ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ

ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS

SCHOOL OF BUSINESS

DEPARTMENT OF MANAGEMENT SCIENCE & TECHNOLOGY

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

ACADEMIC YEAR OF 2021 – 2022

MACHINE LEARNING & CONTENT ANALYTICS

KONSTANTINOS NINAS - f2822108

STAMATIOS SIDERIS – f2822113

YIANNIS VANIOTIS – f2822101

ORESTIS LOUKOPOULOS – f2822104

SUPERVISING INSTRUCTOR - CHARIS PAPAGEORGIOU

Table of Contents

1 - Introduction

Nowadays, given the explosive amounts of data, more and more applications emerge that are associated with the inference and the summarization of articles. In specific, the huge amounts of data have led to the creation of machine learning models that can utilize text sequences and extract useful information. Most readers usually choose their articles based on their titles, without knowing in advance the contents of the article. As a result, they can easily be misled. For this reason, many machine learning models, based on Neural Networks, have been developed to tackle this issue. These algorithms have been trained to summarize the most important points made by the author in a few sentences, while at the same time retaining the semantic scope in which it was written. In addition, recommendation systems have gotten very popular in the last decade, from platforms such as Spotify, Netflix and more. As expected, such technologies have spread to other industries, including the news and scientific industry because newsreaders are keen on having a more personalized feed of news articles that will align more with their interests.

1.1 - Our Project

The main task of the project is to implement a machine learning algorithm in the domain of neural networks that will aim to demonstrate a solution for a business application problem. The project that will be presented revolves around two different segments. The first part is the summarization of texts from different article sources, with the fundamental idea being that the critical information available in a newspaper or an online article can be summarized in a fairly small text. To produce the aforementioned summaries, we will utilize the 'Facebook Bart large' pretrained text summarization model with further training and tuning and a new text summarization model developed from our team.

The second part of the project aims to take advantage of the constructed summaries of the articles, produced from the best of the two models, in order to create clusters. The constructed clusters will contain articles of similar topics. An additional feature that will be a vital part of our project is to use these clusters to develop a basic recommendation system that will identify the news articles most similar to the one the user is reading and recommend them an article with a related topic.

1.2 – Our Vision/Goals

Regarding our goals that stem from the first segment of our project, we aim to minimize the readers' amount of time spent reading a news article and to extract the crucial information that may or may not interest them. This has the added benefit that the users will be able to maximize the number of news stories that they will be able to read and increase the useful information that they can receive.

On the other hand, regarding the implementation of the second segment of our project, it will enable an organization to group together relevant articles together. This application will, not only help the users to easily find articles similar to the ones they are interested in but will also create a more user-friendly environment that will increase the organization's engagement with its customers. Furthermore, this implementation will increase the time spent by users in a website and maximize the potential earnings of the business unit.

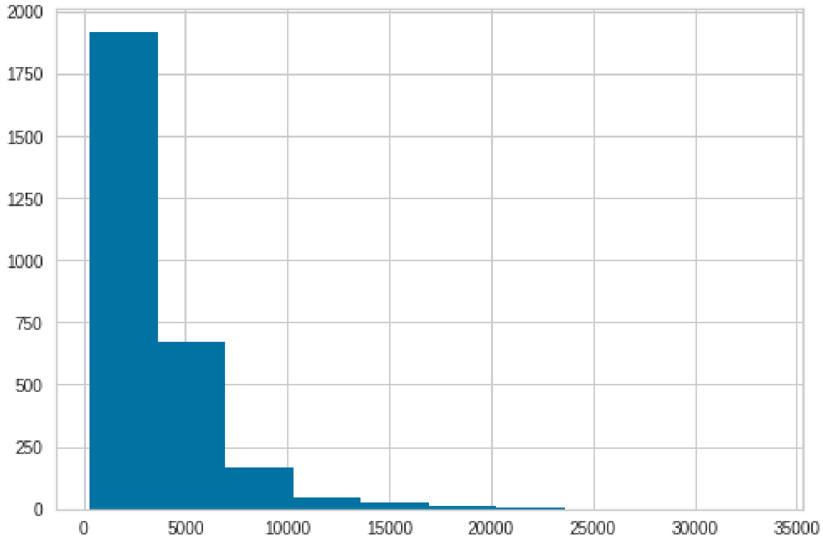
2 – Methodology

As stated above, the main purpose of the project is to construct an article summarization model and a basic recommendation engine. These tools aim to improve the readers' experience on various news websites and provide them with a news feed that will be customized according to their interests. To accomplice those goals, it was firstly deemed necessary to identify a dataset that would include both the news articles and their validated human-written summaries. Once the dataset was found, we had to clean and explore the data in order to develop a greater understanding of its structure and contents. It was deemed beneficial to create 2 different models that would produce articles' summaries. For the two models two radically different approaches were implemented. The first approach included the utilization of a pretrained model and its corresponding tools, while the second approach included the creation of a articles' summarization model from scratch. As expected, the model with the best performance was the pretrained one. The summaries produced from that model, were later used to create clusters of articles that supposedly have similar contexts. Finally, these clusters were utilized to produce a news basic recommendation system for the end users.

