Big Text Data Analytics in selected domain areas

ISKANDER AKHMETOV

KBTU DATA SCIENCE LABORATORY, 2023

Outline

- Big Text Data – what is it?
- What can we do about it? Types of text data analytics
- Past projects
- Current projects

Big Text Data – what is it?

What is Big Data?

BIG DATA REFERS TO EXTREMELY LARGE DATA SETS THAT MAY BE ANALYZED COMPUTATIONALLY TO REVEAL PATTERNS, TRENDS, AND ASSOCIATIONS, ESPECIALLY RELATING TO HUMAN BEHAVIOR AND INTERACTIONS.

PROPERTIES: IT'S NOT JUST ABOUT THE DATA SIZE BUT ALSO ABOUT ITS COMPLEXITY, VARIETY, AND THE SPEED AT WHICH IT'S GENERATED.

SOURCES: BIG DATA CAN INCLUDE DATA FROM SOCIAL MEDIA, SENSOR NETWORKS, SCIENTIFIC EXPERIMENTS, MEDICAL RECORDS, MILITARY SURVEILLANCE, AND MANY OTHER SOURCES.

Key Characteristics of Big Data (5Vs):

Volume: The quantity of generated and stored data. The size of the data determines the value and potential insight, and whether it can be considered big data or not.

Velocity: The speed at which new data is generated and the speed at which data moves around. Social media posts can get hundreds of updates per minute, sensor data can flood in real-time, and high-frequency trading systems can generate millions of transactions per second.

Variety: The type and nature of the data. This helps people who analyze it to effectively use the resulting insight. Big Data can be structured, semi-structured, or unstructured.

Veracity: The quality of the data. With many forms of big data, quality and accuracy are less controllable (just think of Twitter posts with hashtags, abbreviations, typos and colloquial speech).

Value: It's all well and good having access to big data but unless we can turn it into value, it is useless. So, it could be said that 'value' is the most important V of Big Data.

Applications of Big Data

Science: analyzing sensor data, articles and proceedings, experimental image processing etc.

Healthcare: Analyzing patient records, treatments, and outcomes to find patterns that may guide future treatments.

Retail: Understanding customer purchase history and using it to recommend other purchases, manage inventory, and offer discounts.

Finance: Analyzing trends to optimize trading, manage risk, and detect fraud.

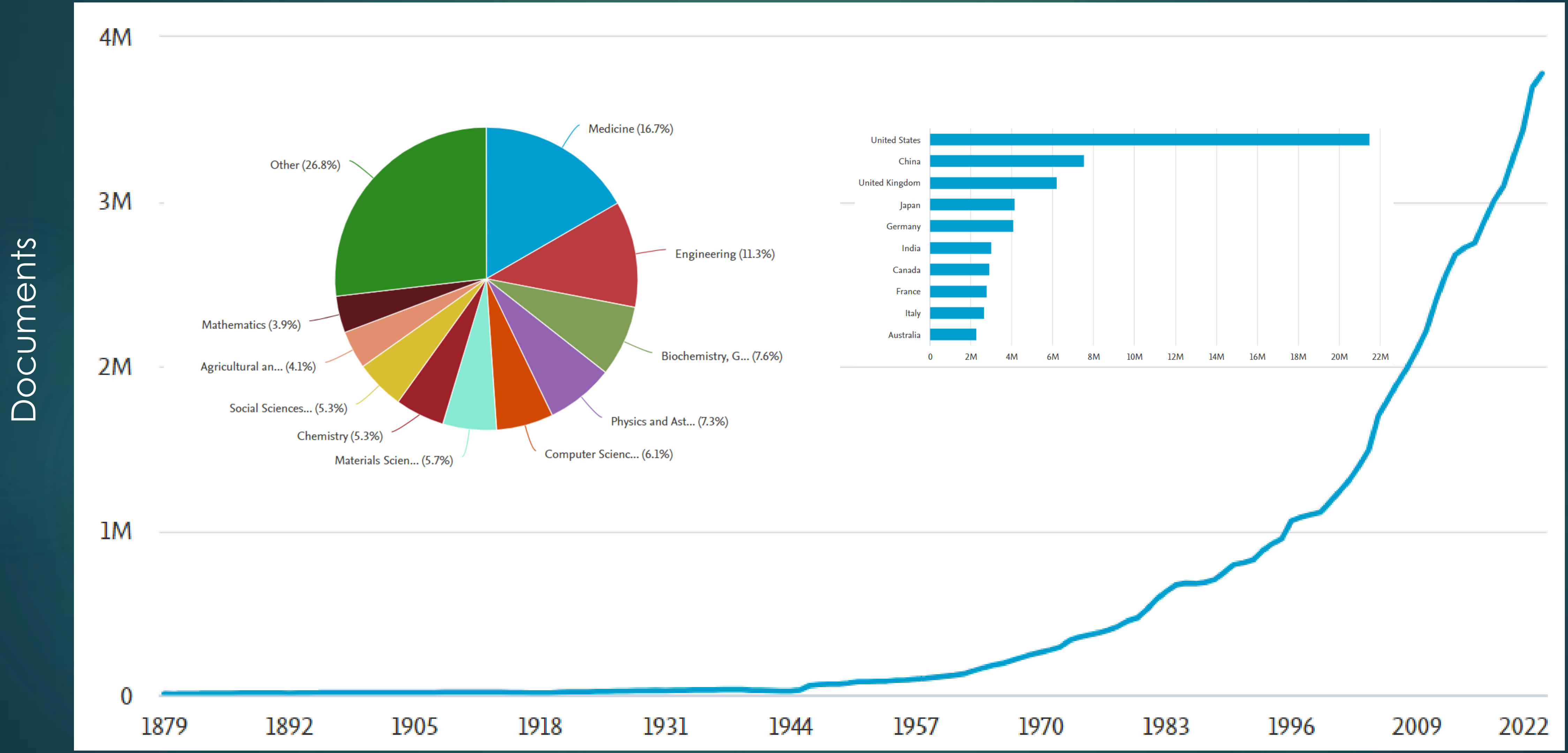
Manufacturing: Using sensors and other data to optimize processes, perform predictive maintenance, and improve quality.

Transportation: Optimizing routes, improving safety, and managing vehicle maintenance.

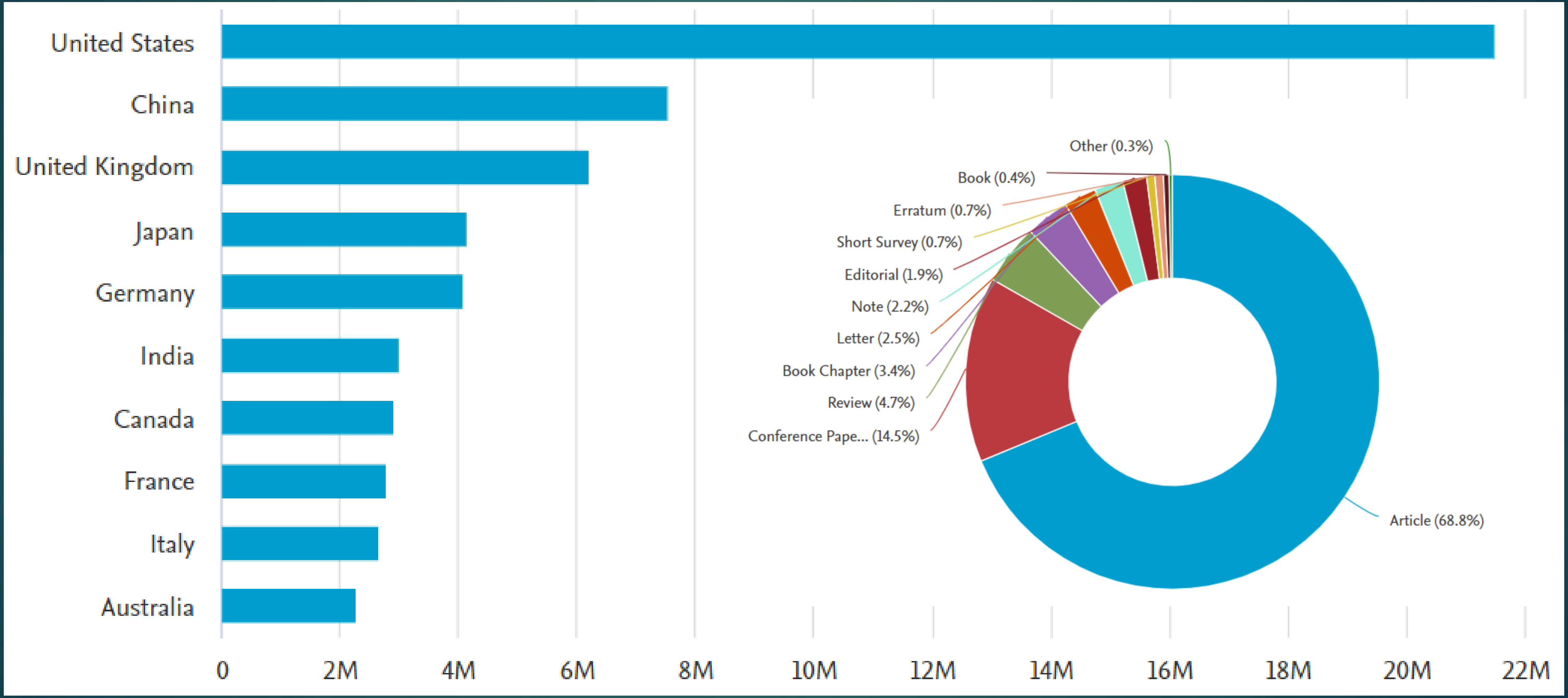
Challenges

- **Storage**: Managing and storing large volumes of data efficiently.
- **Processing**: Analyzing the data in a timely manner.
- **Security**: Protecting the data from unauthorized access and ensuring privacy.
- **Quality**: Ensuring that the data is reliable, accurate, and available for use.
- **Integration**: Combining data from different sources and making it available for analysis.

