CNN articles: Topic Modeling and Text summarization

Text Mining and Search project

- *Riva Leonardo, 830647, l.riva37@campus.unimib.it*
- *Stoffa Giacomo, 830159, g.stoffa1@campus.unimib.it*

Abstract

This project aims to implement a topic modeling and text summarization pipeline to news articles by CNN. With the first task, the main topics in the news are defined and evaluated. After that, the text summarization follows three different techniques to generate a slimmer version of the articles.

Contents

1. Introduction **2**

2. Dataset **2**

3. Preprocessing **2**

4. Topic modeling **3**

4.1. Sklearn LDA 3

4.2. Gensim LDA 5

4.3. Evaluation 7

5. Text summarization **7**

5.1. Extractive summarization: indicator representation approach 8

5.2. Extractive summarization: topic representation approach 8

5.3. Abstractive summarization 9

5.4. Results 9

5.5. Evaluation 10

6. Conclusions **10**

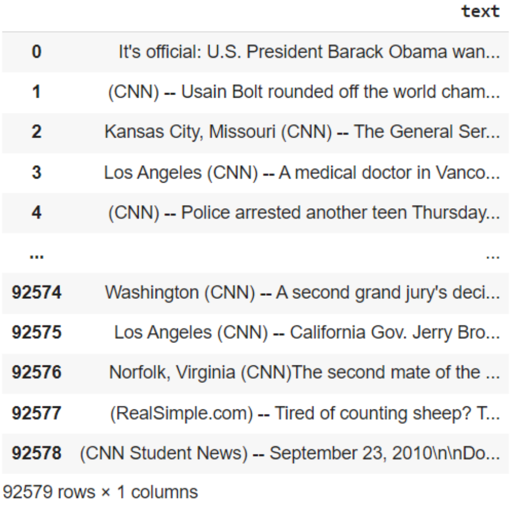
7. References **10**

1. Introduction

The tasks that are going to be carried out are two standard Natural Language Processing operations: Topic Modeling and Text summarization.

2. Dataset

The dataset, available on Github [1], is composed of 92 579 raw articles from CNN (a U.S. television broadcaster).



	text
0	It's official: U.S. President Barack Obama wan...
1	(CNN) -- Usain Bolt rounded off the world cham...
2	Kansas City, Missouri (CNN) -- The General Ser...
3	Los Angeles (CNN) -- A medical doctor in Vanco...
4	(CNN) -- Police arrested another teen Thursday...
...	...
92574	Washington (CNN) -- A second grand jury's deci...
92575	Los Angeles (CNN) -- California Gov. Jerry Bro...
92576	Norfolk, Virginia (CNN)The second mate of the ...
92577	(RealSimple.com) -- Tired of counting sheep? T...
92578	(CNN Student News) -- September 23, 2010\n\nDo...

92579 rows × 1 columns

Figure 1. The CNN dataset.

These texts have various lengths and don't have any additional features. There aren't any missing values nor duplicates.

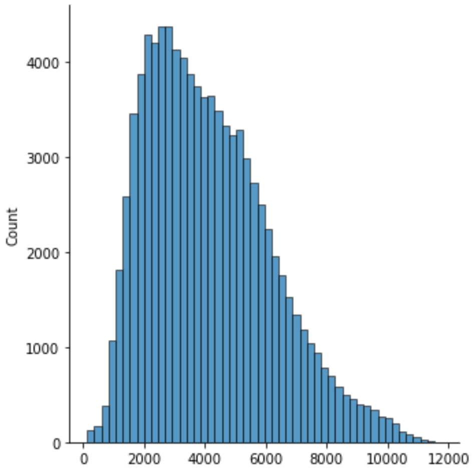


Figure 2. Distribution of articles' length.

3. Preprocessing

This process, applied to the raw texts, consists in various operations, to facilitate the next tasks.

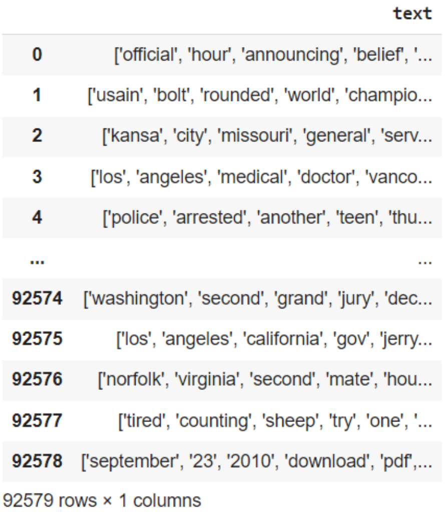
First, the removal of:

- specific portions of text (e.g. “@highlight”, source at the start of the articles, line breaks, etc.), which are essentially not useful
- emojis
- links, through a regex: _“(https?:\/\/)?([\da-z\.-]+)\.([a-z\.]{2,6})([\/\w \.-]*)”_
- stopwords
- punctuation
- whitespaces in excess

Then, language contractions were expanded, (e.g. “don’t” → “do not”) to manage words more easily.