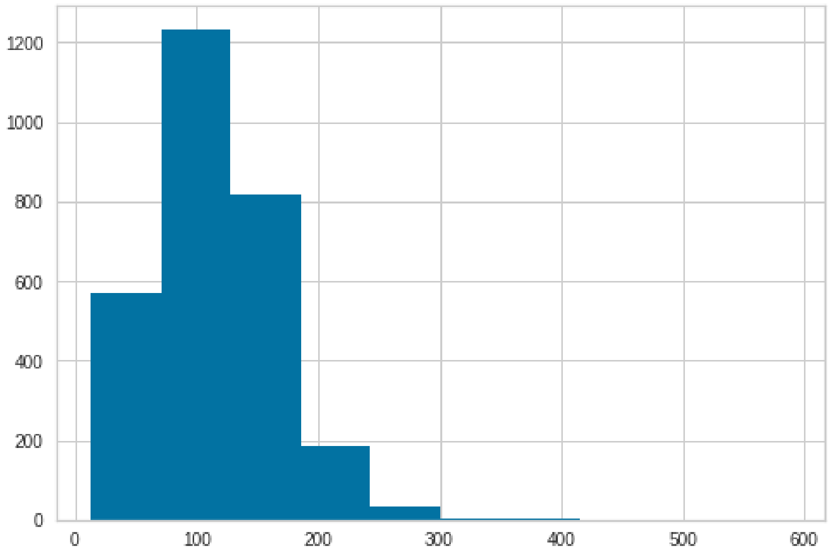
2.1 – Data Collection & Overview

To identify the proper dataset a number of data sources were explored. Our needs forced us to search for clean well understood and topic specified text, alongside with its validated summary that could be used as the base of a proper summarization model. The dataset that seemed to cover our needs was the BigPatent Dataset [1]. This dataset has summaries that contain richer discourse structure with more recurring entities, evenly distributed content and shorter extractive fragments present in the summaries.

This dataset consists of texts that describe patents and their uses in both full-text and summarized format. Datasets such as this, can provide summary-worthy content especially for the purposes of our project given the ability to highly compress the available information of the articles. Upon exploring the data, it was discovered that the median length of the descriptions based on the words included was almost equal to 2600, while that of the summaries was almost equal to 100. Additionally, the 1st and the 3rd quantiles of the length of the descriptions are approximately equal to 1800 and 4100 words respectively (Plot 1). The 1st and the 3rd quantiles of the length of the summaries are approximately 75 and 140 respectively (Plot 2).



Plot 1. Length of Descriptions



Plot 2. Length of Summaries

2.2 - Data Processing/Annotation/Normalization

Initially, the data were split to multiple directories. Each directory contained multiple subdirectories, which contained the zipped articles in a json format. It was necessary to convert the data to a human readable format (such as .csv). To accomplish this, the programming language python and some pre-installed libraries such as ‘json’ and ‘pandas’ were used, to read iteratively all the json files through all directories and subdirectories and to save them in a csv file. This csv file includes 3 columns, the first one that has the publication number of the article, the second one that includes the full text of the article and the third column that contains the article’s summary. Following that, the data were checked for missing values and duplicates based on their publication number, with no such values being found in the data. Additionally, the text from the dataset was transformed to lowercase.

Now with the dataset cleaned, the next step, was to split it in three different parts. But prior to the splitting, the data were shuffled. The first part is used to train our model and will have approximately 80% of the total number of observations in the dataset. The second part was used to test the results of the different configurations of the models that were created, using 10% of the remaining observations. The third, and final, part was used to evaluate the performance of the best model and contained the last 10% of all the observations.

2.3 - Algorithms, NLP architectures/systems

As previously discussed, we developed two distinct text summarization models. The first model was a pretrained model ('Facebook Bart Large') [2], while the second one was developed and trained from our team from scratch. The reasoning behind the construction of the two different models was to compare their performance and keep the one that is more efficient in the production of article summaries.

Firstly, we will further discuss the pretrained model that was further trained and fine tuned in our needs. The Facebook Bart model was trained in the English language (the same language as our texts). According to Facebook and Hugging Face "Bart is a transformer encoder-decoder (seq2seq) model with a bidirectional (Bert-like) encoder and an autoregressive (GPT-like) decoder. Bart is pre-trained by corrupting text with an arbitrary noising function and learning a model to reconstruct the original text. Bart is particularly effective when fine-tuned for text generation (e.g., summarization, translation) but also works well for comprehension tasks (e.g., text classification, question answering)" [2]. The first part of Bart uses the bi-directional encoder of Bert to find the best representation of its input sequence. For every text sequence in its input, the Bert encoder outputs an embedding vector for each token in the sequence as well as an additional vector containing sentence-level information. In this way, the decoder can learn for both token and sentence-level tasks making it a robust starting point for any future fine-tuning tasks. In addition to the Bart Model, we also used the Bart Tokenizer. The tokenizer is used to convert all the words in each article into a sequence of numbers.

The model's structure starts with 2 token embedding layers and a token embedding positional layer, followed by multiple encoder layers. The encoder layers include an attention layer, followed by a normalization layer, an activation function and multiple linear layers. Each attention layer includes multiple linear layers. After the encoder layer, the decoder layer can be identified. The decoder layer starts with a token embedding and a token embedding positional layer. Then, there are multiple decoder sublayers. Each decoder sublayer starts with an attention layer, which also includes multiple linear layers, followed by an activation function. Then, a normalization layer follows, and then another attention layer. Following, there is another normalization layer and multiple linear layers. The output layers are a normalization layer followed by a linear layer.