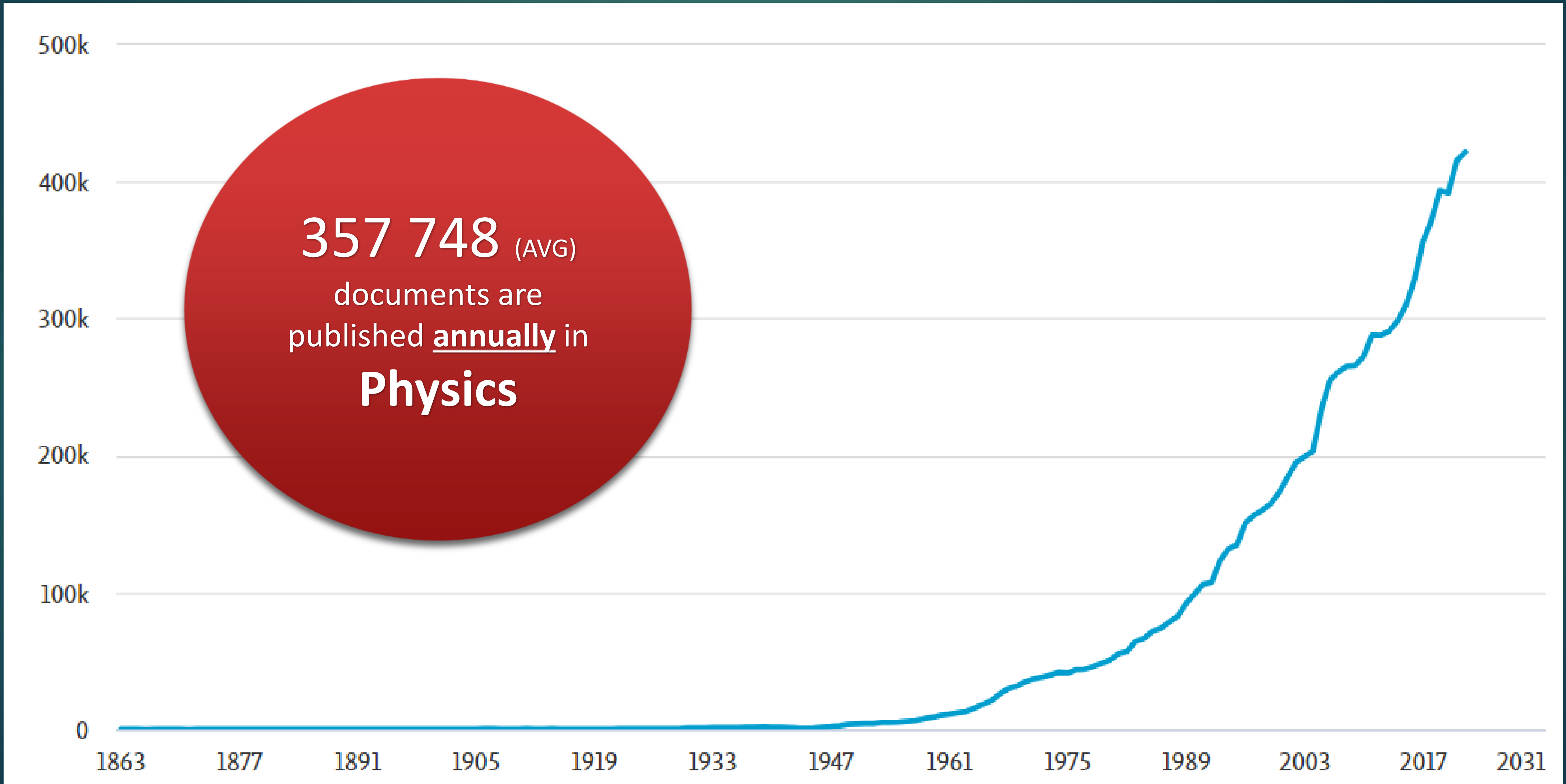
What is Big Text Data?



What is Big Text Data?



What about Physics?

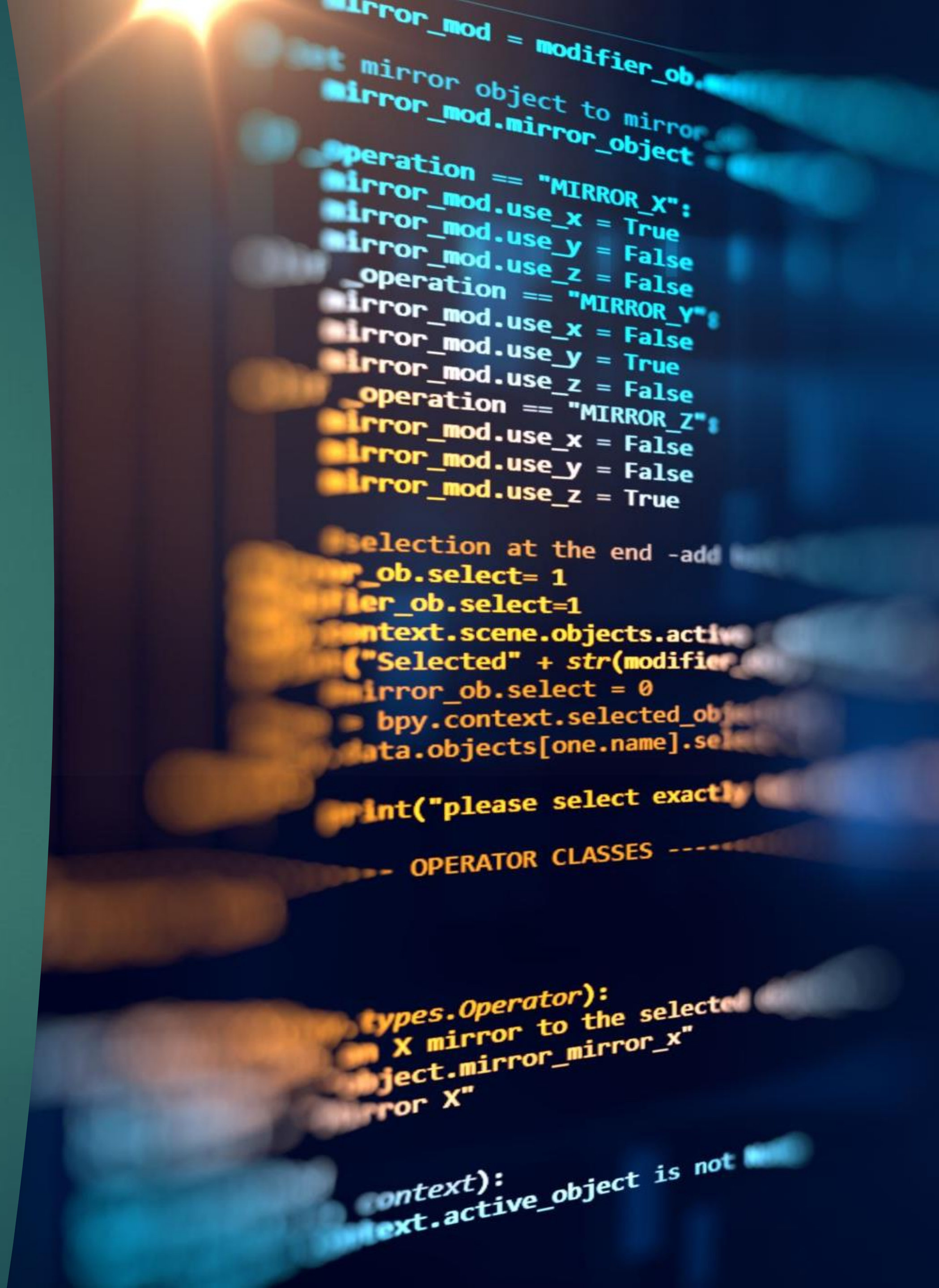


357 748 (AVG) documents are published **annually** in **Physics**

What can we do about it?

Types of text data analytics

- Text classification and clustering
- Sentiment analysis
- Topic modelling
- Summarization
- Question answering
- Named Entity Recognition (NER)
- Language detection
- Machine translation



```
irror_mod = modifier_ob.
t mirror object to mirror
irror_mod.mirror_object
peration == "MIRROR_X":
irror_mod.use_x = True
irror_mod.use_y = False
irror_mod.use_z = False
_operation == "MIRROR_Y":
irror_mod.use_x = False
irror_mod.use_y = True
irror_mod.use_z = False
_operation == "MIRROR_Z":
irror_mod.use_x = False
irror_mod.use_y = False
irror_mod.use_z = True

selection at the end -add
_ob.select= 1
ier_ob.select=1
ntext.scene.objects.activ
("Selected" + str(modifier
irror_ob.select = 0
 bpy.context.selected_obj
ata.objects[one.name].sel

int("please select exactl

- OPERATOR CLASSES ----

types.Operator):
 X mirror to the selected
ject.mirror_mirror_x"
rror X"

ontext):
xt.active_object is not
```

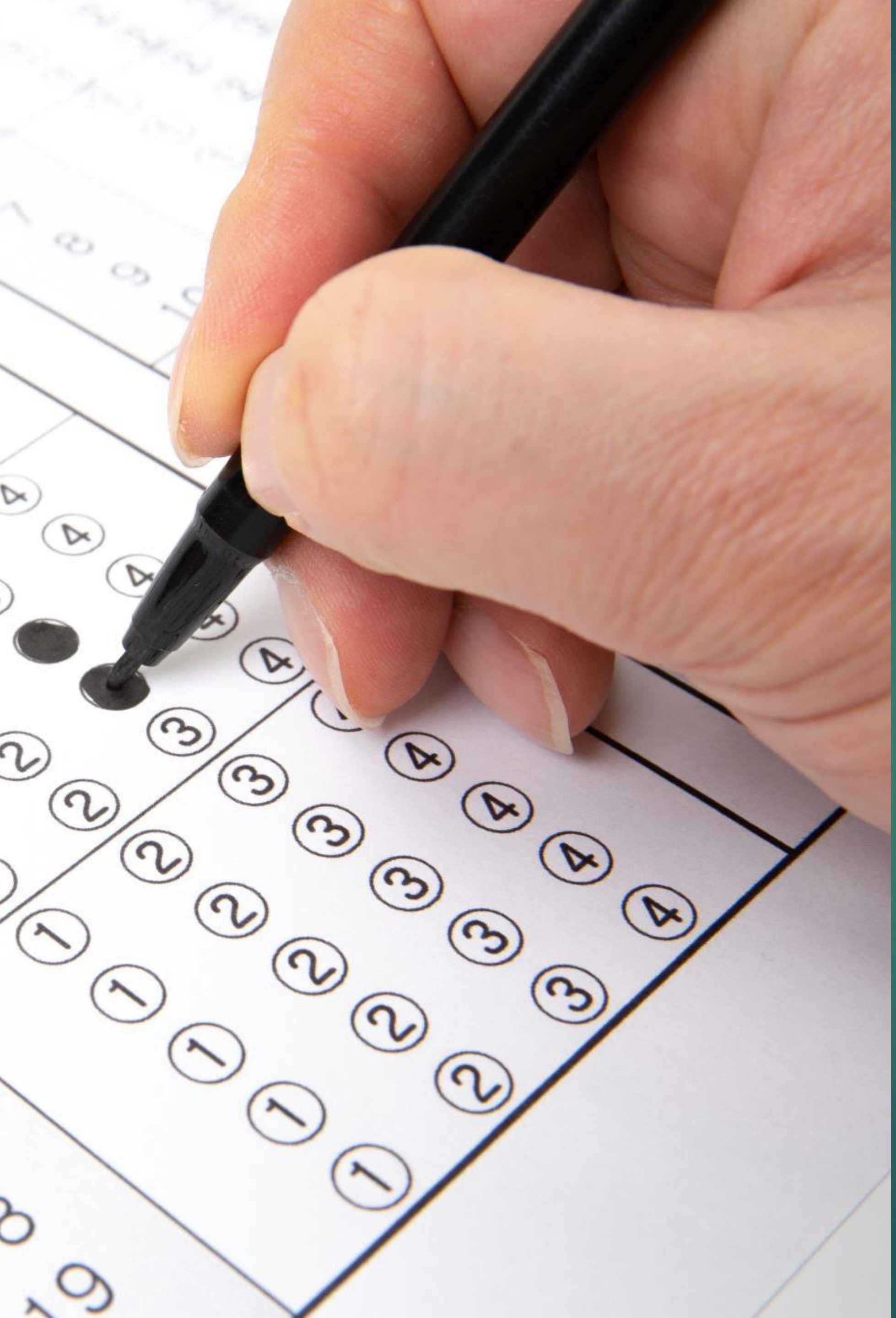
Past projects

Development of an approach for abstracting scientific texts in English

PH.D. CANDIDATE: ISKANDER AKHMETOV, KBTU.

SUPERVISOR: ALEXANDER PAK, KBTU.