After that, a tokenization of words (i.e. the text is split into single-word lists).

Finally, a lemmatization, using NLTK Wordnet Lemmatizer [2].



	text
0	['official', 'hour', 'announcing', 'belief', '...
1	['usain', 'bolt', 'rounded', 'world', 'champio...
2	['kansa', 'city', 'missouri', 'general', 'serv...
3	['los', 'angeles', 'medical', 'doctor', 'vanco...
4	['police', 'arrested', 'another', 'teen', 'thu...
...	...
92574	['washington', 'second', 'grand', 'jury', 'dec...
92575	['los', 'angeles', 'california', 'gov', 'jerry...
92576	['norfolk', 'virginia', 'second', 'mate', 'hou...
92577	['tired', 'counting', 'sheep', 'try', 'one', '...
92578	['september', '23', '2010', 'download', 'pdf',...

92579 rows × 1 columns

Figure 3. The preprocessed dataset.

4. Topic modeling

This is an unsupervised machine learning technique capable of scanning a set of documents, detecting word and phrase patterns within them, and automatically clustering word groups (topics) and similar expressions that best characterize a set of documents.

These topics will emerge during the topic modeling process based on the technique known as Latent Dirichlet Allocation (LDA) used to extract topics from a corpus. LDA’s approach to topic modeling is to consider each document as a collection of topics, and each topic as a collection of keywords.

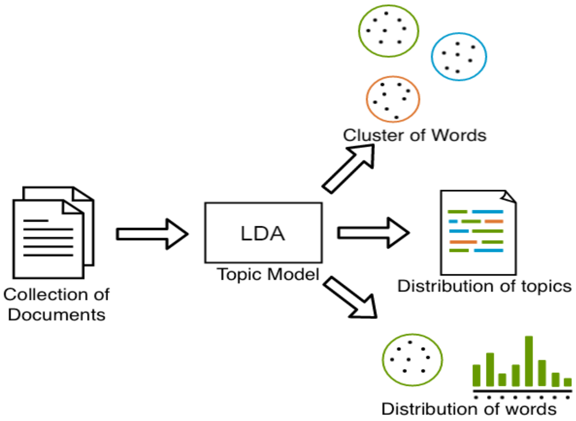
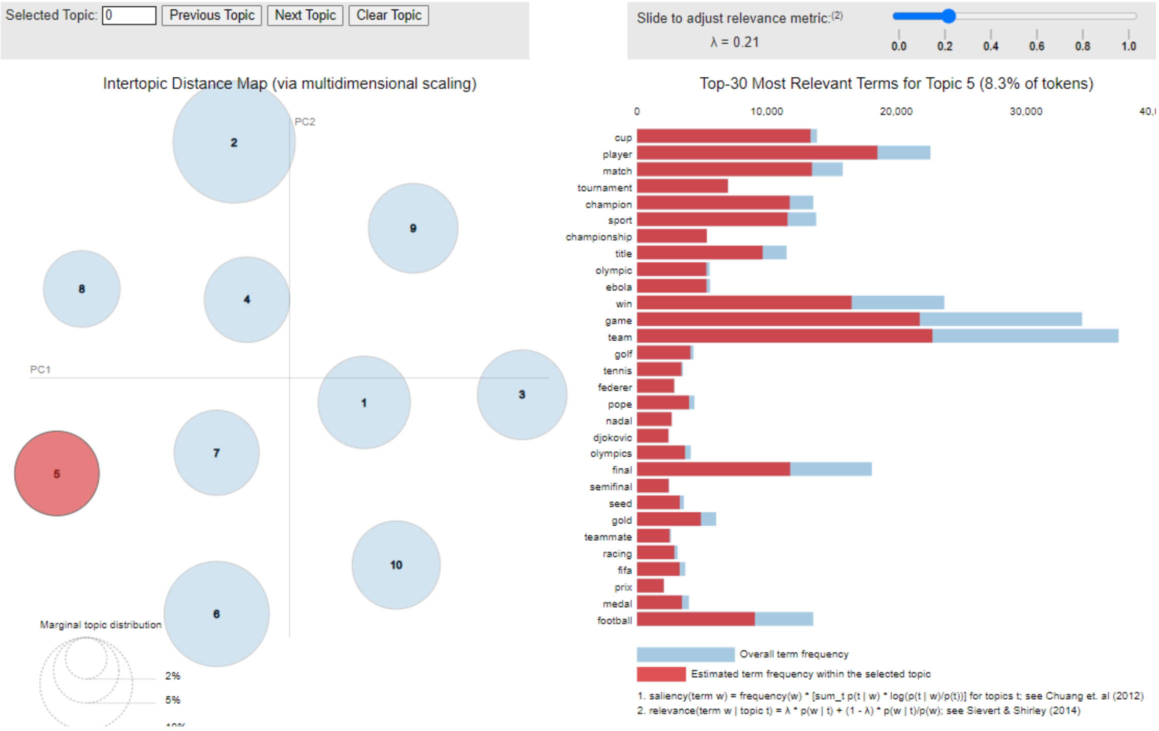


Figure 4. Workflow of LDA topic modeling.

