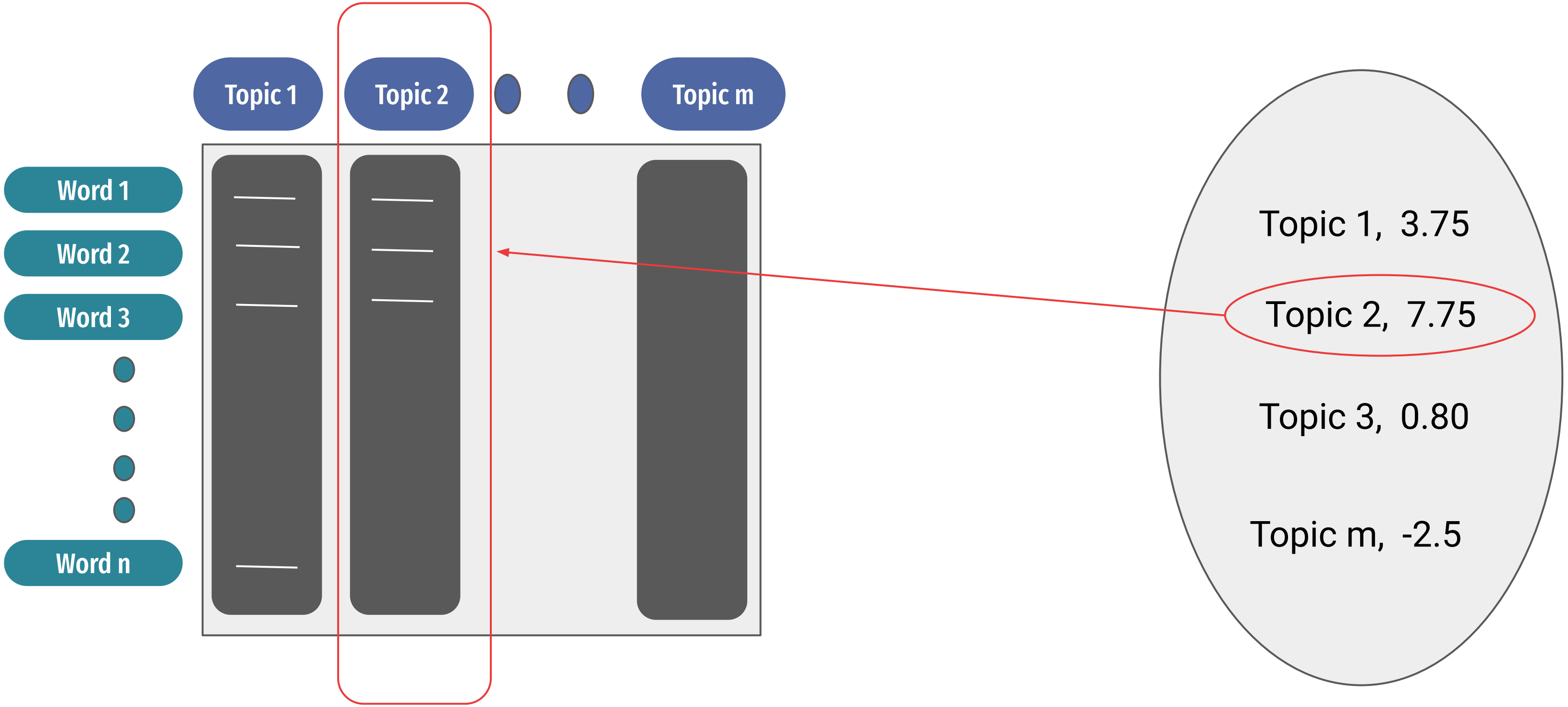
Topic 1, 3.75

Topic 2, 7.75

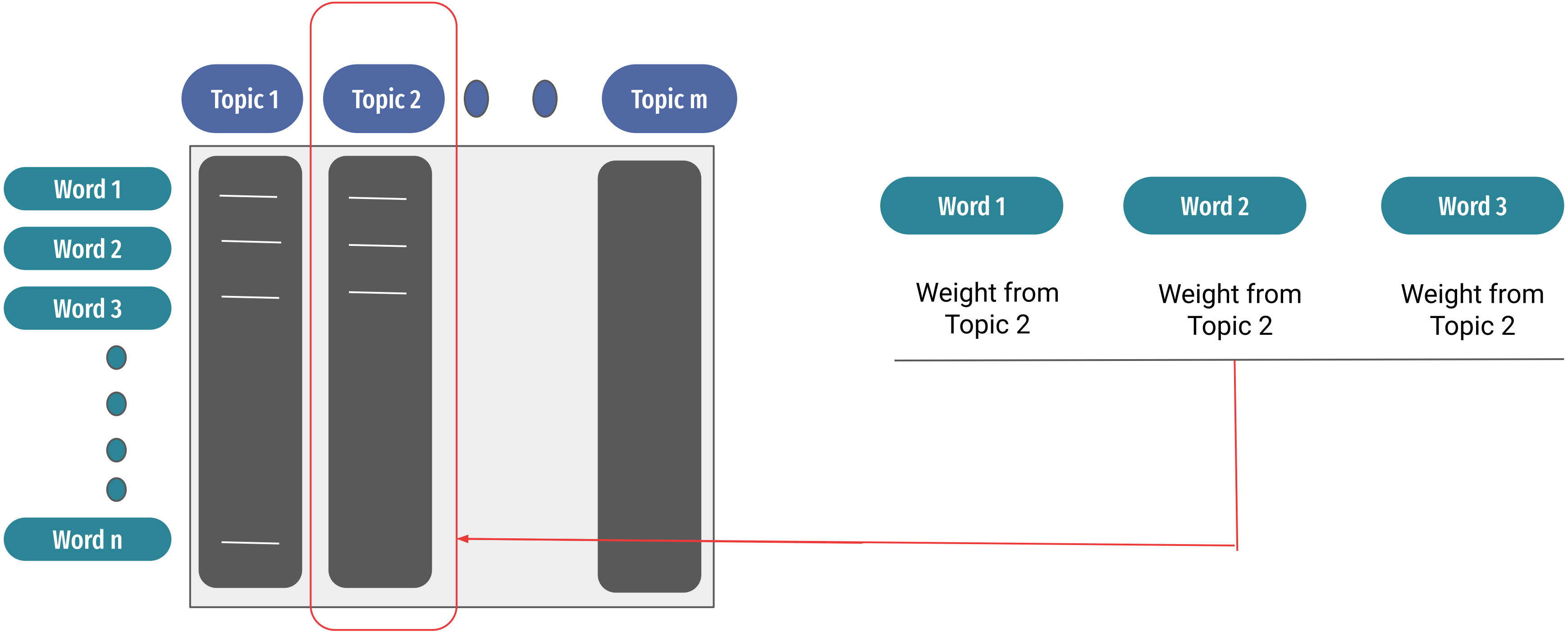
Topic 3, 0.80

Topic m, -2.5

Sentence Ranking

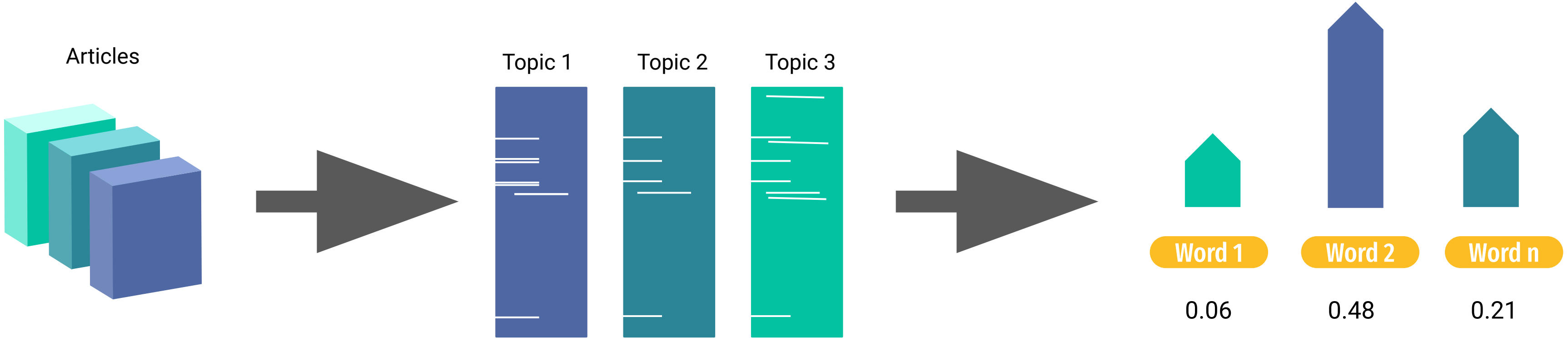


Sentence Ranking



Latent Dirichlet Allocation

```
from gensim.models.ldamodel import LdaModel