FOREIGN SUPERVISOR: ALEXANDER GELBUKH, IPN.



Outline

- Problem Statement
- Application
- Research objectives
- Data
- Methods
- Evaluation
- Experiment
- Results
- Conclusion

Problem Statement

In today's world with the abundance of information we need tools to handle and digest large volumes of information in a short period of time.

Given a scientific article text extract the most important sentences to assemble an informative summary.

Why not just use the article abstract?

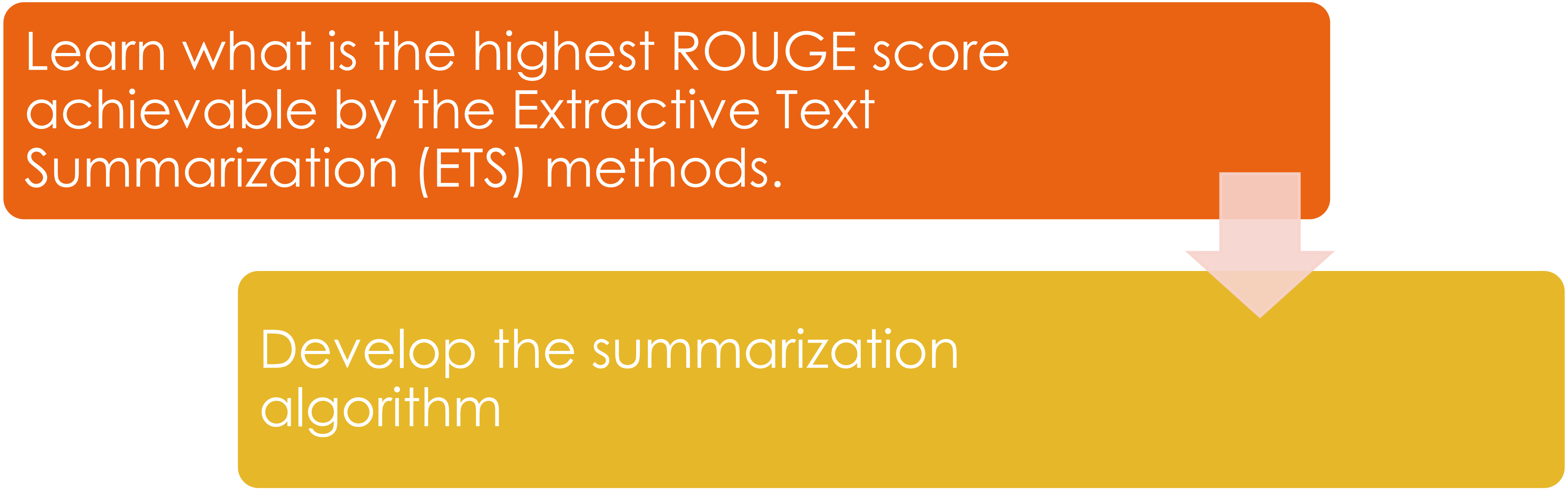
Article abstract might be an indicative summary of the article, which is just teasing to read the whole text.

The abstract itself might not be a good summary.

Applications

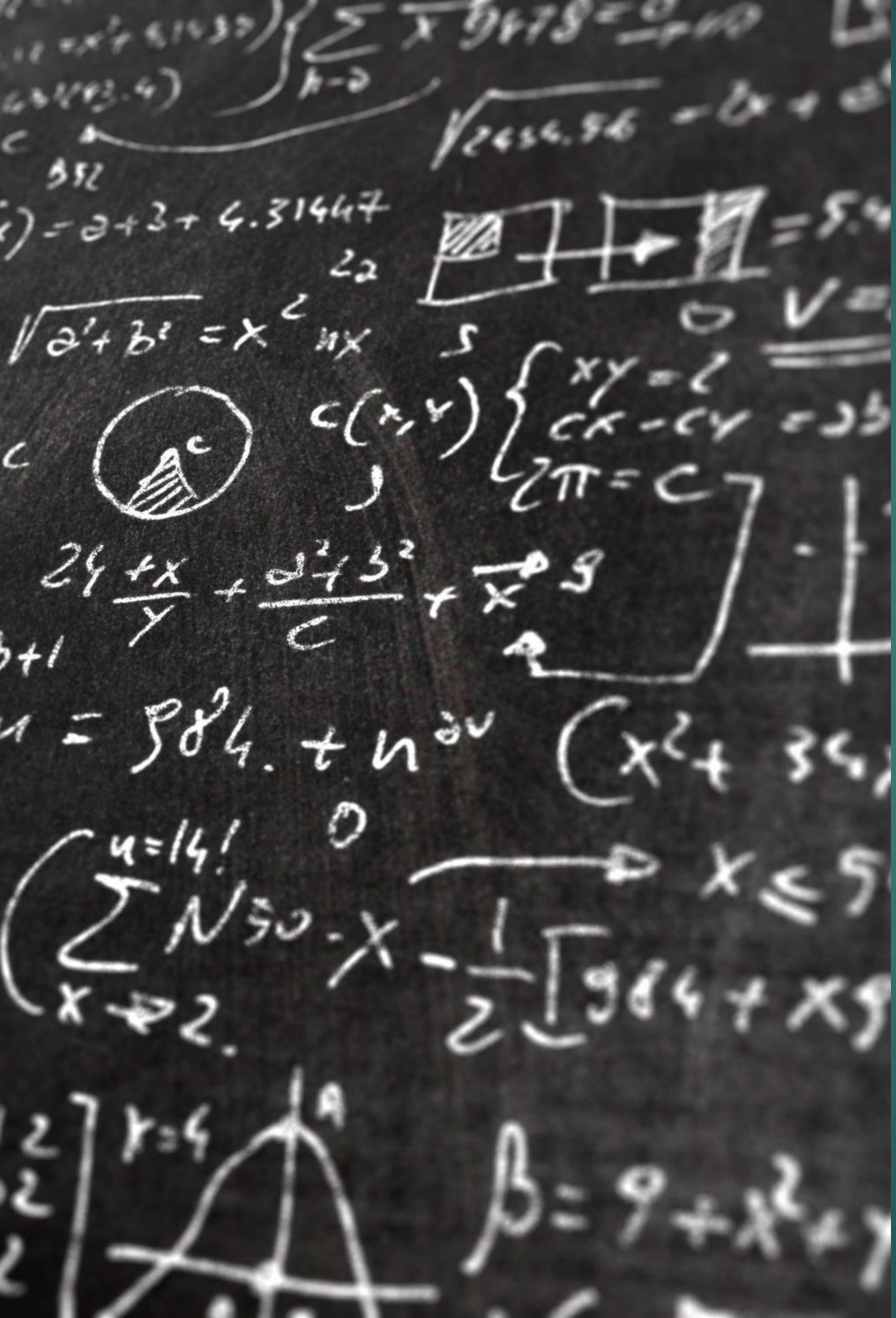
- Literature review
- Making notes automatically on the article you read
- Topic modeling

Research objectives



Learn what is the highest ROUGE score achievable by the Extractive Text Summarization (ETS) methods.

Develop the summarization algorithm

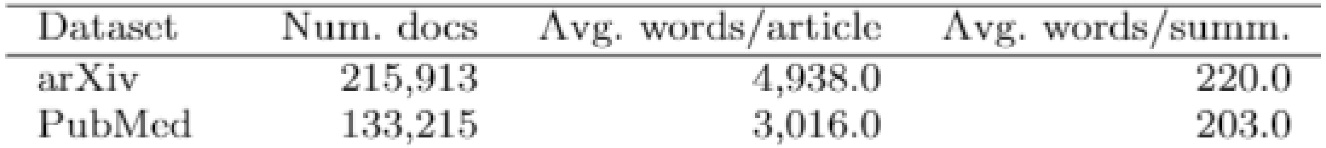


Contribution

1. Discovery of the top-line for the Extractive Summarization techniques (VNS, Greedy) and found that VNS initialized by the Greedy algorithm performs even better than any of the algorithms on their own for the task,
2. Proposing of an Extractive Summarization method based on a greedy algorithm that is performing on a high level despite its relative simplicity,
3. Cleaned dataset with different types of high-ROUGE summaries and useful text statistics.

Data

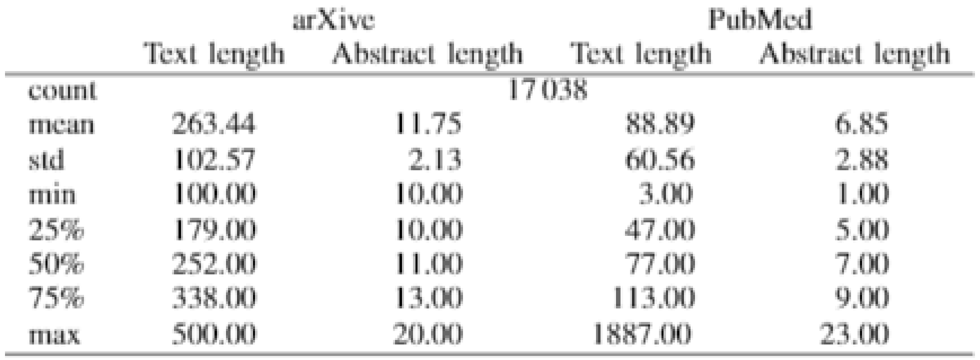
- ArXiv dataset for long document summarization contains 215,913 documents from the database official website and first collected by Cohan et al.
- The PubMed dataset consists of 133,215 scientific publications from the PubMed database (MEDLINE searching).
- Cleaning the datasets from too short/long abstracts/texts we left 17,038 articles for each of arXive and PubMed to work with.



Dataset	Num. docs	Avg. words/article	Avg. words/summ.
arXiv	215,913	4,938.0	220.0
PubMed	133,215	3,016.0	203.0