A transformer model is a neural network that learns context and thus meaning by tracking relationships in sequential data like the words in this sentence [3]. The role of the token embeddings layer is to transform words into vector representations of fixed dimension. In the case of the Bart model, each word is represented as a 1024-dimensional vector. The Position Embeddings layer is a lookup table of size (1026, 1024) where the first row is the vector representation of any word in the first position, the second row is the vector representation of any word in the second position, etc. As transformers do not encode the sequential nature of their inputs, this layer helps our model to understand the difference between 2 similar words, existing in different positions in a sentence, and assign them with different vectors. The encoder layer looks at the entire sequence and learns high-dimensional representations with bi-directional information. The decoder layer takes these thought vectors and regressively predicts the next token creating the final output iteratively. The attention mechanism can help a neural network to memorize long sequences of the data by dynamically highlighting relevant features of the input data. The normalization layer is used to normalize the vector representations of the input data in order to improve stability and sometimes quality [4]. In other words, an activation function decides whether a neuron should be activated or not. This means that it will decide whether the neuron's input to the network is important or not in the process of prediction using simpler mathematical operations [5]. In our case, the GELU activation function is used. The linear layers are used to transform input features to output features by using matrix multiplication with a weight matrix.

After the description of the pretrained model, we will present the architecture of the model that we constructed from the ground up below. Our model is an encoder-decoder Seq2Seq (a special class of Recurrent neural networks) model. The encoder layer includes a token embedding layer, and multiple LSTM layers. The decoder layer starts with a token embedding layer, followed by an LSTM layer. The decoder layer is followed by an attention and a denser layers. The dense layer is creating the output summary of the articles. Recurrent neural networks recognize the input data's sequential characteristics and use patterns to predict the next output. Recurrent Neural Networks suffer from short-term memory. If a sequence is long enough, they'll have a hard time carrying information from earlier time steps to later ones. The LSTM (Long Short-Term Memory) layer is used because it has the capability to learn order dependencies in sequence predictions. For our model, we created a new tokenizer, and fitted it on our data, in order to work with a much smaller dictionary of tokens.

Once both models well trained and evaluated, we identified that the pretrained Bart model was more efficient in the production of the articles' summaries. As such, all the articles were summarized again using the best model. These summaries were transformed into vectors using the Bert sentence embedding pretrained model. Following, the vectors were used in order to conduct K-Means clustering to the articles. Specifically, we conducted the clustering for various number of clusters and evaluated them based on a set of metrics (such as silhouette score). Upon identifying the most fitting number of clusters for our data, we built a basic recommendation system based on those clusters. That system receives an article/text sequence as input, then proceeds to predict the cluster it would assign it to, given the existing clusters, and recommends the most similar articles that belong in that group of articles.

3 - Experiments – Setup, Configuration

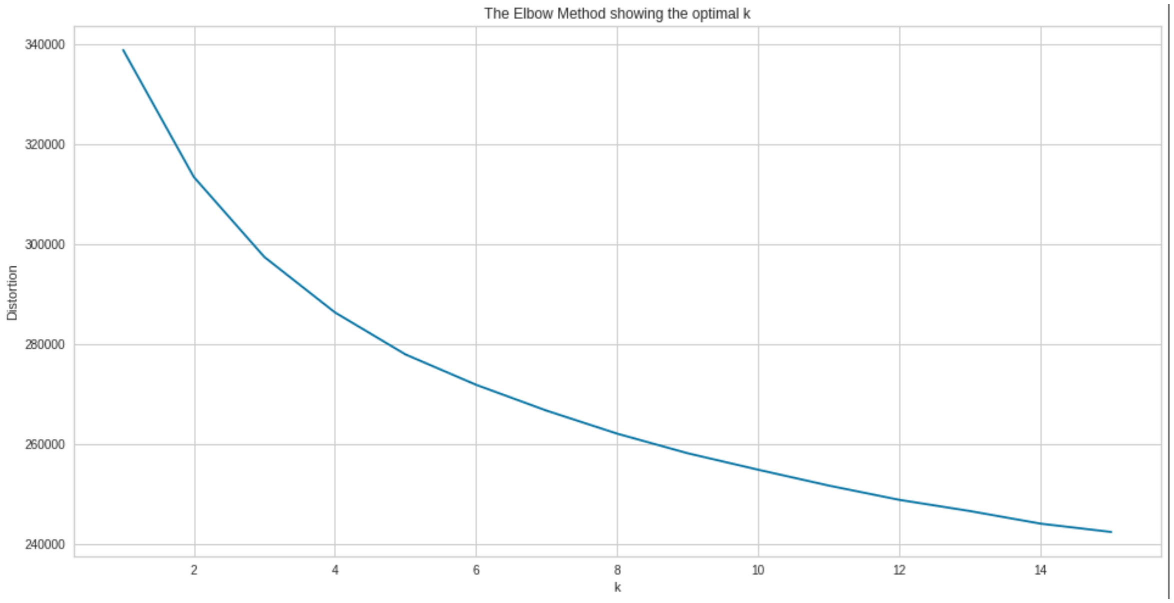
Before we started to apply the text inputs in our two models a setup processes needed to be implemented to bring the available data texts in a suitable format. In both models, all input sequences should have the same length. To tackle that issue, the tokenizer was used to pad and truncate the sequences to ensure that all sequences of numbers have the same length. Padding is the process of lengthening a vector of size n to size k (where k > n), by adding a finite number of zeros, in order for that vector to be the same length as the rest of the vectors available. While on the other hand, truncation is the process of shortening a vector of size n to size z (where z < n), by removing elements of the vector, so that vector has the same length as the rest of the vectors. In our case we implemented both of those processes at the end of the vectors, given that the Bart tokenizer has a limit of 1024 tokens (words) per text and that the articles have lengths of various sizes.

In order to produce more desirable results, we had to fine-tune the arguments and the configurations for both the pretrained and our custom model. In specific, regarding the pretrained model, a number of parameters where enabled (e.g., evaluate generated text, evaluate during training verbose). We also set a train batch size equal to 5, a number of epochs equal to 10 and a maximum sequence length equal to 1024 (equal to that of the tokenizer) (Plot 3). In addition, we set a minimum and maximum length for the produced summaries equal to 70 and 150 respectively.