```



Generates a probability distribution of words for each topic

Algorithm

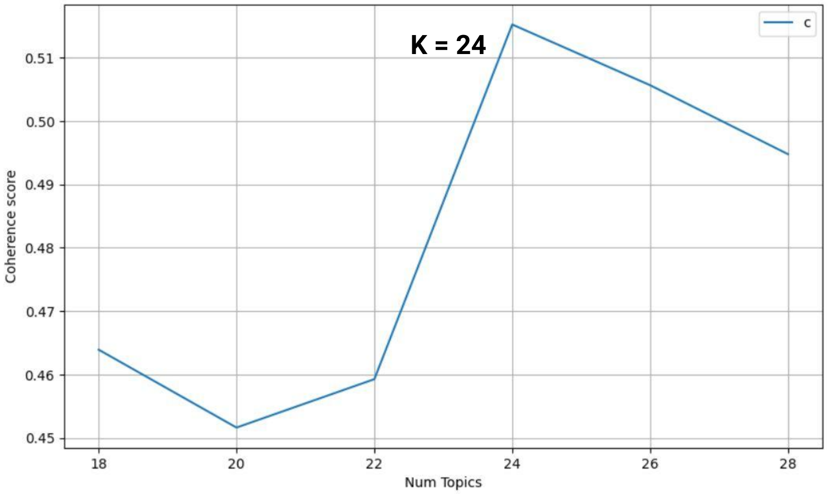
1. Go through each word in article and randomly assign it to any one topic
2. Calculate **p(topic t | document d)** and **p(word w| topic t)**