Data

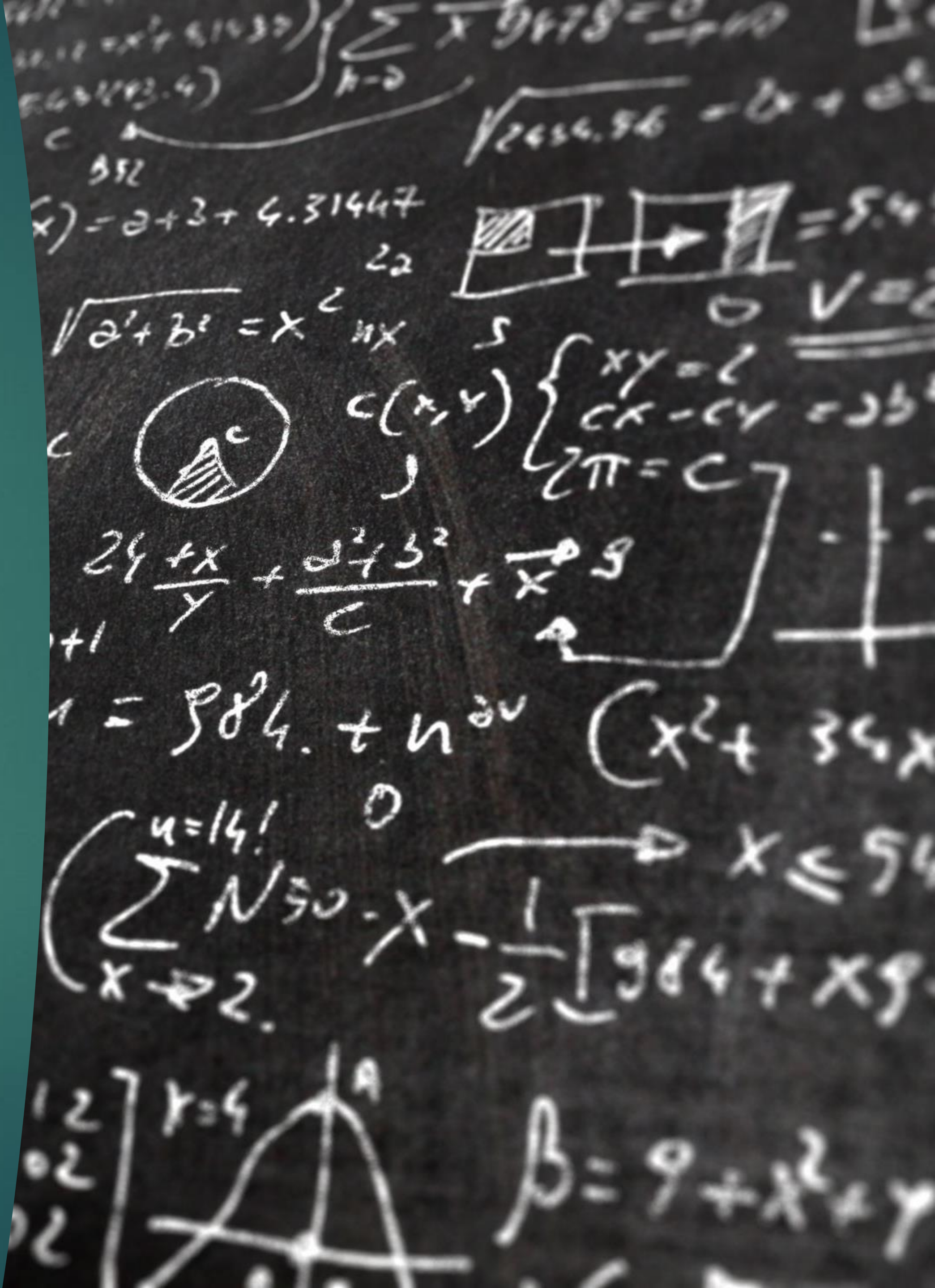
- Cleaning the datasets from too short/long abstracts/texts we left 17,038 articles for each of arXive and PubMed to work with.



	arXive		PubMed	
	Text length	Abstract length	Text length	Abstract length
count	17 038			
mean	263.44	11.75	88.89	6.85
std	102.57	2.13	60.56	2.88
min	100.00	10.00	3.00	1.00
25%	179.00	10.00	47.00	5.00
50%	252.00	11.00	77.00	7.00
75%	338.00	13.00	113.00	9.00
max	500.00	20.00	1887.00	23.00

The methods used in the research

- For finding the best ROUGE score achievable by the ETS:
    - Variable Neighborhood Search (VNS) by Mladenovic et al.
    - Greedy algorithm
- For the development of summarization algorithm
    - TFIDF
    - Greedy algorithm



Evaluation

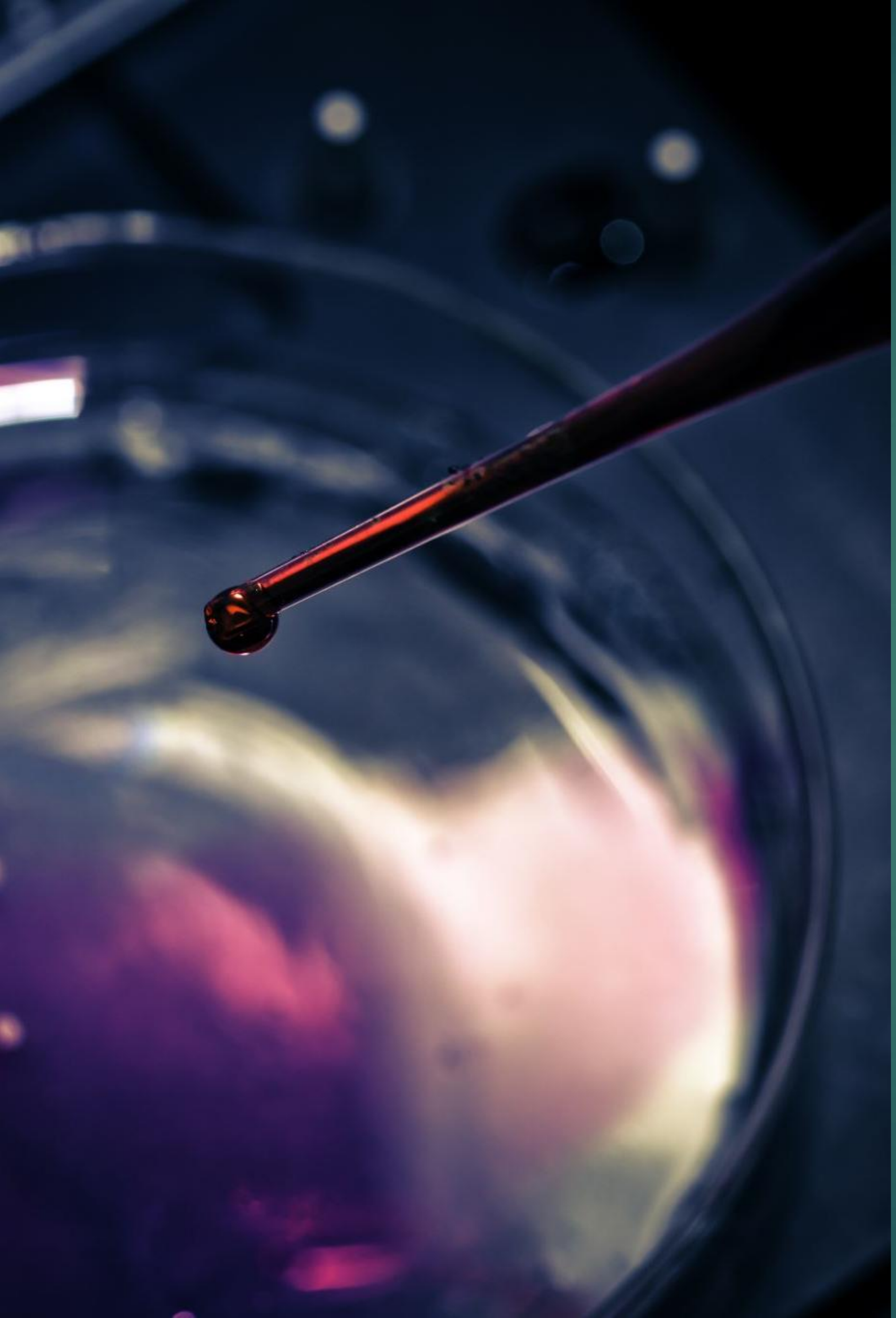
Recall-Oriented Understudy for Gisting Evaluation(ROUGE)

- The basic idea of the metric is based on calculating the share of intersection n-grams between candidate summary and reference summary, in reference and candidate summaries. The integration of recall and precision is calculated as the harmonic mean between two and is called the F1 score.

$$\text{recall} = \frac{\text{len}(C \cap R)}{\text{len}(R)},$$

$$\text{precision} = \frac{\text{len}(C \cap R)}{\text{len}(C)}.$$

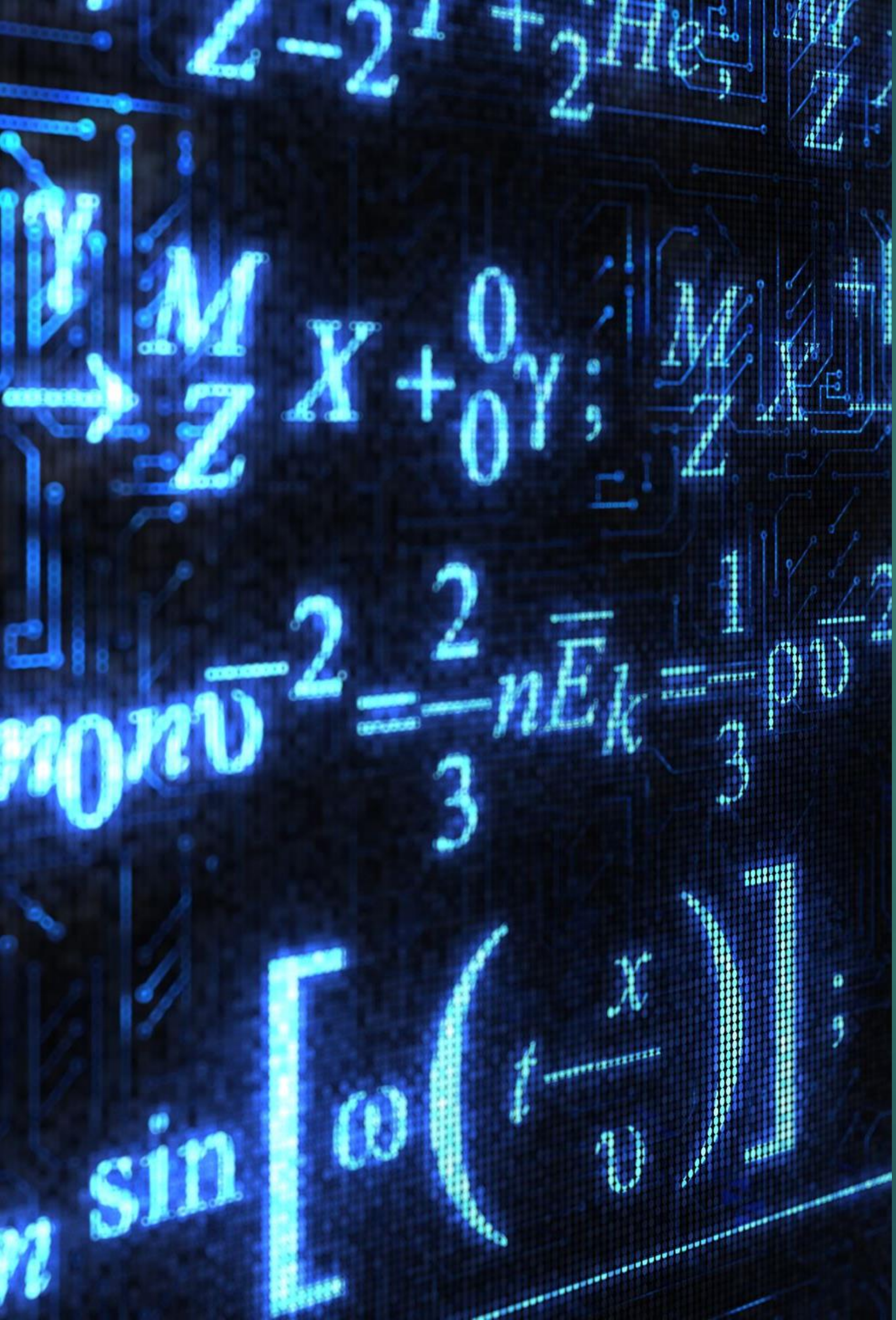
$$\text{F1 score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}.$$