We set the minimum and maximum lengths of the summaries according to the length of the 1st and 3rd quantiles of the existing human-written summaries. We also set a length penalty equal to 2, to discourage the model from producing big summaries and enabled early stopping, so that the training may stop early if the model's performance deteriorates during the training process. We used n-grams of size 3, that would be used at most 1 time in the predicted summary. Finally, we used a number of beams equal to 4, which means that in every step that the decoder predicts the next token, it creates 4 branches of token sequences and finally keeps the best based on some metric.

On the other hand, the configurations of our custom article summarization model were much simpler. To be exact, we used embeddings of size equal to 200 in order to try to produce better results. The number of training epochs was equal to 20, while the batch size that would be received as input in the model in each instance was equal to 5. Finally, we set 20% of the training data to be used for validation from the model.

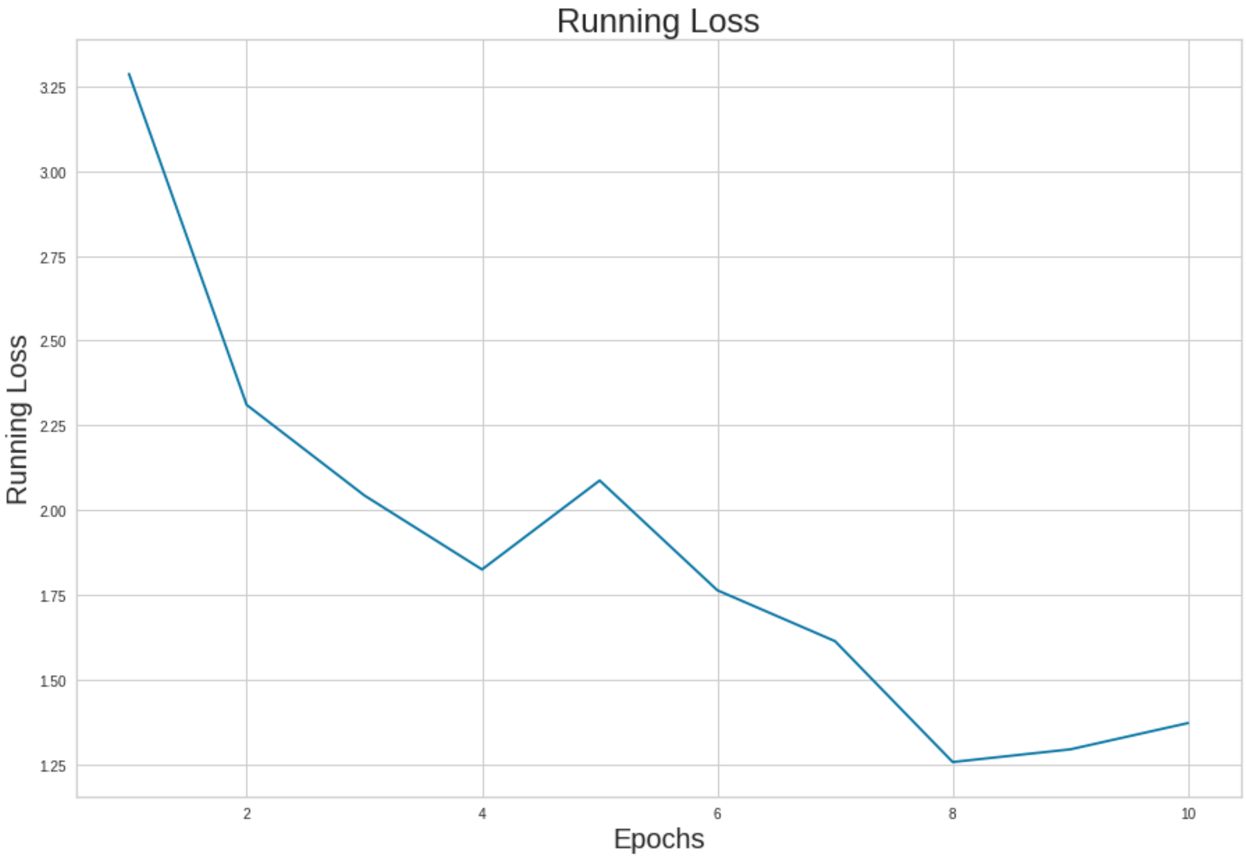
Regarding the clustering that followed, we used the elbow method to determine the most suitable number of clusters, given our data (plot 4). It was decided that the final number of clusters that we needed to produce was equal to 6.



Plot 4. Elbow Method on Clustering

3.1 - Results & Quantitative Analysis

To evaluate and compare our two models we used the running loss and the rouge score of the two models during their training and evaluation. The pretrained model had better performance according to both measures. The running loss occurs from a function that quantifies the difference between the expected outcome (validated summary) and the produced summary from each model. The running loss that occurred from the training of the Bart model can be observed in plot 5. As it can be seen from the plot, in each epoch the running loss tends to drop.



Plot 5. Running Loss of Pretrained Model

The rouge score is a measure that compares the number of words that are included in the predicted outcome that are also present in the validated summary, while considering the total number of words that exist in the expected outcome. It should be mentioned that the rouge score can be divided to 3 subscores, Rouge-1, Rouge-2 and Rouge-L. Rouge-1 finds the overlap of each word (unigram) between the provided summary and the generated one, while the Rouge-2 metric finds the overlap of sets of two words (bigrams) between the provided summary and the generated one.