For this purpose, it was decided to use two different approaches, the first using the sklearn library [3] and the second using gensim [4].

4.1. Sklearn LDA

After carrying out the correct preprocessing methodologies, it was decided, after several tests, to create the model with 10 clusters.



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)

Figure 5. Interactive visualization of the Sklearn LDA result.

Through the interactive view in *Figure 5* it is possible to see the subdivision of the clusters for the different topics, also adjusting the relevance metric λ to make the clusters more exclusive considering the possibility of associating a word to a single topic cluster.

To show the most associated word, another useful visualization is the word cloud, which can display the 30 most associated words with each grouping. The size of a word depends on the relevance with the topic. Some example:

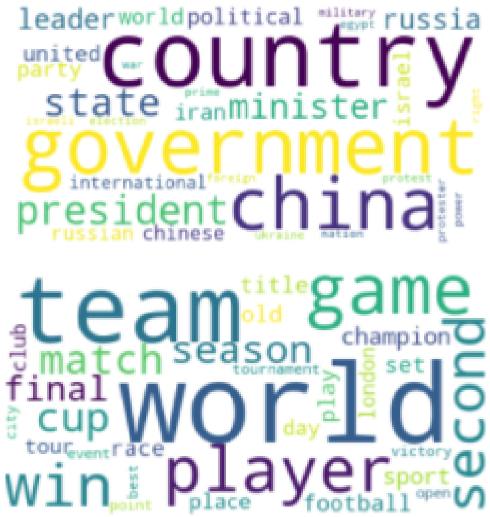
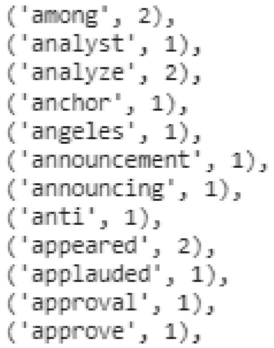


Figure 6. Examples of two word clouds, about topics government and sport respectively.

4.2. Gensim LDA

This module needs as input a dictionary of words, in addition to the corpus. This has been filtered out of tokens that appear in less than 10 documents, for computational purposes. By doing so, the number of words in the dictionary dropped down from 223687 to 49781 words.

With the two inputs, it creates a mapping of (word_id, word_frequency).



```
('among', 2),
('analyst', 1),
('analyze', 2),
('anchor', 1),
('angeles', 1),
('announcement', 1),
('announcing', 1),
('anti', 1),
('appeared', 2),
('applauded', 1),
('approval', 1),
('approve', 1),
```

Figure 7. Examples of the mappings created by Gensim LDA.

You need to provide the number of topics as well. The Gensim library provides more flexibility to evaluate the number of clusters: some measures have been tested [5] on the model with different numbers of clusters (6, 8, 10, 12, 14, 16). The two measures are:

- model perplexity is an intrinsic evaluation metric that captures how surprised a model is of new data it has not seen before, and is measured as the normalized log-likelihood of a held-out test set.
- topic coherence measures the score a single topic by measuring the degree of semantic similarity between high scoring words in the topic.