Experiments: finding the best ROUGE score

VNS

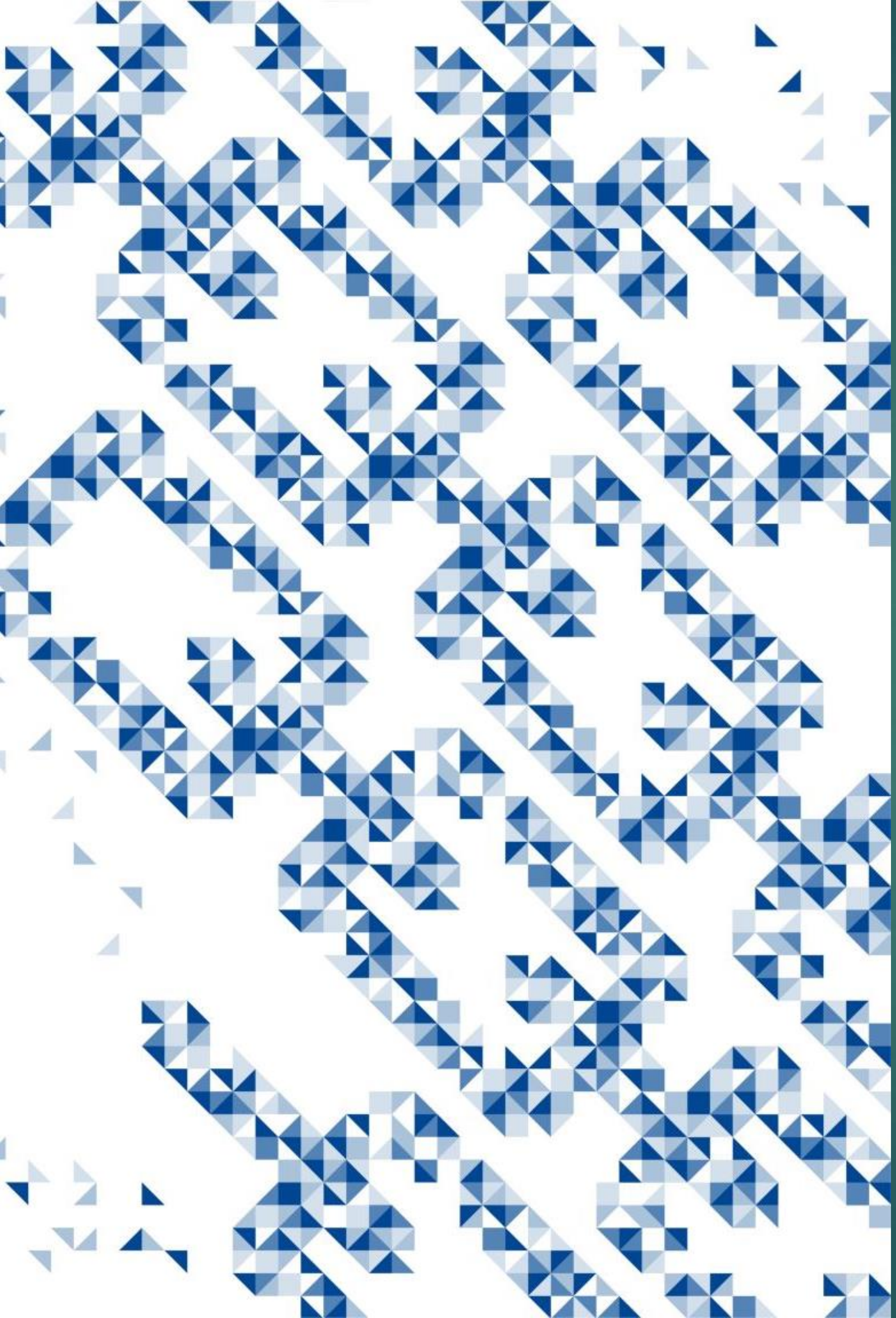
- **Initial solution** - is the first, usually random approximation of the objective function. We initialize our search for a solution by a random set of sentences x in Nk=($\frac{N_t}{N_a}$) space of possible neighborhood structures.
- **Shaking** - is the process of systematical modification of the initial solution to the extent specified by the k_{max} parameter.
- **Incumbent solution** - is the best current solution achieved after shaking.
- **Stop condition** - the cycle is limited by 5000 iterations or 60 seconds. Suppose no ROUGE-1 score improvement occurs after 700 consecutive iterations, the cycle breaks.



Experiments: finding the best ROUGE score

Greedy algorithm

1. Obtain a unique word list from A as a Vocabulary (V).
2. Build matrix M, where for each sentence in T as rows, we determine the presence of the words from V in binary mode.
3. Until M is empty:
   - Sum the M along 0 axis (or rows) and obtain the maximum value index, the sentence index containing a maximum of words from the A. Save the index to an Index List (IL).
   - Update the M by deleting the columns corresponding to non-zero values for the sentence with the maximum number of words from A.
4. To get the number of sentences in summary resulting in the maximum ROUGE score:•
   - Calculate ROUGE scores for all combinations of sentences starting from 1 up to a length of IL.
   - Select the number of sentences with the maximum ROUGE score.
   - Update IL with the best sentence combination.
5. Sort indices in IL in the ascending order and recover the original order of the sentences in the article.
6. Collect summary by taking sentences from T by the indices in sorted I.
7. Compute ROUGE score between generated summary and A.



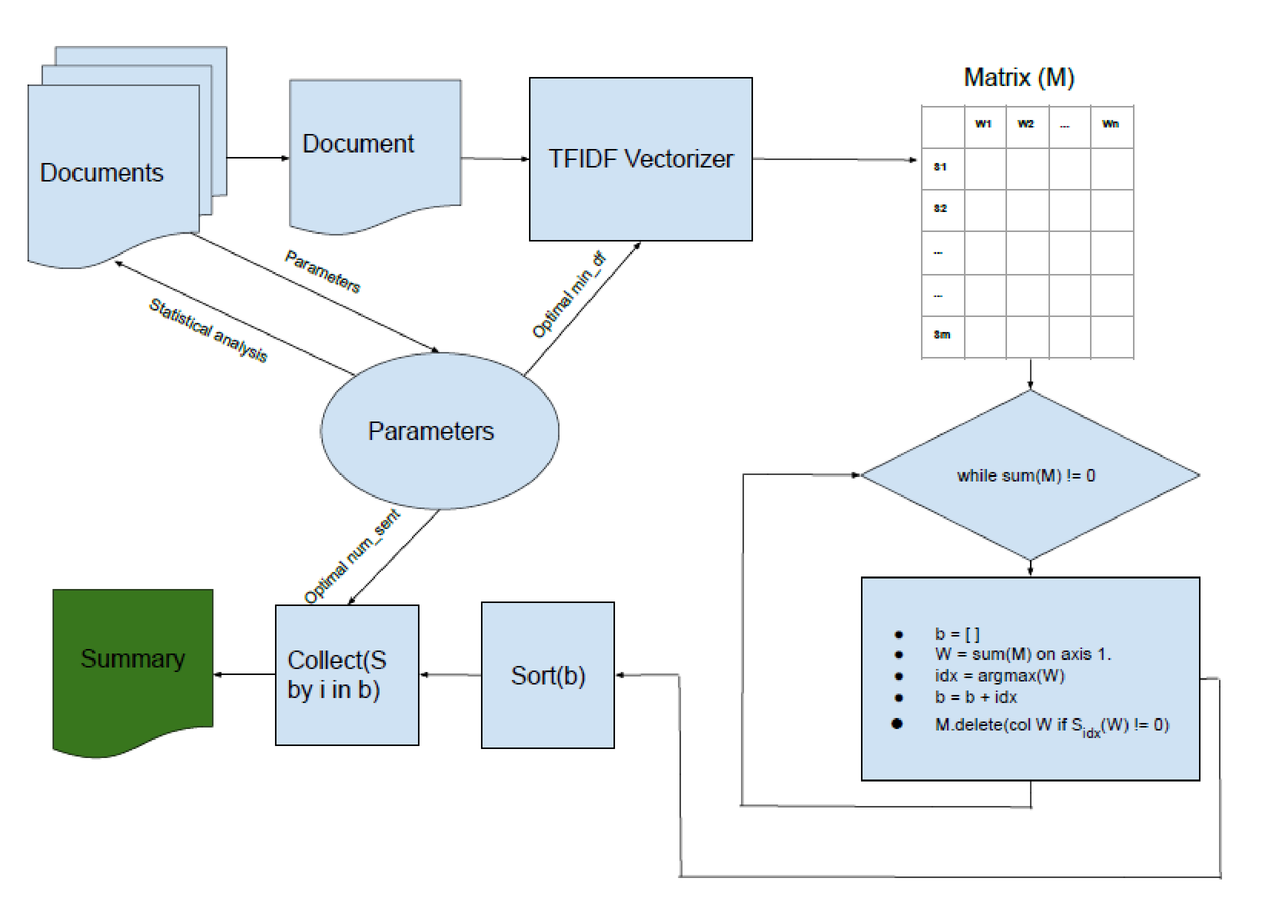
Experiments: summarization algorithm

Greedy algorithm

Given an Article full text (T) separated into Sentences(S):

1. TfIdfVectorize T in with min_df=0.042 (found empirically) returning matrix (M). We use here TfidfVectorizer, because this time we have to account for the importance of the words in the text.
2. Until Mx is empty:
   - Sum the M along 0 axis (or rows) and obtain the maximum value index, the sentence index containing a maximum of TFIDF value words from the T. Save the index to an Index List (IL).
   - Update the M by deleting the columns corresponding to non-zero values for the sentence with the maximum sum of TFIDF value words from T.
3. We take the top 8 sentence indices from the IL with the maximum sum of TFIDF value words from T. On average, the Greedy gets approximately eight sentences (or 7.7 sentences to be more precise) to achieve the maximum ROUGE score.
4. Sort indices in IL in the ascending order to recover the original sequence order of the sentences in the article.
5. Collect summary by taking sentences from T by the indices in sorted IL.
6. Compute ROUGE score between generated summary and A.

Experiments: summarization algorithm

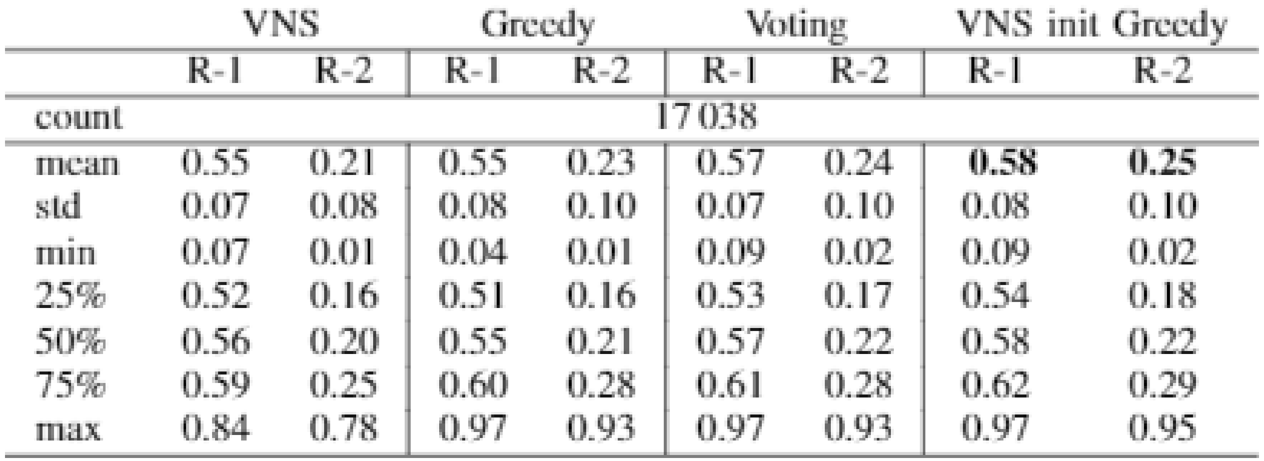


Matrix (M)