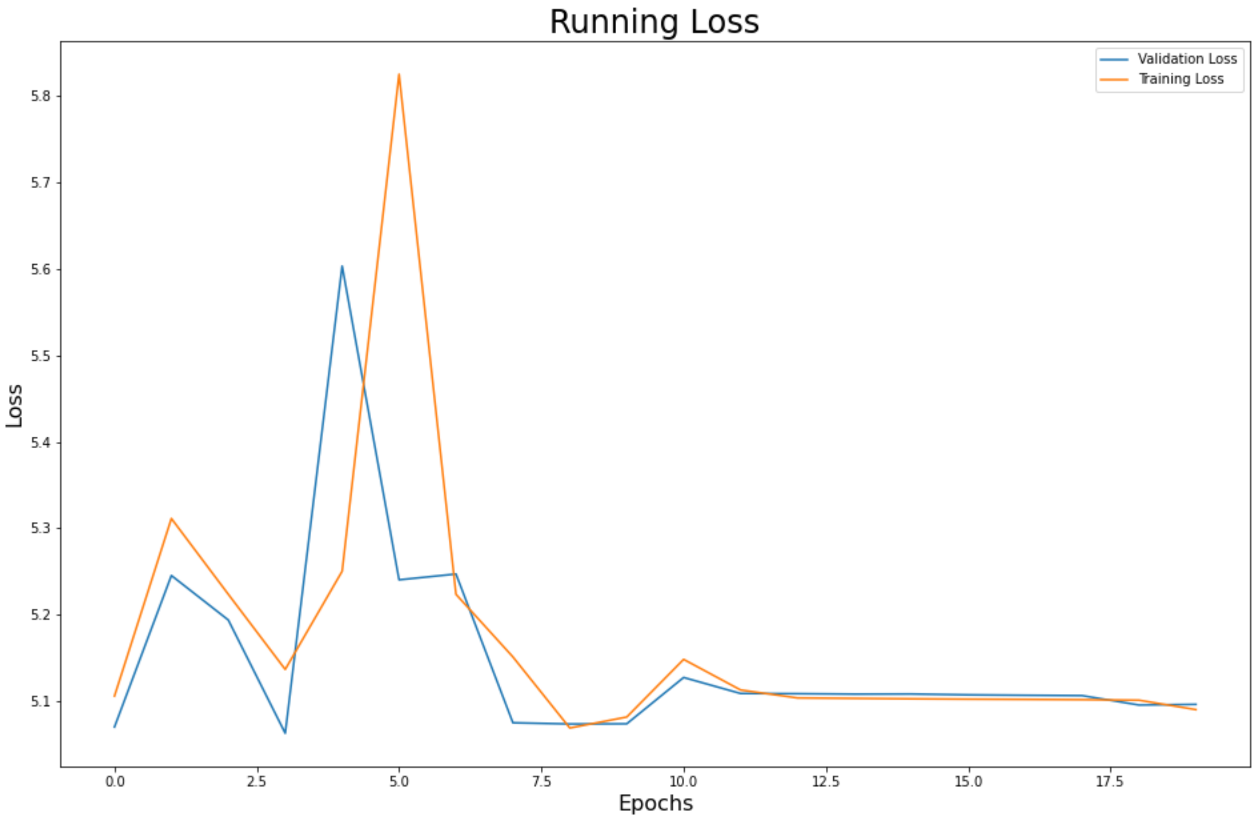
Lastly, the Rouge-L metric measures the longest common subsequence (LCS) between the predicted output of the model and the reference. In other words, when the predicted output and the target share longer word sequences, it leads to greater similarity between the two, and thus to a greater Rouge-L value. Each one of the rouge scores, has a recall, a precision and a f1-score. The recall measure estimates the number of n-grams that exist in both the generated and the validated summary, and then divides that number by the number of the n-grams that exist in the validated summary. The precision measure exacts the same calculation, but instead of dividing the number of common n-grams by the number of n-grams in the validated summary, it divides it by the number of those in the generated summary. The f1-score is a weighted average of the recall and the precision. The results of the rouge scores that were calculated on the predictions of the test data (with data that were unseen to the model up to that point) can be observed at table 1.

	Rouge-1	Rouge-2	Rouge-L
Recall	0.293	0.093	0.256
Precision	0.326	0.094	0.284
F1-Score	0.299	0.089	0.260

Table 1. Rouge Scores on All Data

According to the results of table 1, we can interpret that in average, almost 3 out of 10 words exist in both the predicted and the validated summaries. Additionally, in average 1 out of 10 bigrams exist in both the predicted and the validated summaries. Finally, the least common subsequence found in both summaries is equal to almost ¼ of both text sequences.

As stated above, the performance of our custom article summarization model was much worse than that of the pretrained model in terms of both measures considered. To be specific, the running loss that was calculated from the training (Plot 6), for both the training and the valuation data was much greater than that of the pretrained model. Additionally, all rouge measures were significantly lower than their corresponding ones of the pretrained model.