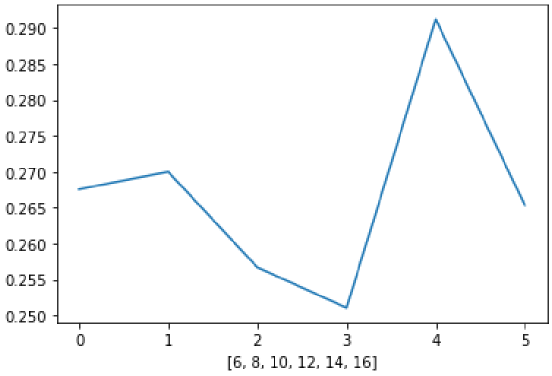
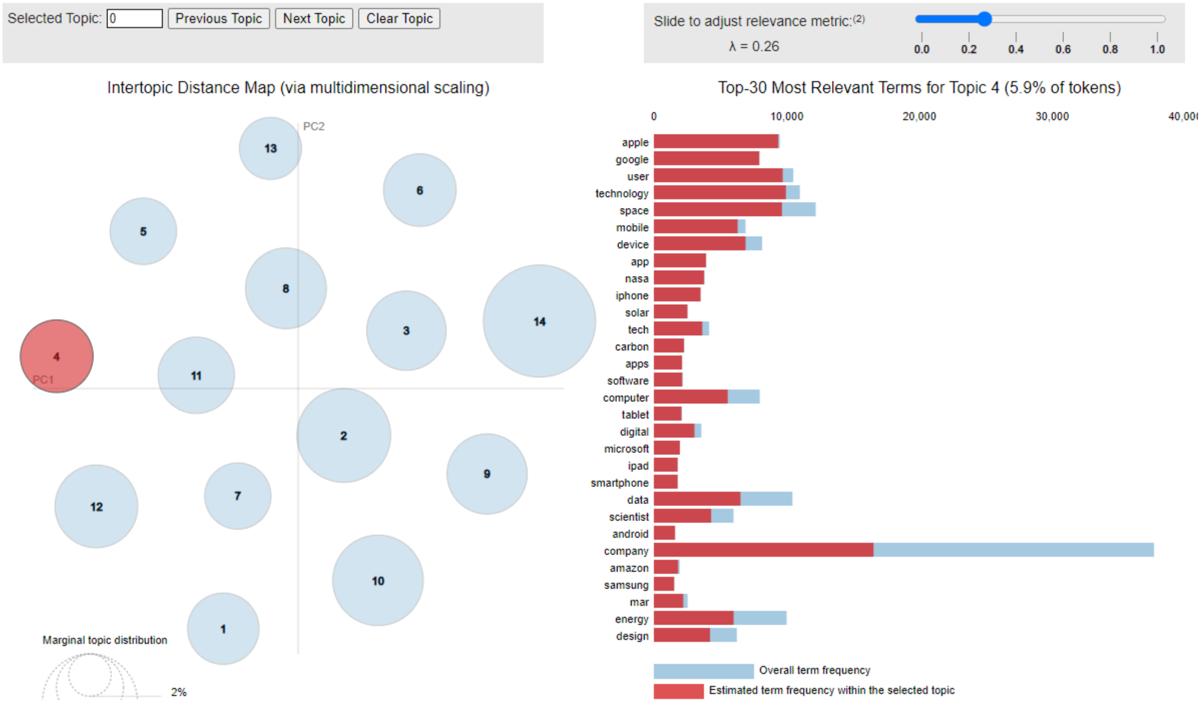


Figure 8. Results on the test. While perplexity is constant, coherence reaches its highest value with 14 topics.

Therefore, the LDA model is built with 14 different topics.