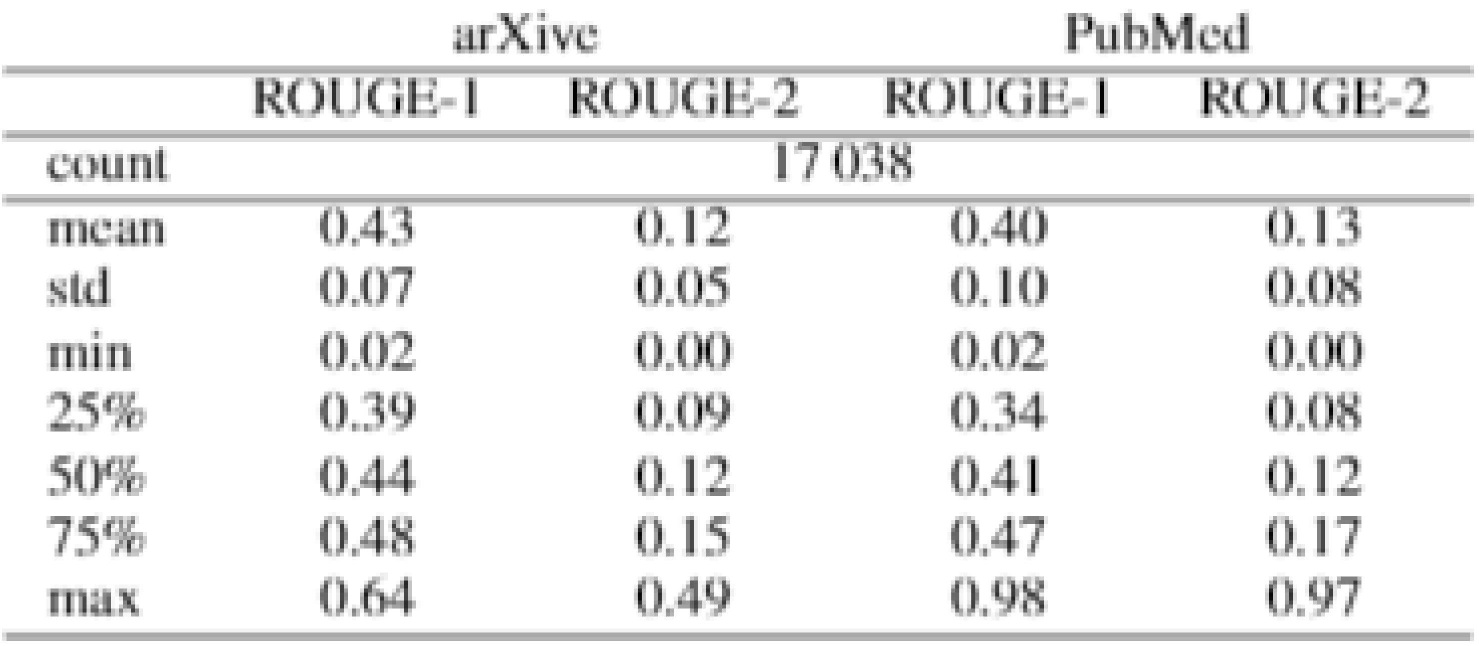
	W1	W2	...	Wn
S1				
S2				
...				
...				
Sm				

- b = [ ]
- W = sum(M) on axis 1.
- idx = argmax(W)
- b = b + idx
- M.delete(col W if S_{idx}(W) != 0)

Results: finding the best ROUGE score



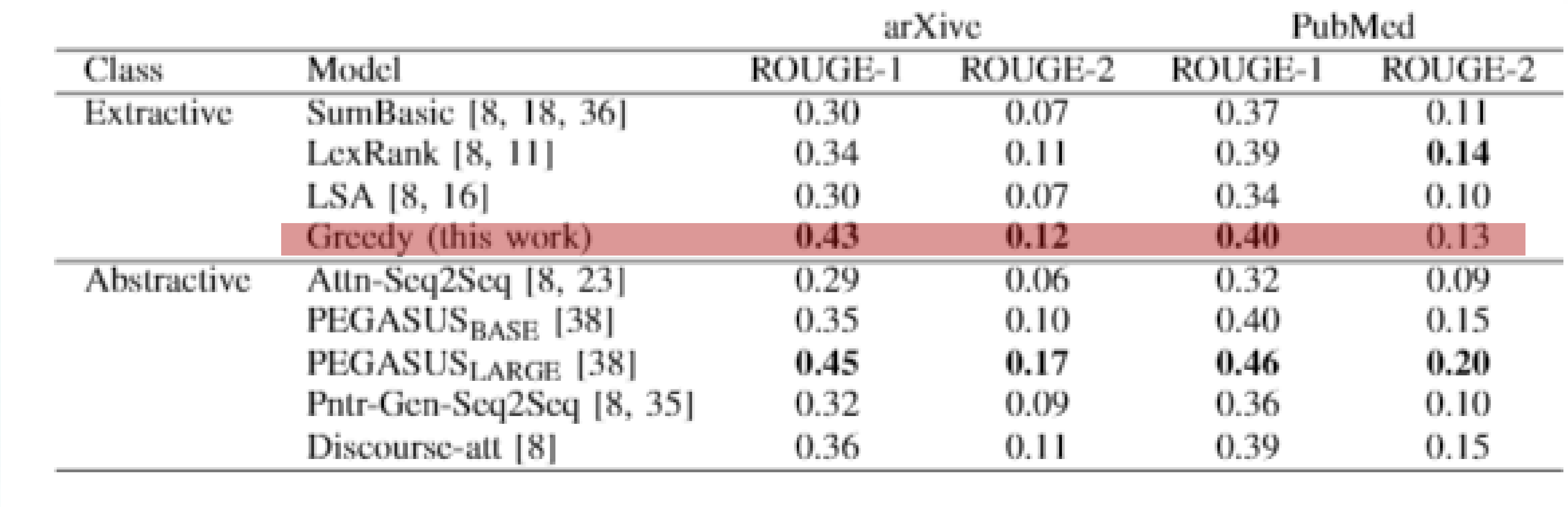
	VNS		Greedy		Voting		VNS init Greedy	
	R-1	R-2	R-1	R-2	R-1	R-2	R-1	R-2
count				17 038				
mean	0.55	0.21	0.55	0.23	0.57	0.24	**0.58**	**0.25**
std	0.07	0.08	0.08	0.10	0.07	0.10	0.08	0.10
min	0.07	0.01	0.04	0.01	0.09	0.02	0.09	0.02
25%	0.52	0.16	0.51	0.16	0.53	0.17	0.54	0.18
50%	0.56	0.20	0.55	0.21	0.57	0.22	0.58	0.22
75%	0.59	0.25	0.60	0.28	0.61	0.28	0.62	0.29
max	0.84	0.78	0.97	0.93	0.97	0.93	0.97	0.95



	arXive		PubMed	
	ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2
count	17 038			
mean	0.43	0.12	0.40	0.13
std	0.07	0.05	0.10	0.08
min	0.02	0.00	0.02	0.00
25%	0.39	0.09	0.34	0.08
50%	0.44	0.12	0.41	0.12
75%	0.48	0.15	0.47	0.17
max	0.64	0.49	0.98	0.97

Results: summarization algorithm

Results: summarization algorithm



		arXive		PubMed	
Class	Model	ROUGE-1	ROUGE-2	ROUGE-1	ROUGE-2
Extractive	SumBasic [8, 18, 36]	0.30	0.07	0.37	0.11
	LexRank [8, 11]	0.34	0.11	0.39	**0.14**
	LSA [8, 16]	0.30	0.07	0.34	0.10
	Greedy (this work)	**0.43**	**0.12**	**0.40**	0.13
Abstractive	Attn-Seq2Seq [8, 23]	0.29	0.06	0.32	0.09
	PEGASUS$_{\text{BASE}}$ [38]	0.35	0.10	0.40	0.15
	PEGASUS$_{\text{LARGE}}$ [38]	**0.45**	**0.17**	**0.46**	**0.20**
	Pntr-Gen-Seq2Seq [8, 35]	0.32	0.09	0.36	0.10
	Discourse-att [8]	0.36	0.11	0.39	0.15

Conclusion

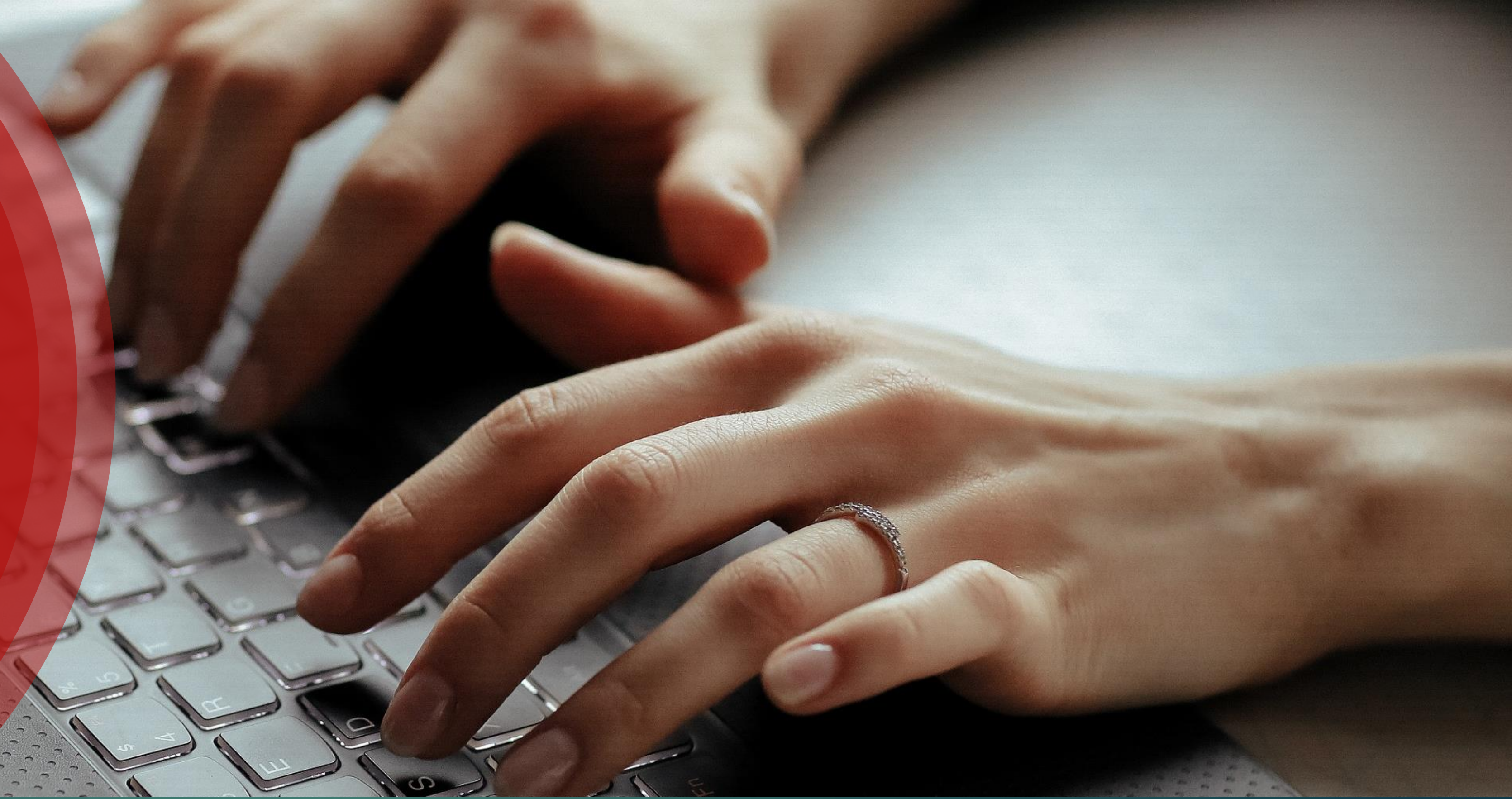
- This work showed two methods to determine the best possible ROUGE score for Extractive Summarization on the arXive dataset. The first one involved the VNS technique and was used in our previous publication, and the second approach employed a greedy algorithm. Both approaches showed similar performance, but the Greedy approach takes significantly less time to complete. The study showed that there is still room to develop Extractive Summarization methods as the best possible ROUGE-1 score is on average0.55 while state-of-the-art models do not surpass a 0.50 level.
- The use of the Greedy approach induced us to use a modified algorithm for Extractive Summarization, and it showed results comparable to those of state-of-the-art models in our tests. Thus, generally, we can say that Extractive Methods of Summarization still have the potential for development, supporting the opinion of Sebastian Ruder when he says that the significant role is played not by the complexity of the method but by proper hypeparameter tuning and data preprocessing.