3. Update probability of word in topic: **p(word w with topic t) = p(topic t | document d) * p(word w | topic t)**

Choosing number of topics for models

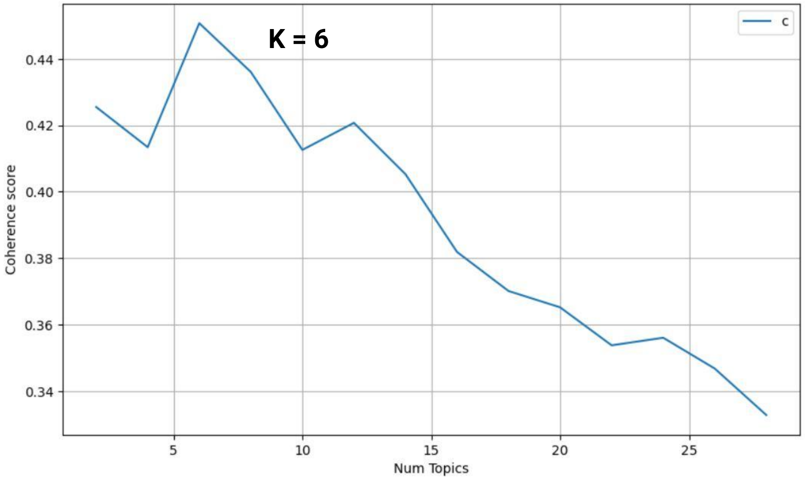
What is topic coherence?

measures the degree of semantic similarity between high scoring words in the topic