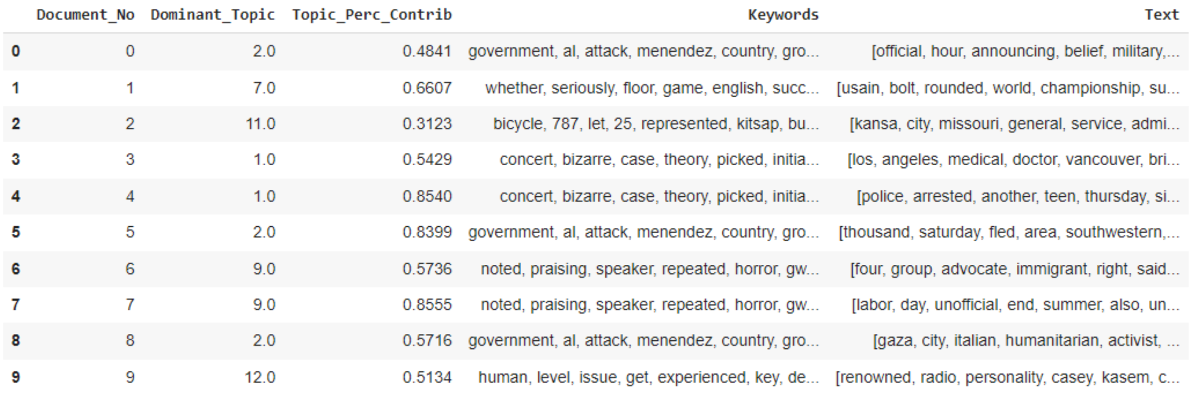


Selected Topic: 0 | Previous Topic | Next Topic | Clear Topic

Slide to adjust relevance metric:(2) λ = 0.26

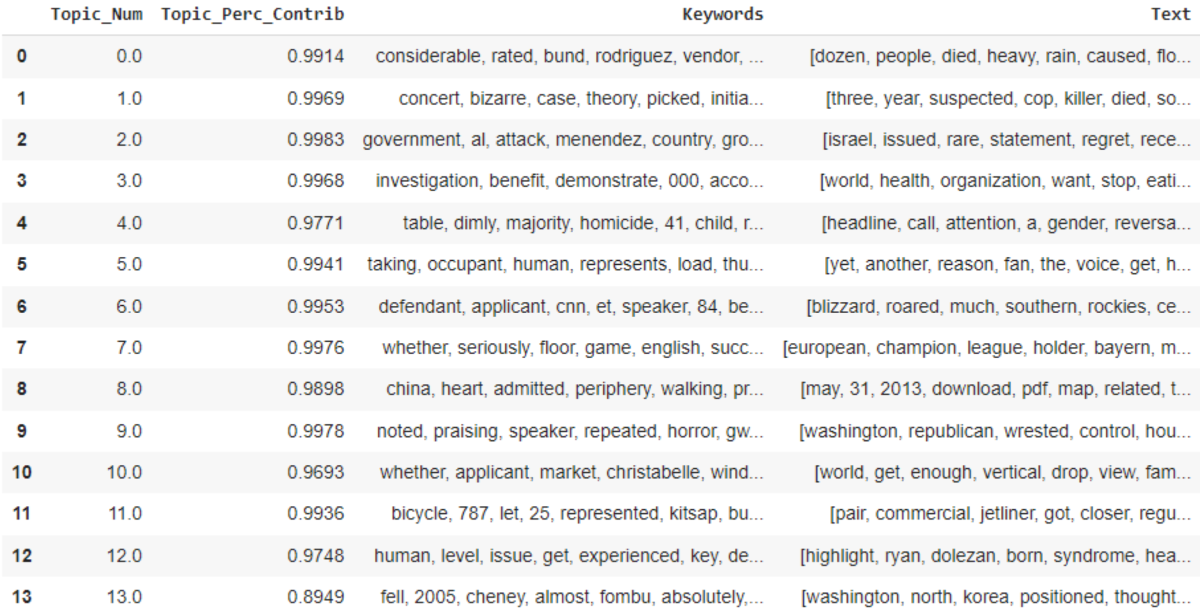
Figure 9. Interactive visualization of the Gensim LDA result. It was preferred to consider a threshold of λ = 0.25 in order to exclude too generic words.

The following visualizations aim to show different peculiarities of the generated topics.



	Document_No	Dominant_Topic	Topic_Perc_Contrib	Keywords	Text
0	0	2.0	0.4841	government, al, attack, menendez, country, gro...	[official, hour, announcing, belief, military,...
1	1	7.0	0.6607	whether, seriously, floor, game, english, succ...	[usain, bolt, rounded, world, championship, su...
2	2	11.0	0.3123	bicycle, 787, let, 25, represented, kitsap, bu...	[kansa, city, missouri, general, service, admi...
3	3	1.0	0.5429	concert, bizarre, case, theory, picked, initia...	[los, angeles, medical, doctor, vancouver, bri...
4	4	1.0	0.8540	concert, bizarre, case, theory, picked, initia...	[police, arrested, another, teen, thursday, si...
5	5	2.0	0.8399	government, al, attack, menendez, country, gro...	[thousand, saturday, fled, area, southwestern,...
6	6	9.0	0.5736	noted, praising, speaker, repeated, horror, gw...	[four, group, advocate, immigrant, right, said...
7	7	9.0	0.8555	noted, praising, speaker, repeated, horror, gw...	[labor, day, unofficial, end, summer, also, un...
8	8	2.0	0.5716	government, al, attack, menendez, country, gro...	[gaza, city, italian, humanitarian, activist, ...
9	9	12.0	0.5134	human, level, issue, get, experienced, key, de...	[renowned, radio, personality, casey, kasem, c...

Figure 10. Examples of the most dominant topics that better describe different articles, with a weight of contribution.



	Topic_Num	Topic_Perc_Contrib	Keywords	Text
0	0.0	0.9914	considerable, rated, bund, rodriguez, vendor, ...	[dozen, people, died, heavy, rain, caused, flo...
1	1.0	0.9969	concert, bizarre, case, theory, picked, initia...	[three, year, suspected, cop, killer, died, so...
2	2.0	0.9983	government, al, attack, menendez, country, gro...	[israel, issued, rare, statement, regret, rece...
3	3.0	0.9968	investigation, benefit, demonstrate, 000, acco...	[world, health, organization, want, stop, eati...
4	4.0	0.9771	table, dimly, majority, homicide, 41, child, r...	[headline, call, attention, a, gender, reversa...
5	5.0	0.9941	taking, occupant, human, represents, load, thu...	[yet, another, reason, fan, the, voice, get, h...
6	6.0	0.9953	defendant, applicant, cnn, et, speaker, 84, be...	[blizzard, roared, much, southern, rockies, ce...
7	7.0	0.9976	whether, seriously, floor, game, english, succ...	[european, champion, league, holder, bayern, m...
8	8.0	0.9898	china, heart, admitted, periphery, walking, pr...	[may, 31, 2013, download, pdf, map, related, t...
9	9.0	0.9978	noted, praising, speaker, repeated, horror, gw...	[washington, republican, wrested, control, hou...
10	10.0	0.9693	whether, applicant, market, christabelle, wind...	[world, get, enough, vertical, drop, view, fam...
11	11.0	0.9936	bicycle, 787, let, 25, represented, kitsap, bu...	[pair, commercial, jetliner, got, closer, regu...
12	12.0	0.9748	human, level, issue, get, experienced, key, de...	[highlight, ryan, dolezan, born, syndrome, hea...
13	13.0	0.8949	fell, 2005, cheney, almost, fombu, absolutely,...	[washington, north, korea, positioned, thought...