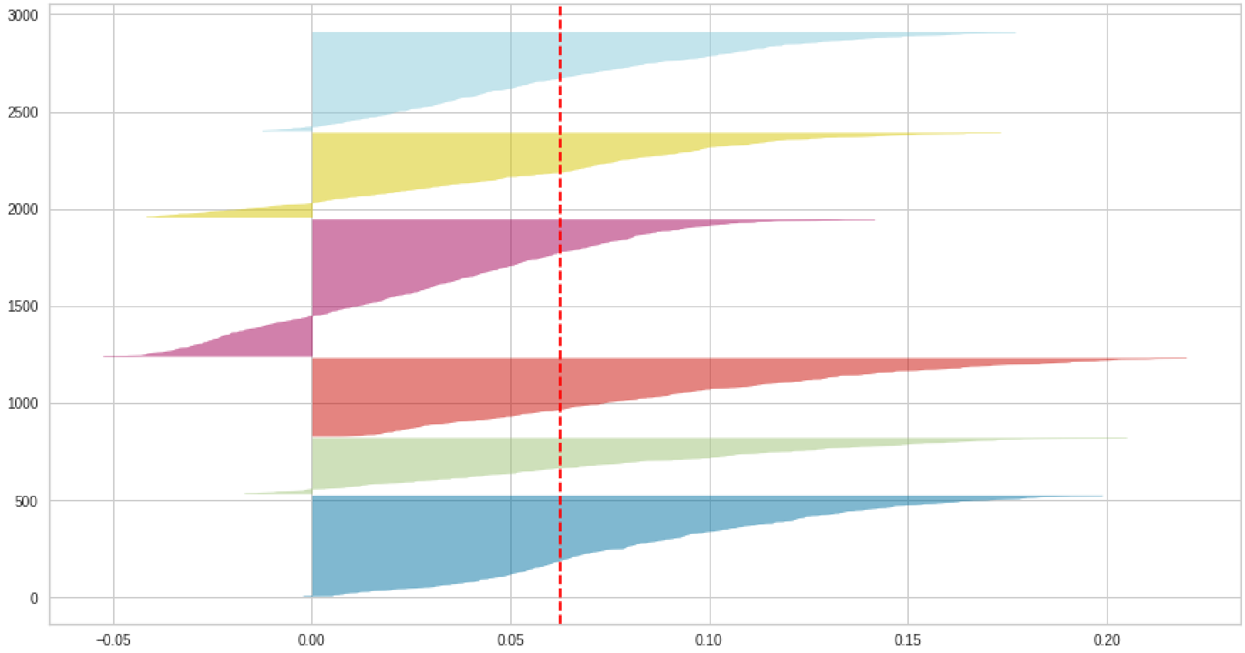


Plot 6. Running Loss of Custom Model

	Rouge-1	Rouge-2	Rouge-L
Recall	0.00017	0	0.00017
Precision	0.00158	0	0.00158
F1-Score	0.00030	0	0.00030

Table 2. Rouge Scores on All Data

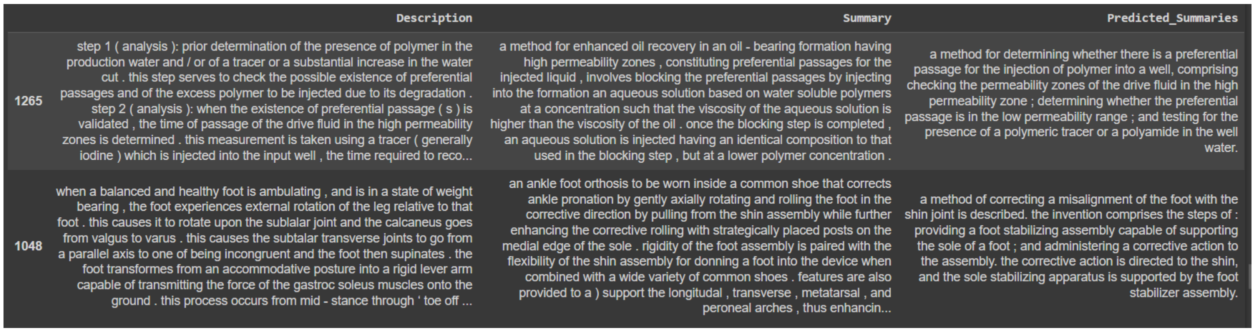
The clustering was evaluated by measuring the silhouette score. The silhouette score quantifies how good is the clustering at hand. Its values range from -1 to +1. To be specific, if an article is closer to another cluster of articles than the one it is currently grouped in, it receives a negative value. The average silhouette score presents the overall performance of the clustering technique implemented. In our case, the average score is equal to 0.06 (Plot 7), which can be translated to the clusters overlapping each other. This phenomenon is attributed to the fact that our initial data are not large in number and the majority of them refer to completely different topics. We strongly believe that once we collect more data, more clearly defined clusters can be formed.



Plot 7. Silhouette Plot

3.2 - Qualitative & Error Analysis

To try to understand where our model didn't succeed in producing adequate summaries, we proceeded to analyze a random sample of the created summaries, as well as the summaries with the lowest rouge scores. During our analysis in the random sample, we identified that for some articles, the length of the predicted and that of the actual summaries was significantly mismatched. Two example of such a case can be identified for the articles depicted in table 3.