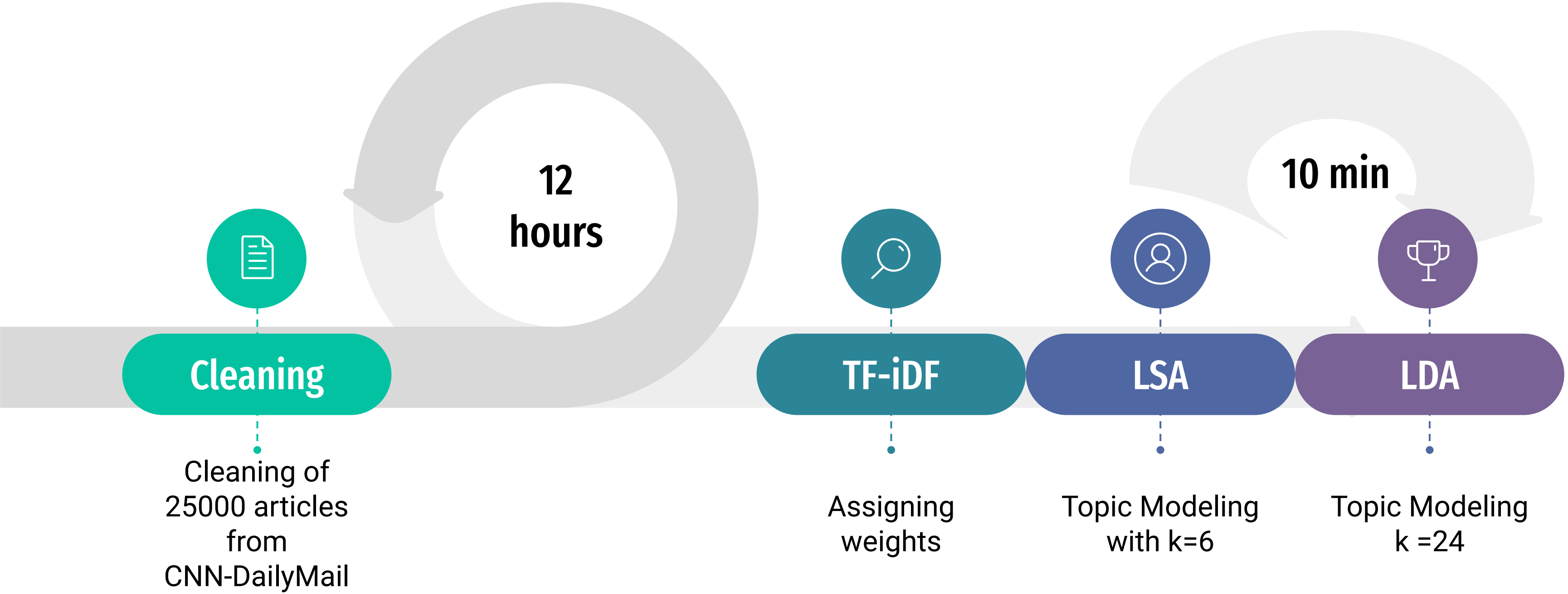
LDA


K = 24

LSA


K = 6

Model Training



Cleaning of 25000 articles from CNN-DailyMail

Assigning weights

Topic Modeling with k=6

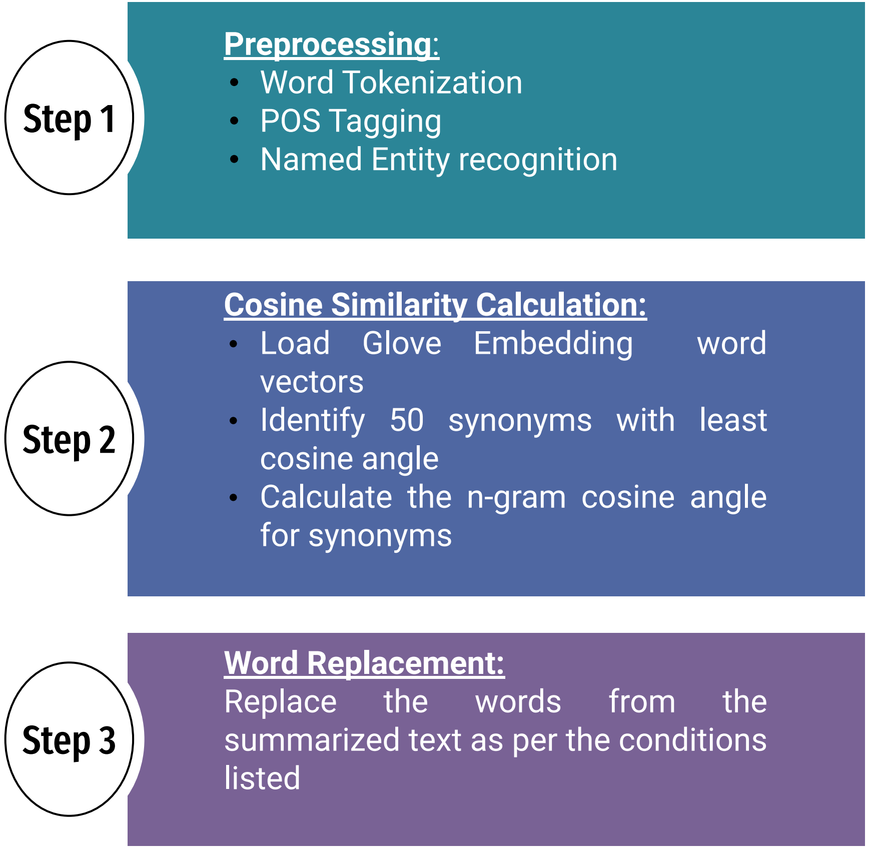
Topic Modeling k =24

Paraphraser

This is a synonym replacement paraphraser with least cosine angle between its word embedding vectors

Word Replacement Conditions:

- Do not replace for stopwords, punctuations, named entities, measurable quantities, and its units
- Do not replace for the following POS tags: 'NN', 'CD', 'RB', 'MD', 'VBN', 'VBD', 'NNP', 'NNPS'
- Replace the synonym with least Trigram cosine angle for the other POS tags by giving first preference for the same POS tag synonym



Step 1

Preprocessing:

- Word Tokenization
- POS Tagging
- Named Entity recognition

Step 2

Cosine Similarity Calculation:

- Load Glove Embedding word vectors
- Identify 50 synonyms with least cosine angle
- Calculate the n-gram cosine angle for synonyms

Step 3

Word Replacement:

Replace the words from the summarized text as per the conditions listed

ROUGE Evaluation Matrix

Rouge 1

Counts the number of overlapping units

Rouge 2

bigram count

Tot count from ref.summary

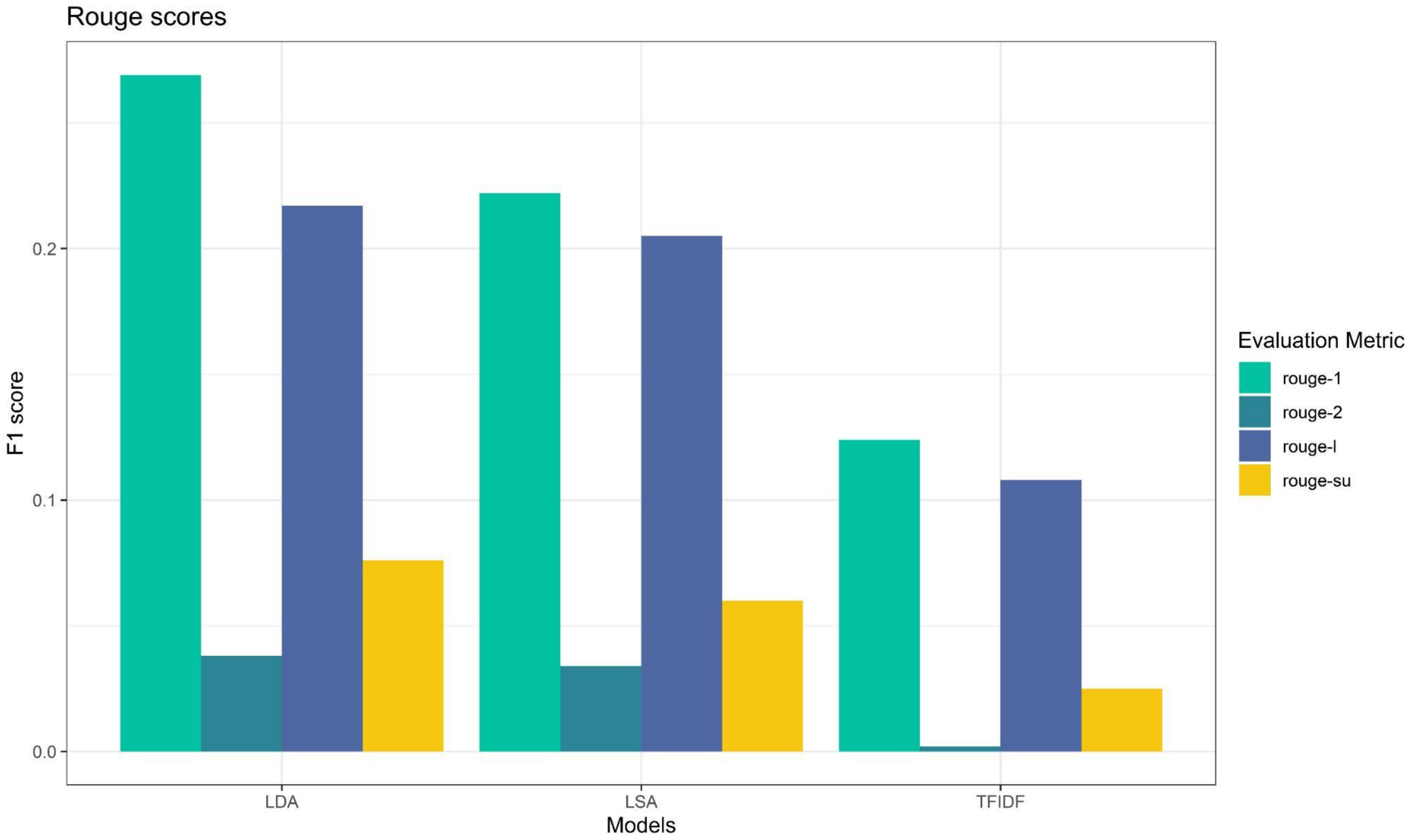
Rouge-L

Decides based on longest common sub-sequence

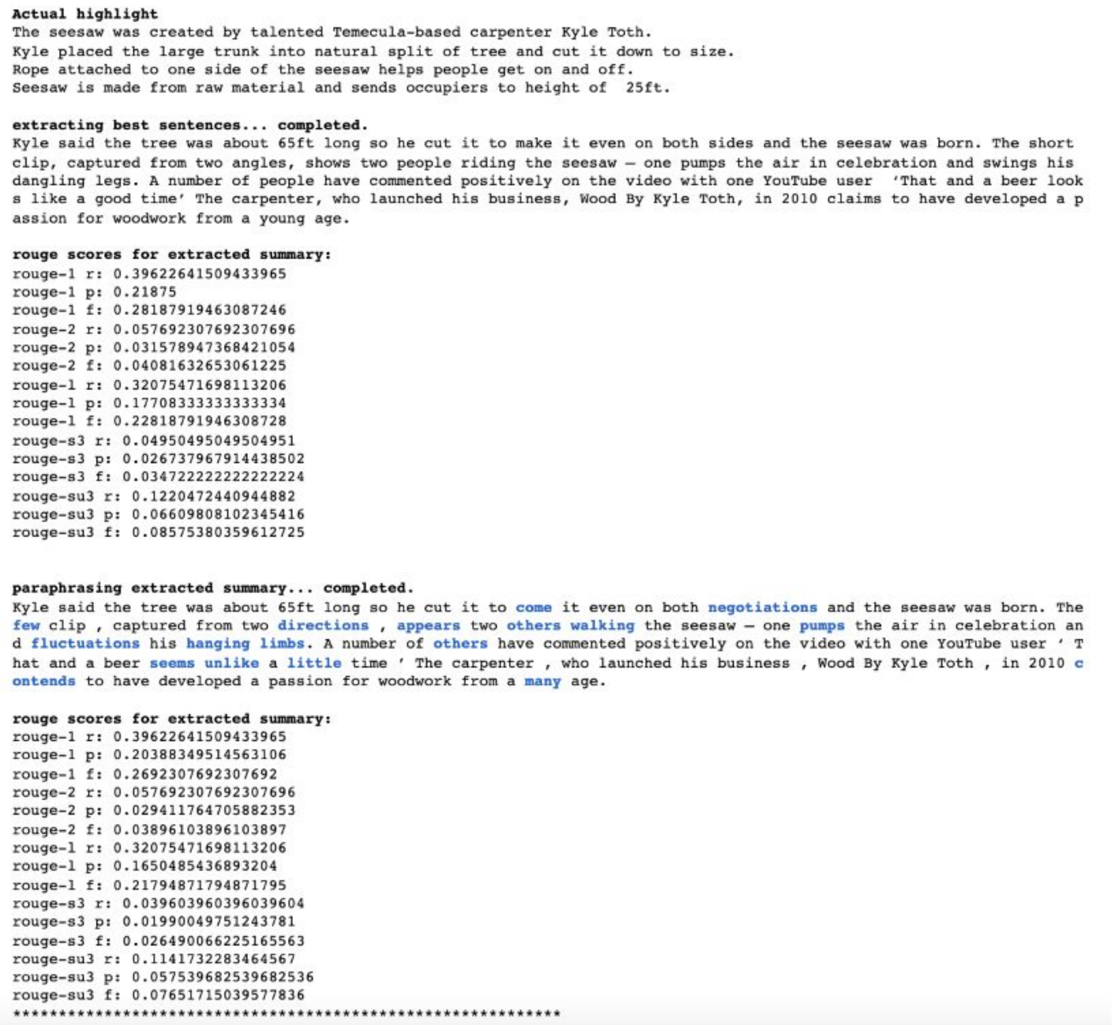
Rouge-SU

Uses the concept of skip-gram

ROUGE Scores of the model

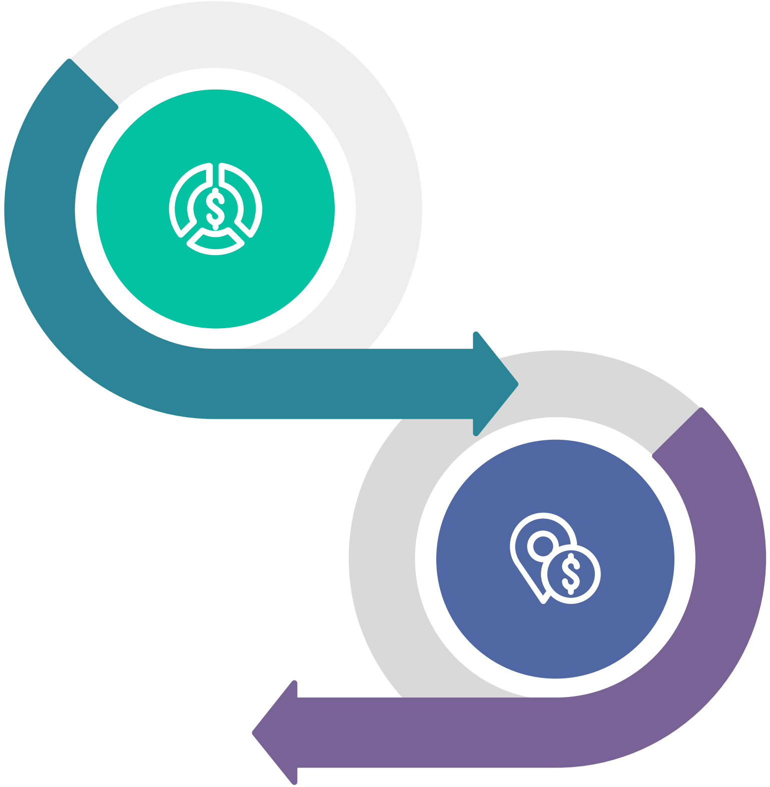


LDA Output



```
Actual highlight
The seesaw was created by talented Temecula-based carpenter Kyle Toth.
Kyle placed the large trunk into natural split of tree and cut it down to size.
Rope attached to one side of the seesaw helps people get on and off.
Seesaw is made from raw material and sends occupiers to height of  25ft.

extracting best sentences... completed.
Kyle said the tree was about 65ft long so he cut it to make it even on both sides and the seesaw was born. The short