Figure 11. Topics prototypes: the most weighted documents for each topic.

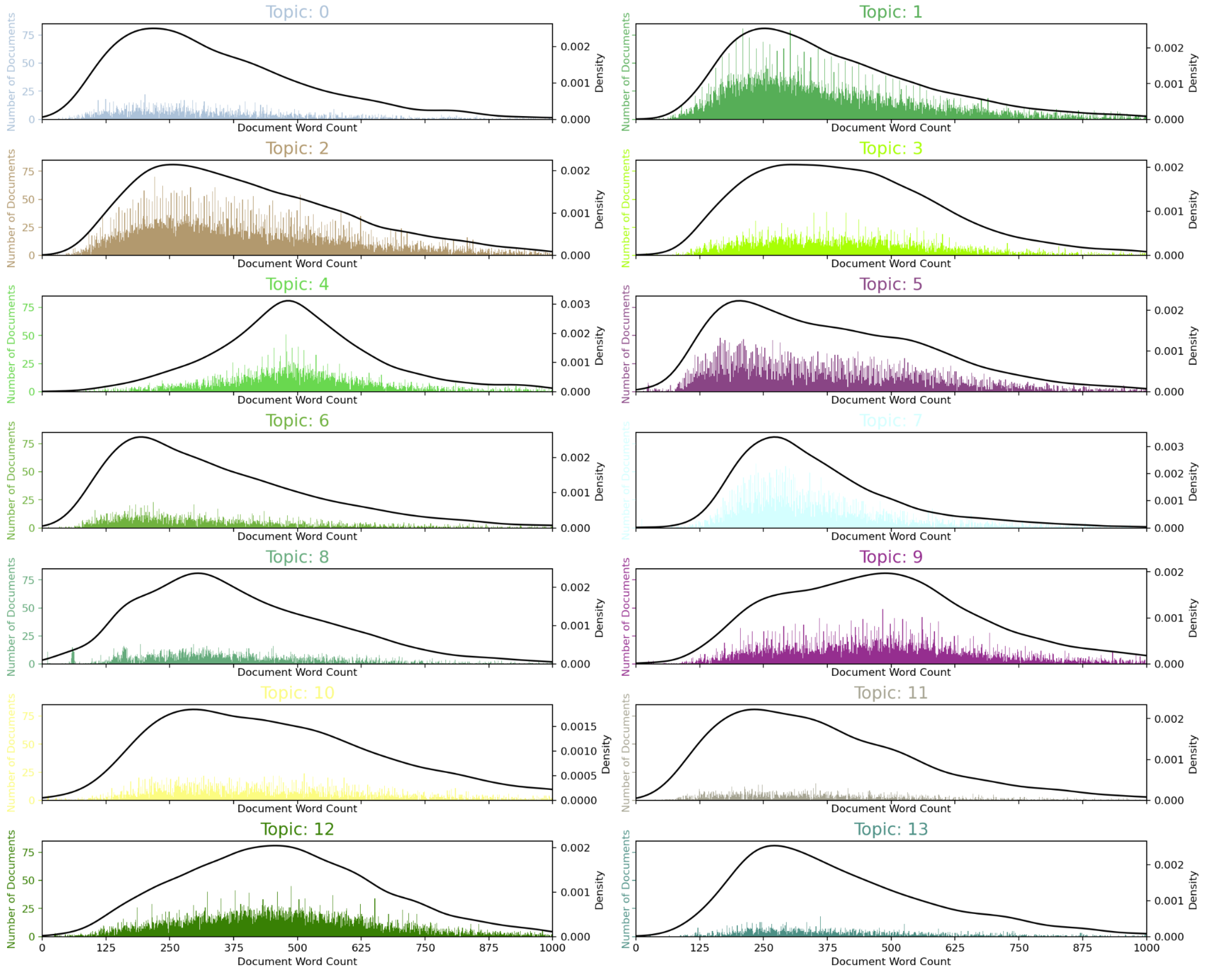
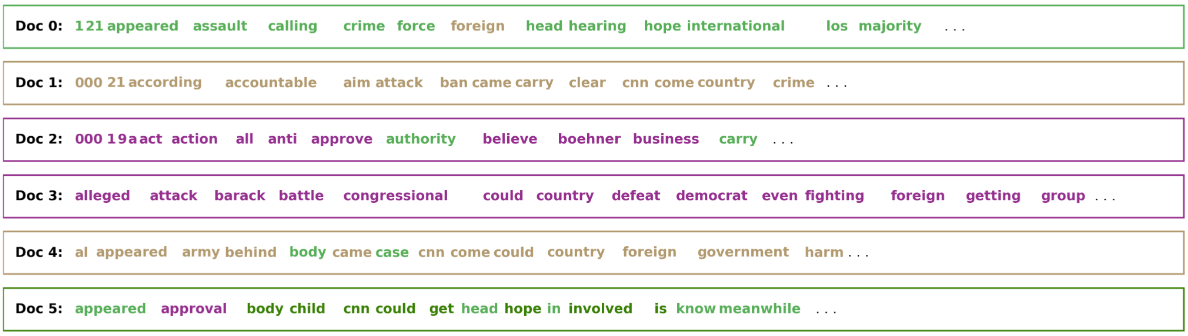