Development of machine learning methods for increasing text coherence in the problem of summarizing large text arrays

Prepared by: Dilyara Akhmetova

OUTLINE

- Problem statement
- Contribution to scientific knowledge
- Coherence and cohesion
- Dataset used
- Experiment design and evaluation
- Results and conclusion



> “In recent years there has been a surge in the number of published academic papers, with > 7 million new papers
> each year and > 1.8 million papers with ⩾ 5 references”
>
> Fire, M., Guestrin, C.

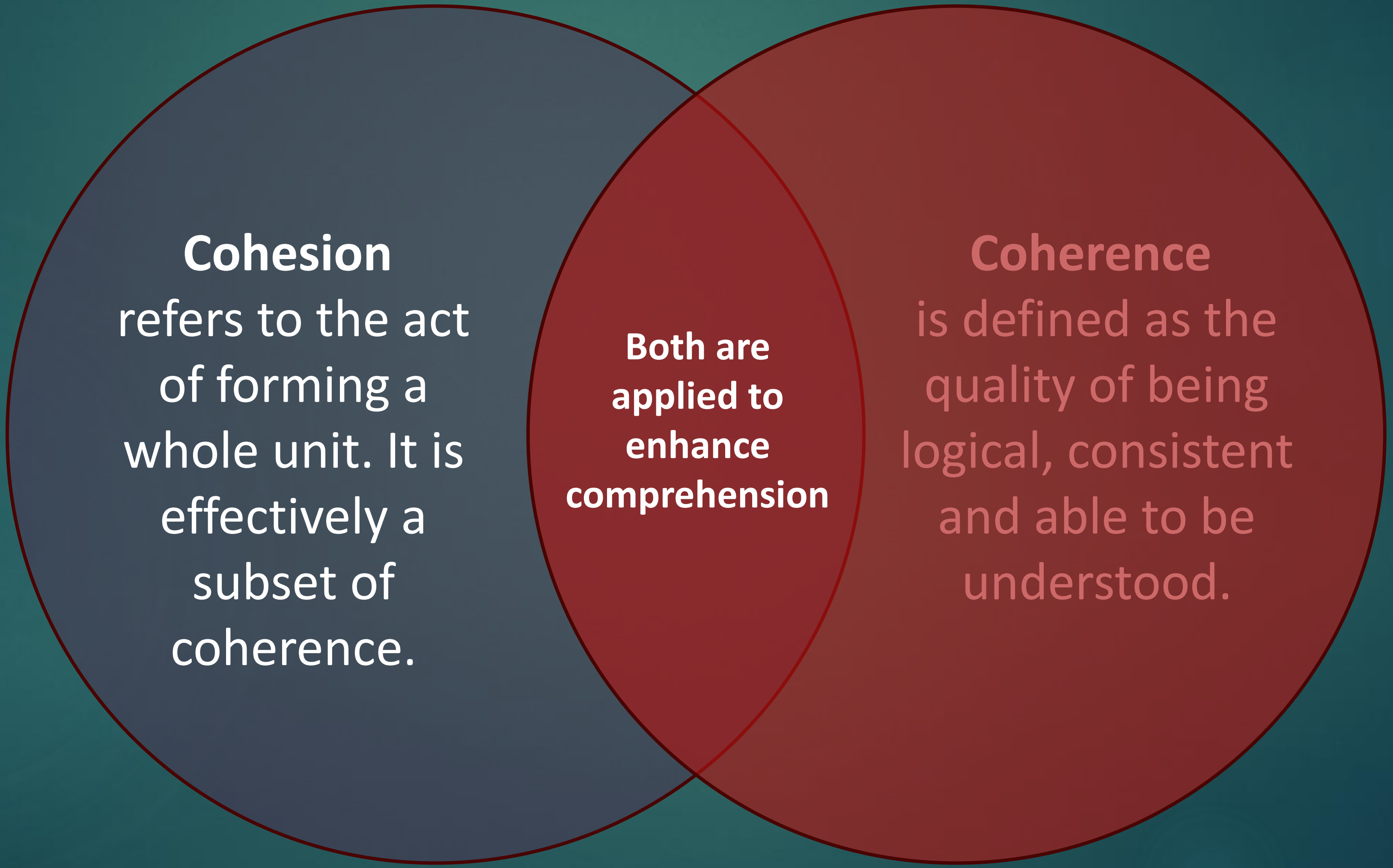
PROBLEM STATEMENT

The main objective of this work is an increasing coherence of the summaries, obtained by extractive summarization method. It means that the readability and understanding of the text will be improved.

CONTRIBUTION TO SCIENTIFIC KNOWLEDGE

- We applied Large Language Model (LLM) fine-tuning approach to solve a specialized problem of increasing coherence level in summaries, obtained by extractive summarization method.
- We developed a method for determining the coherence of the text

COHESION AND COHERENCE



Cohesion
refers to the act of forming a whole unit. It is effectively a subset of coherence.

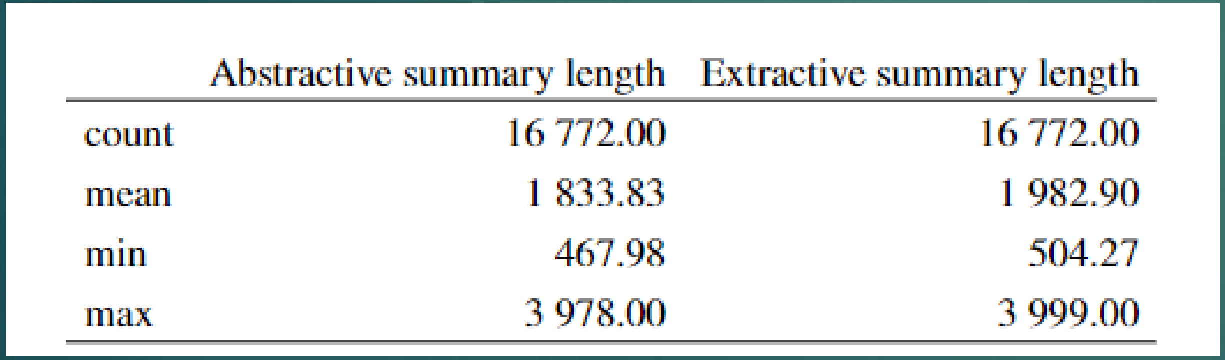
Both are applied to enhance comprehension

Coherence
is defined as the quality of being logical, consistent and able to be understood.

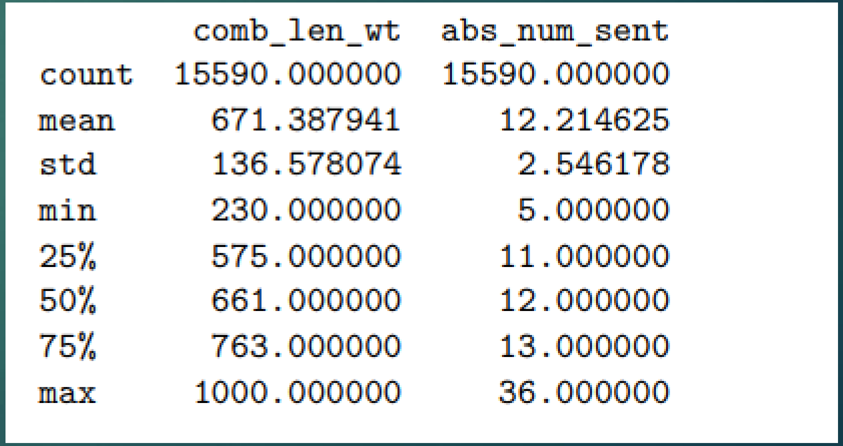
DATA

We experimented with the dataset containing 16 772 pairs of extractive and corresponding abstractive summaries of scientific papers in English. The dataset was obtained during the work of determining the possible upper boundaries of extractive summary quality.

Reaching for upper bound ROUGE score of extractive summarization methods /
Iskander Akhmetov, Alexander Gelbukh, Rustam Mussabayev // – 2022. – Vol. 8:e1103



	Abstractive summary length	Extractive summary length
count	16 772.00	16 772.00
mean	1 833.83	1 982.90
min	467.98	504.27
max	3 978.00	3 999.00



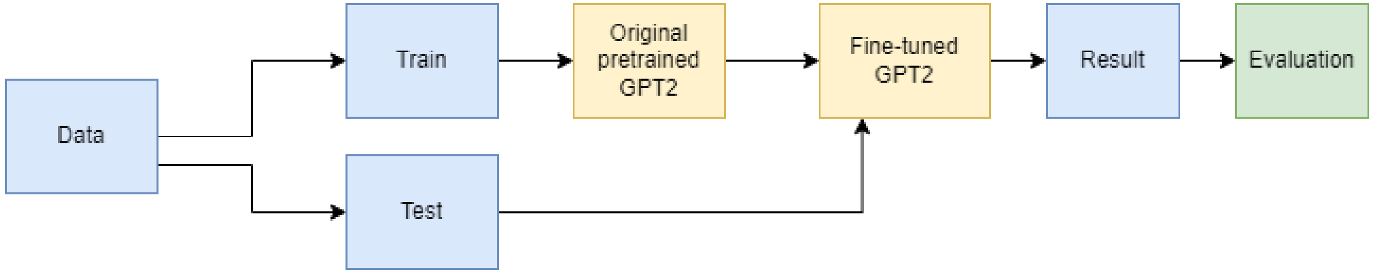
```
        comb_len_wt   abs_num_sent
count  15590.000000   15590.000000
mean     671.387941      12.214625
std      136.578074       2.546178
min      230.000000       5.000000
25%      575.000000      11.000000
50%      661.000000      12.000000
75%      763.000000      13.000000
max     1000.000000      36.000000
```

EXPERIMENT

- Multi-class classification approach failed due to overfitting. This approach gives sub-optimal results. Fighting overfitting requires large memory capacities.
- Sequence-to-sequence (Seq2seq) approach failed and yielded no practical results due to the following reasons:
  - Character-based approach works for very small texts as phrases or single sentences but fails for larger texts.
  - Word-based approach required more than 81GB memory on GPU and ended with Memory Exhaustion Error.