clip, captured from two angles, shows two people riding the seesaw – one pumps the air in celebration and swings his
dangling legs. A number of people have commented positively on the video with one YouTube user  'That and a beer look
s like a good time' The carpenter, who launched his business, Wood By Kyle Toth, in 2010 claims to have developed a p
assion for woodwork from a young age.

rouge scores for extracted summary:
rouge-1 r: 0.39622641509433965
rouge-1 p: 0.21875
rouge-1 f: 0.28187919463087246
rouge-2 r: 0.057692307692307696
rouge-2 p: 0.031578947368421054
rouge-2 f: 0.04081632653061225
rouge-l r: 0.32075471698113206
rouge-l p: 0.17708333333333334
rouge-l f: 0.22818791946308728
rouge-s3 r: 0.04950495049504951
rouge-s3 p: 0.026737967914438502
rouge-s3 f: 0.034722222222222224
rouge-su3 r: 0.1220472440944882
rouge-su3 p: 0.06609808102345416
rouge-su3 f: 0.08575380359612725


paraphrasing extracted summary... completed.
Kyle said the tree was about 65ft long so he cut it to come it even on both negotiations and the seesaw was born. The
few clip , captured from two directions , appears two others walking the seesaw – one pumps the air in celebration an
d fluctuations his hanging limbs. A number of others have commented positively on the video with one YouTube user ' T
hat and a beer seems unlike a little time ' The carpenter , who launched his business , Wood By Kyle Toth , in 2010 c
ontends to have developed a passion for woodwork from a many age.

rouge scores for extracted summary:
rouge-1 r: 0.39622641509433965
rouge-1 p: 0.20388349514563106
rouge-1 f: 0.2692307692307692
rouge-2 r: 0.057692307692307696
rouge-2 p: 0.029411764705882353
rouge-2 f: 0.03896103896103897
rouge-l r: 0.32075471698113206