Figure 12. Distribution of word counts by topic.



Doc 0: 121 appeared assault calling crime force foreign head hearing hope international los majority . . .

Doc 1: 000 21 according accountable aim attack ban came carry clear cnn come country crime . . .

Doc 2: 000 19 a act action all anti approve authority believe boehner business carry . . .

Doc 3: alleged attack barack battle congressional could country defeat democrat even fighting foreign getting group . . .

Doc 4: al appeared army behind body came case cnn come could country foreign government harm . . .

Doc 5: appeared approval body child cnn could get head hope in involved is know meanwhile . . .

Figure 13. Examples of different membership to topics by each word of a document. Each topic has a different color.

4.3. Evaluation

The model created using the first approach was done by empirically choosing 10 clusters, after different tests, while the Gensim library was easier to manage and we were able to determine the best model by calculating the coherence and perplexity indices.

Also, the creation of the model through sklearn was much slower (>10m vs 2m): this is another reason to prefer a different approach.

5. Text summarization

The goal of this task is, given a random article, summarize it in a few sentences. We also need the dataset with a lighter preprocessing (without stopwords removal, punctuation removal, tokenization and lemmatization), because, in the end, the original sentences are necessary to form a summary.

Three different techniques have been tested and compared. The first two fall under the category of extractive summarization: they extract the most meaningful sentences from the document. The last one is an abstractive summarization, meaning it creates a completely new text, based on the document.

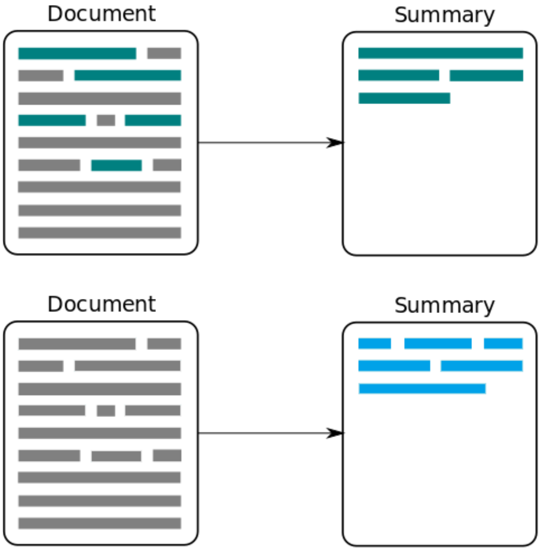


Image X. Visual comparison of extractive and abstractive summarization.

5.1. Extractive summarization: indicator representation approach

It creates a graph using TF-IDF vectors to identify the best sentences.

Steps:

1. **Text normalization, tokenization and lemmatization**. These operations are done while preserving the original sentences.

2. **TF-IDF vectorization** of words, which are calculated based on the tokens.

3. **Graph creation**. Designed after a similarity matrix of TF-IDF vectors.

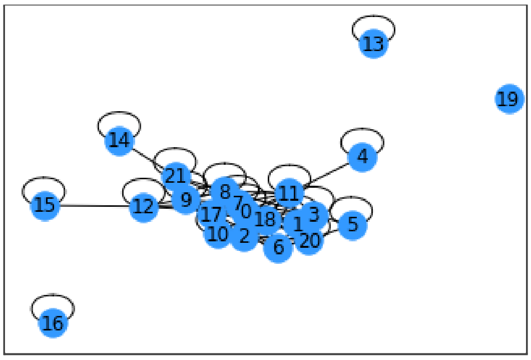


Image 2: Example of a graph.