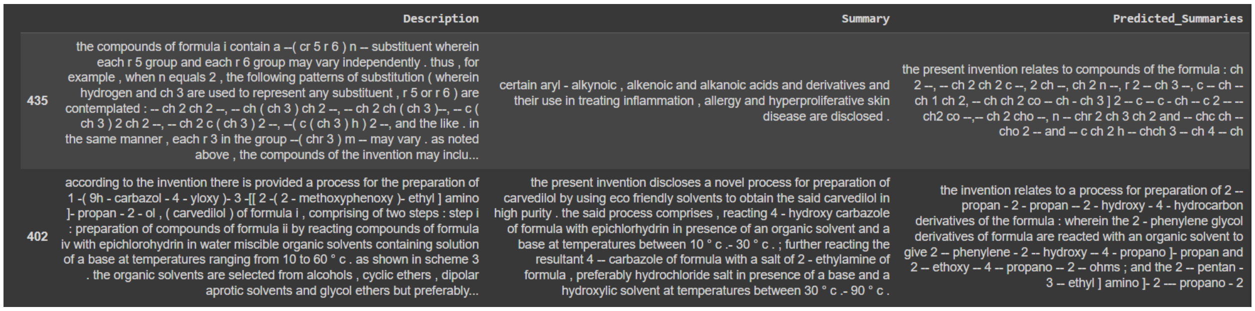


	Description	Summary	Predicted_Summaries
1265	step 1 (analysis): prior determination of the presence of polymer in the production water and / or of a tracer or a substantial increase in the water cut . this step serves to check the possible existence of preferential passages and of the excess polymer to be injected due to its degradation . step 2 (analysis): when the existence of preferential passage (s) is validated , the time of passage of the drive fluid in the high permeability zones is determined . this measurement is taken using a tracer (generally iodine) which is injected into the input well , the time required to reco...	a method for enhanced oil recovery in an oil - bearing formation having high permeability zones , constituting preferential passages for the injected liquid , involves blocking the preferential passages by injecting into the formation an aqueous solution based on water soluble polymers at a concentration such that the viscosity of the aqueous solution is higher than the viscosity of the oil . once the blocking step is completed , an aqueous solution is injected having an identical composition to that used in the blocking step , but at a lower polymer concentration .	a method for determining whether there is a preferential passage for the injection of polymer into a well, comprising checking the permeability zones of the drive fluid in the high permeability zone ; determining whether the preferential passage is in the low permeability range ; and testing for the presence of a polymeric tracer or a polyamide in the well water.
1048	when a balanced and healthy foot is ambulating , and is in a state of weight bearing , the foot experiences external rotation of the leg relative to that foot . this causes it to rotate upon the sublalar joint and the calcaneus goes from valgus to varus . this causes the subtalar transverse joints to go from a parallel axis to one of being incongruent and the foot then supinates . the foot transformes from an accommodative posture into a rigid lever arm capable of transmitting the force of the gastroc soleus muscles onto the ground . this process occurs from mid - stance through ' toe off ...	an ankle foot orthosis to be worn inside a common shoe that corrects ankle pronation by gently axially rotating and rolling the foot in the corrective direction by pulling from the shin assembly while further enhancing the corrective rolling with strategically placed posts on the medial edge of the sole . rigidity of the foot assembly is paired with the flexibility of the shin assembly for donning a foot into the device when combined with a wide variety of common shoes . features are also provided to a) support the longitudal , transverse , metatarsal , and peroneal arches , thus enhancin...	a method of correcting a misalignment of the foot with the shin joint is described. the invention comprises the steps of : providing a foot stabilizing assembly capable of supporting the sole of a foot ; and administering a corrective action to the assembly. the corrective action is directed to the shin, and the sole stabilizing apparatus is supported by the foot stabilizer assembly.

Table 3. Error Analysis 1 – Summaries with different length

We have deduced that this occurs due to the fact that we have a set a length penalty for the predicted summaries that the model produces to discourage it from creating big text sequences. We also have set an upper bound for the summaries to be created from the model, which also attributes to the fact that they are much smaller than their corresponding human-written ones. This may result to lower rouge scores since the original summaries include many more words than the predicted ones, and thus it is less possible that they have many common n-grams (or words).

Another issue that occurs and deteriorates the rouge scores of the predictions comes from the lack of clear contextualization and topic delimitation in the articles' main body (Table 4). That may confuse the model on where it should pay more attention and lead it to the production of summaries that don't correspond with the validated ones. As it can be seen from the examples presented below, the model seems to overly focus on the chemical sequences (e.g., ch-2) and not on the main concept of the article.



	Description	Summary	Predicted_Summaries
435	the compounds of formula i contain a --(cr 5 r 6) n -- substituent wherein each r 5 group and each r 6 group may vary independently . thus , for example , when n equals 2 , the following patterns of substitution (wherein hydrogen and ch 3 are used to represent any substituent , r 5 or r 6) are contemplated : -- ch 2 ch 2 --, -- ch (ch 3) ch 2 --, -- ch 2 ch (ch 3)--, -- c (ch 3) 2 ch 2 --, -- ch 2 c (ch 3) 2 --, --(c (ch 3) h) 2 --, and the like . in the same manner , each r 3 in the group --(chr 3) m -- may vary . as noted above , the compounds of the invention may inclu...	certain aryl - alkynoic , alkenoic and alkanoic acids and derivatives and their use in treating inflammation , allergy and hyperproliferative skin disease are disclosed .	the present invention relates to compounds of the formula : ch 2 --, -- ch 2 ch 2 c --, 2 ch --, ch 2 n --, r 2 -- ch 3 --, c -- ch -- ch 1 ch 2, -- ch ch 2 co -- ch - ch 3] 2 -- c -- c - ch -- c 2 -- -- ch2 co --,-- ch 2 cho --, n -- chr 2 ch 3 ch 2 and -- chc ch -- cho 2 -- and -- c ch 2 h -- chch 3 -- ch 4 -- ch
402	according to the invention there is provided a process for the preparation of 1 -(9h - carbazol - 4 - yloxy)- 3 -[[2 -(2 - methoxyphenoxy)- ethyl] amino]- propan - 2 - ol , (carvedilol) of formula i , comprising of two steps : step i : preparation of compounds of formula ii by reacting compounds of formula iv with epichlorohydrin in water miscible organic solvents containing solution of a base at temperatures ranging from 10 to 60 ° c . as shown in scheme 3 . the organic solvents are selected from alcohols , cyclic ethers , dipolar aprotic solvents and glycol ethers but preferably...	the present invention discloses a novel process for preparation of carvedilol by using eco friendly solvents to obtain the said carvedilol in high purity . the said process comprises , reacting 4 - hydroxy carbazole of formula with epichlorhydrin in presence of an organic solvent and a base at temperatures between 10 ° c .- 30 ° c . ; further reacting the resultant 4 -- carbazole of formula with a salt of 2 - ethylamine of formula , preferably hydrochloride salt in presence of a base and a hydroxylic solvent at temperatures between 30 ° c .- 90 ° c .	the invention relates to a process for preparation of 2 -- propan - 2 - propan -- 2 - hydroxy - 4 - hydrocarbon derivatives of the formula : wherein the 2 - phenylene glycol derivatives of formula are reacted with an organic solvent to give 2 -- phenylene - 2 -- hydroxy -- 4 - propano]- propan and 2 -- ethoxy -- 4 -- propano -- 2 -- ohms ; and the 2 -- pentan - 3 -- ethyl] amino]- 2 --- propano - 2