rouge-l p: 0.1650485436893204
rouge-l f: 0.21794871794871795
rouge-s3 r: 0.039603960396039604
rouge-s3 p: 0.01990049751243781
rouge-s3 f: 0.026490066225165563
rouge-su3 r: 0.1141732283464567
rouge-su3 p: 0.05753968253968254
rouge-su3 f: 0.07651715039577836
************************************************************
```

Drawbacks of the model



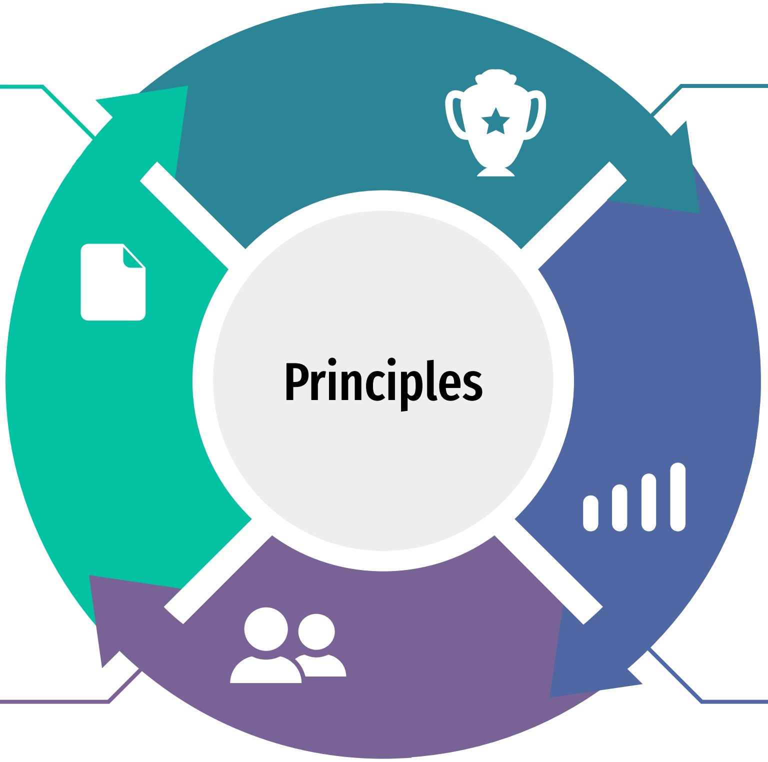
Time Complexity

The model takes a high computation time.

Evaluation Metric Value

The model gives out low scores when evaluated using ROUGE metrics.

Further Improvements



Optimization

More code optimization may result in lower time complexity

Computational linguistics

Add more concepts of computational linguistics in the paraphrase model.

Fine Tuning

Fine tuning of some parameters of the model may improve the model.

Transformers

Use transformer in the paraphrase for outstanding results

Thank you!