4. **Rank sentences**. This step is performed by the PageRank algorithm [6]. The sentences are then sorted by their ranks.

5. **Filter the best sentences**. It returns the top N sentences (in this case, N=5 looks like they can contain enough information, for the articles' length).

5.2. Extractive summarization: topic representation approach

In this case, word frequencies are calculated to detect the best sentences.

Steps:

1. **Text normalization, tokenization and lemmatization**. As before, these operations are done while preserving the original sentences.

2. **Count words in the document**. This absolute count is then converted to relative values, between 0 and 1.

3. **Calculate sentences' scores**. The score of a sentence is defined as the sum of its words relative count. Their score is again converted to relative values. Furthermore, long sentences are ignored (> 30 words), because a summary is, by definition, short.

4. **Filter with threshold.** To choose the final sentences, instead of selecting the top N ones, we can filter with a threshold, which is arbitrarily set as 0.6, in order to show a few sentences.

5.3. Abstractive summarization

A final test has been performed using transformers. An already existing pipeline for summarization has been used to create summaries with new words and sentences. The model used is the default for this pipeline: "distilbart-cnn-12-6" [7].

Of course, using a pre-trained model, this technique is less customizable; the only "personalization" is on the length of the final summary, set between 15 and 100 words.

5.4. Results

The following paragraphs are examples of summarization performed by the three techniques shown before. Each summary is divided into sentences.

Method 1

- *"Kenyan President Uhuru Kenyatta on Wednesday became the first sitting head of state to appear before the International Criminal Court, where he faces charges of crimes against humanity"*
- *"Kenya is the second African nation after Sudan to have a sitting president face charges at the International Criminal Court"*
- *"In ordinary circumstances, the insufficiency of evidence would cause the prosecution to withdraw the charges, the ICC said in September"*
- *"The ICC has said the trials will proceed"*
- *"Kenyatta is the first sitting President to appear before the ICC"*

Method 2

- *"During the hearing Tuesday, the prosecution accused the Kenyan government of not providing key documents in the case against its leader"*
- *"In ordinary circumstances, the insufficiency of evidence would cause the prosecution to withdraw the charges, the ICC said in September"*
- *"Both leaders have denied any links to the violence among their respective ethnic groups, and have said they will cooperate with the court to clear their names"*

Method 3

- *"Uhuru Kenyatta is the first sitting head of state to appear before the International Criminal Court. He is accused of five counts of crimes against humanity for allegedly orchestrating violence after a disputed presidential election in 2007. More than 1,000 people died and hundreds of thousands were displaced when ethnic groups loyal to leading candidates torched homes and hacked rivals in violence that raged until early 2008. The second day of the status hearing will determine whether his case can proceed to trial."*

5.5. Evaluation

There is not a systematic way to define a performance measure, without a test set with a groundtruth. Therefore, some considerations are made after a hand-made analysis.

The method 1 works discreetly, while following a simple methodology.

The method 2 is too naive. Using a threshold, there is no control over the number of sentences, which can be both a good and a bad thing; at the same time, manually selecting a threshold can be restrictive.

The method 3 creates smart summaries, but sometimes sentences don't make much sense; also, it doesn't work with too long documents.

6. Conclusions

We initially started by analyzing the dataset specifically. Then, we understood what topic modeling and text summarization can really achieve. We built a basic topic model using Sklearn/Gensim's LDA and visualized the topics using pyLDAvis. We have created three different summarization models and then define the best one.
As regards to the topic modeling part, a possible improvement could be to make a comparison between LSA and LDA to determine the best model.

7. References

[1] GitHub - abisee/cnn-dailymail: Code to obtain the CNN / Daily Mail dataset (non-anonymized) for summarization (2018). https://github.com/abisee/cnn-dailymail.

[2] NLTK :: nltk.stem.wordnet (2022). https://www.nltk.org/_modules/nltk/stem/wordnet.html.

[3] sklearn.discriminant_analysis.LinearDiscriminantAnalysis — scikit-learn 0.24.1 documentation (2021). Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.discriminant_analysis.LinearDiscriminantAnalysis.html

[4] gensim: topic modelling for humans. (2021). https://radimrehurek.com/gensim/models/ldamodel.html

[5] Shashank Kapadia. (2019). Evaluate Topic Models: Latent Dirichlet Allocation (LDA). Medium; Towards Data Science. https://towardsdatascience.com/evaluate-topic-model-in-python-latent-dirichlet-allocation-lda-7d57484bb5d0

[6] Pagerank - NetworkX 1.7 documentation (2022). https://networkx.org/documentation/networkx-1.7/reference/generated/networkx.algorithms.link_analysis.pagerank_alg.pagerank.html.

[7] sshleifer/distilbart-cnn-12-6 · Hugging Face (2022). https://huggingface.co/sshleifer/distilbart-cnn-12-6.