Table 4. Error Analysis 2 – Lack of Clear Contextualization

It is worth mentioning that it is generally not common for abstractive text summarization machine learning algorithms to overlap with human-written summaries, even though they may remain semantically accurate. This phenomenon occurs due to the fact that, since the produced summary is abstractive, it may use many synonyms to express the same words. As a result, rouge measure usually drops in those cases, since most n-grams between the two text sequences differ, even though they refer to the same topic.

3.3 - Discussion, Comments/Notes and Future Work

The scope of our project was to develop an application that should be able to produce accurate and consistent summaries of articles. At the same time, the application should be able to understand the context of a given article, to categorize it to a group that includes articles with similar topics and to recommend the most relevant ones. As discussed earlier the optimal model that was chosen was the pretrained one that has been fine tuned based on the available data. The pretrained model has a more complex architecture and it had been trained by a remarkably larger amount of data in contrast to our custom model, which attributed significantly to the difference in their performance.

There is an ongoing discussion that the use of pretrained neural network and topic specified models with fine tuning and transfer learning leads to much better results than developing a custom model from scratch. In our case, this assumption has been verified.

Regarding to future improvements, we would consider creating a more complex custom model architecture and collect more data to train it for our task or to experiment with different pretrained models (such as Bert, GPT-2, Google Pegasus and more).

Regarding to future work, we consider testing the final model's performance in different types of articles (for example health, economy, politics). Furthermore, as the model can only encode text sequences, we aim to make it possible for the model to handle most data structures.

Our specific business application based on our project will be an add-on that can be applied on any web page that includes articles of any sector. For this application, a database hosted in a web-server will be necessary that should include all the important features of the articles at our disposal. Such features will be the URL of the web page the article is hosted at, the article's main body, the article's summary (produced from our deployed model), the article's title, a relevant image (if it exists) and the cluster the article belongs to. In this way, it is easier for the website to present to the user the article's image, title and summary in an html format in a more user-friendly and understandable manner. Once the user clicks on an article, our recommendation engine will identify the 3 most similar articles to the one the user is currently reading and present him the articles' image, title and summary in an html block below the main body of the article making them available for reading. The clustering procedure should be repeated in short time periods for the clusters to be updated according to the freshly added articles in the database.

4 - Members/Roles

Our team consists of 4 members that are full-time students of MSc in Business Analytics, hosted by Athens University of Economics and Business for the academic period of 2021-2022. All members of the team contributed equally to all segments of the current project. Following, we will briefly present our team members.

The 1st team member is Orestis Loukopoulos. Orestis received his Bachelor’s degree in Mathematics from the University of Patras. Upon completing his undergraduate studies, he attended the MSc in Business Analytics as a full-time student. He currently works as a data scientist at a Greek bank.

The 2nd team member is Yiannis Vaniotis. Yiannis received his Bachelor’s degree in Accounting & Finance from Athens University of Economics and Business. Upon completing his undergraduate studies, he attended the MSc in Business Analytics as a full-time student. He currently works as a data management consultant at a well renown multinational company.

The 3rd team member is Stamatis Sideris. Stamatis received his Bachelor’s degree in Business Administration from Athens University of Economics and Business. Upon completing his undergraduate studies, he worked for a couple of years in the financial sector. Following, he attended the MSc in Business Analytics as a full-time student. He currently works as a data/machine learning engineer at a Greek company that is active in the energy sector.

The 4th and final member is Konstantinos Ninas. Konstantinos received his Bachelor’s degree in International and European Economic Studies from Athens University of Economics and Business. Upon completing his undergraduate studies, he worked for almost a year in the consulting sector. Following, he attended the MSc in Business Analytics as a full-time student. He currently works as a data/machine learning engineer at a multinational fintech company.

5 – Time Plan

Process	From	To
Identification of Project Idea	Middle of June	End of June
Dataset Search & Collection	27th of June	30th of June
Data Preprocessing	1st of July	6th of July
Construction of the Models	16th of July	10th of August
Evaluation of the best Model	11th of August	14th August
Clustering & Recommendation System	16th of August	20th of August
Report	21st of August	25th of August
Mockup Application	23rd of August	27th of August

6 - Bibliography

1. Metatext, accessed August 2022, https://metatext.io/datasets/bigpatent
2. Hugging Face, August 2022, https://huggingface.co/facebook/bart-large
3. NVIDIA, August 2022, https://blogs.nvidia.com/blog/2022/03/25/what-is-a-transformer-model/
4. Lena Voita, July 2022., https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html
5. Pragati Baheti, July 2022, https://www.v7labs.com/blog/neural-networks-activation-functions