EXPERIMENT (LLM approach)

- The dataset was prepared to have the combined length of extractive and abstractive
summaries not more than 1000 tokens.
- Added the input and output text signals "E:"token at the beginning of the extractive summary and "A:"token at the beginning of the abstractive summary.
- The texts were tokenized and divided into the train and test subsets.
- Original pretrained GPT2 was fine-tuned.
- Fine-tuned GPT2 model was tested.



EXPERIMENT

E:in addition a curve can also be defined as a connected one dimensional compact point set. by effectivizing all these approaches to curves open or namely the sets of points covered by the four classes of computable curves are different. if we look only at curves of computable lengths then the four effectivizations mentioned above are indeed equivalent. namely the sets of points covered by the four classes of computable curves are different. let xmath be the part of xmath in the box xmath. this means that the definition of computable curves is robust at least for curves of computable lengths. on the other hand gu lutz and mayordomo constructed in xcite a computable curve of non computable length such that none of its computable parametrizations can be injective open or i.e. if the class of computable curves of computable lengths is point separable from the class of computable curves of finite lengths. a positive answer will be given in this paper. we will see that these four versions of computable curves are all different.

G: a nontrivial set of classes of curves of computable lengths is studied. its existence has been proved for curves of computable lengths in the previous section. for the class of curves of computable lengths xmath the existence of a class of nontrivial sets of curves with nontrivial set of classes is proved for non trivial classes of curves of nontrivial sets of computable lengths. this class contains the set of non trivial curves of computable length of non computable length that can be injective or i.e. if the class of curves of computable lengths is point separable from the class of nontrivial sets of computable length of finite lengths. the class of nontrivial sets of curves is defined as the ones that satisfy certain conditions of the class of curves of computable lengths for which the class of nontrivial sets of curves has not been proved yet. for the class of nontrivial sets of curves of nontrivial sets of nontrivial sets of computable lengths a nontrivial set is always open under some conditions. keywords discrete geometry discrete curves nontrivial sets nontrivial sets mathematics subject classification.

EVALUATION

1. Coherence metric

Coefficient of Variation approach

The coherence measure proposed in this work is based on calculating the cosine similarities between every two consecutive pairs of sentences in the text in their respective vector representations

$$cossim = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{|\mathbf{A}||\mathbf{B}|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2} \sqrt{\sum\limits_{i=1}^{n} B_i^2}},$$

$$SeqSims = (cossim(S_i, S_{i+1}), i \in (1, n-1)),$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \overline{x})^2}{n-1}},$$

$$CV = \frac{\sigma}{\mu},$$

$$coh = \frac{1}{CV},$$

2. Text similarity metric

As for the cosine similarity measures, we observe output preserves the similarity to the input and reference text, so we might expect no effect of neural network hallucinations when it generates something completely different and out of context. We used cosine similarity metric between vector representation of the texts.

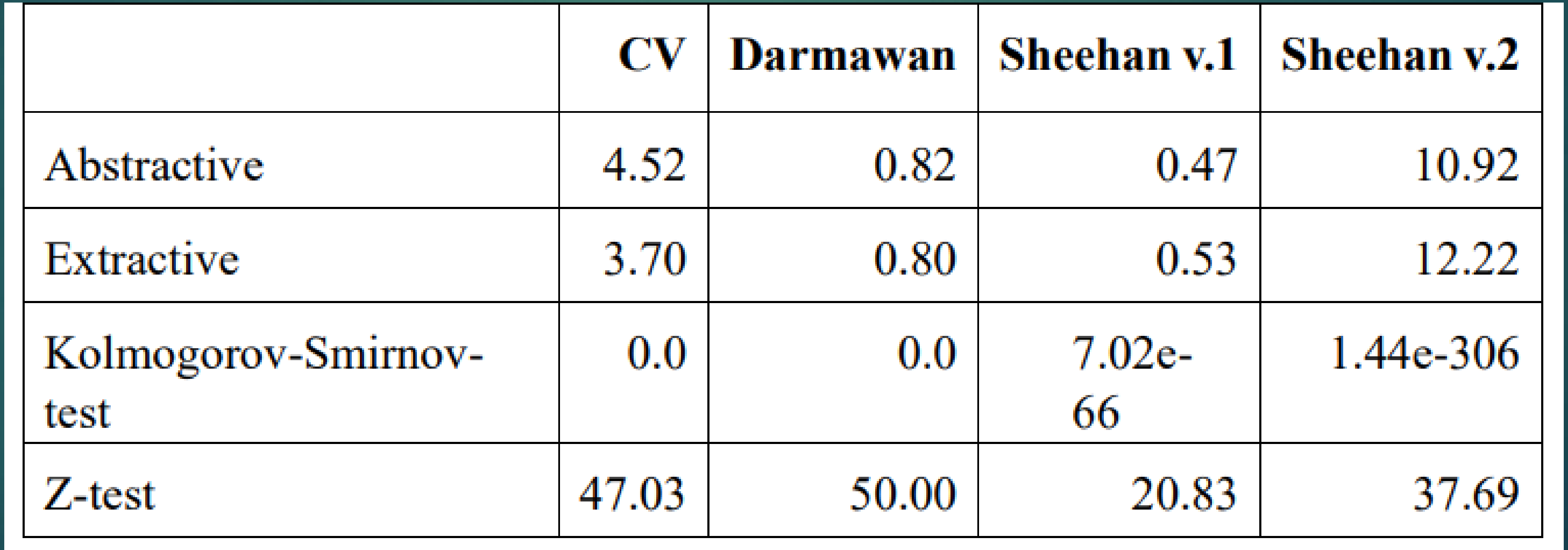
EVALUATION

For the test set average coherence of the input texts, which were totally extractive summaries and output, or predicted texts differ significantly, and we observe the increase in coherence measure for the output relative to the input text:

- P-value for the Kolmogorov-Smirnov test is 1.72e-24 which is smaller than 0.05, so we reject the null hypothesis and check the two samples are from different distributions.
- Z-test is 10.80, which is more than 3.0, so the two samples are highly significantly different.

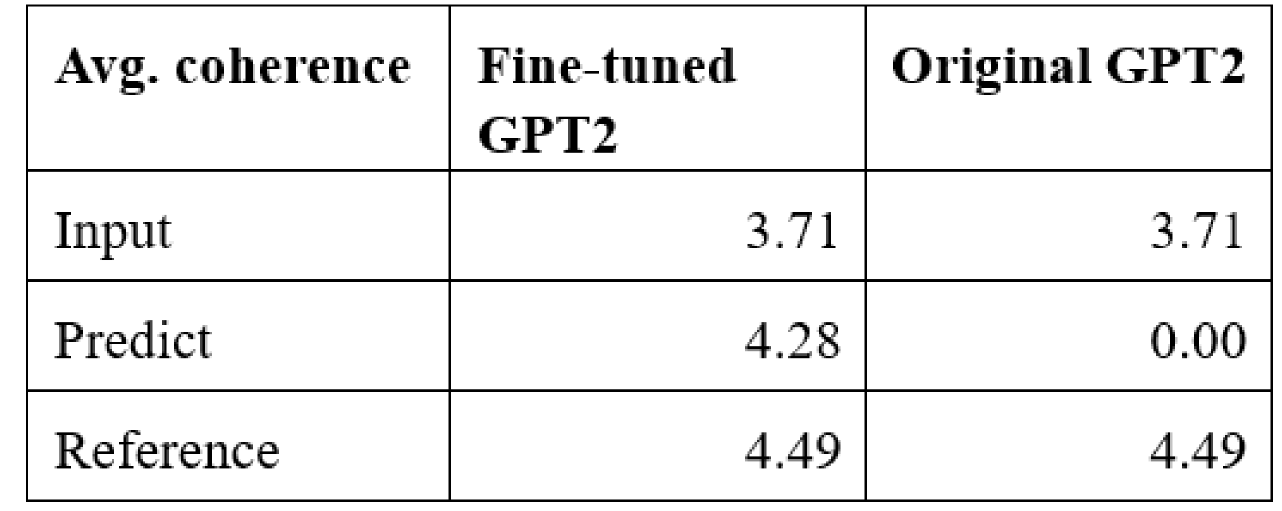
RESULTS

Beforehand we have measured the coherence of abstractive and extractive summaries in our dataset by 4 different methods including our own. And we have found that indeed the extractive summaries have lower coherence than abstractive summaries and their differences is statistically significant.

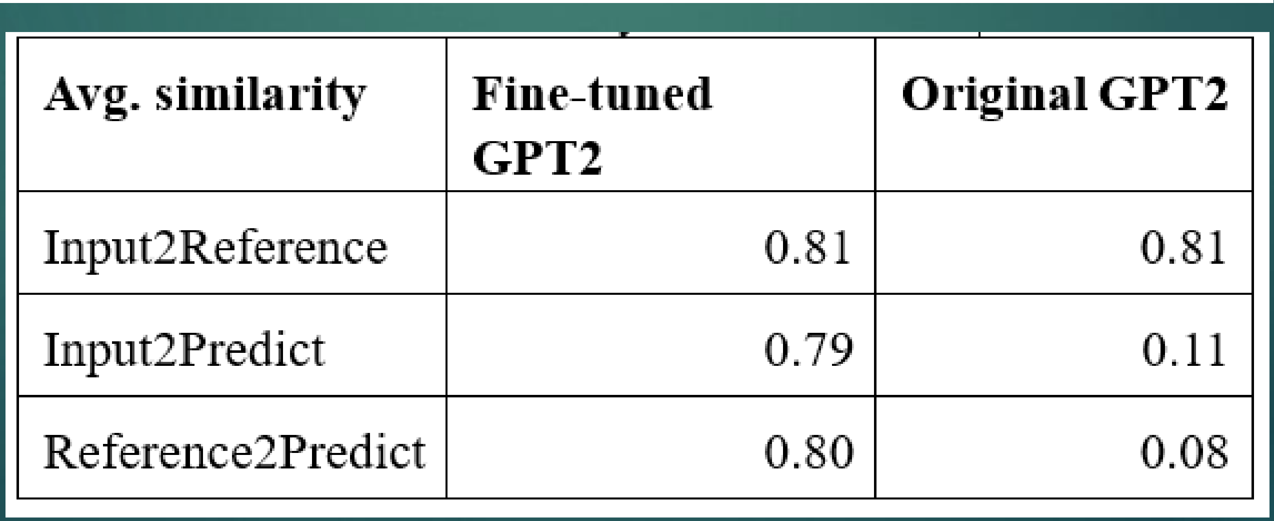


	CV	Darmawan	Sheehan v.1	Sheehan v.2
Abstractive	4.52	0.82	0.47	10.92
Extractive	3.70	0.80	0.53	12.22
Kolmogorov-Smirnov-test	0.0	0.0	7.02e-66	1.44e-306
Z-test	47.03	50.00	20.83	37.69

RESULTS



Avg. coherence	Fine-tuned GPT2	Original GPT2
Input	3.71	3.71
Predict	4.28	0.00
Reference	4.49	4.49



Avg. similarity	Fine-tuned GPT2	Original GPT2
Input2Reference	0.81	0.81
Input2Predict	0.79	0.11
Reference2Predict	0.80	0.08

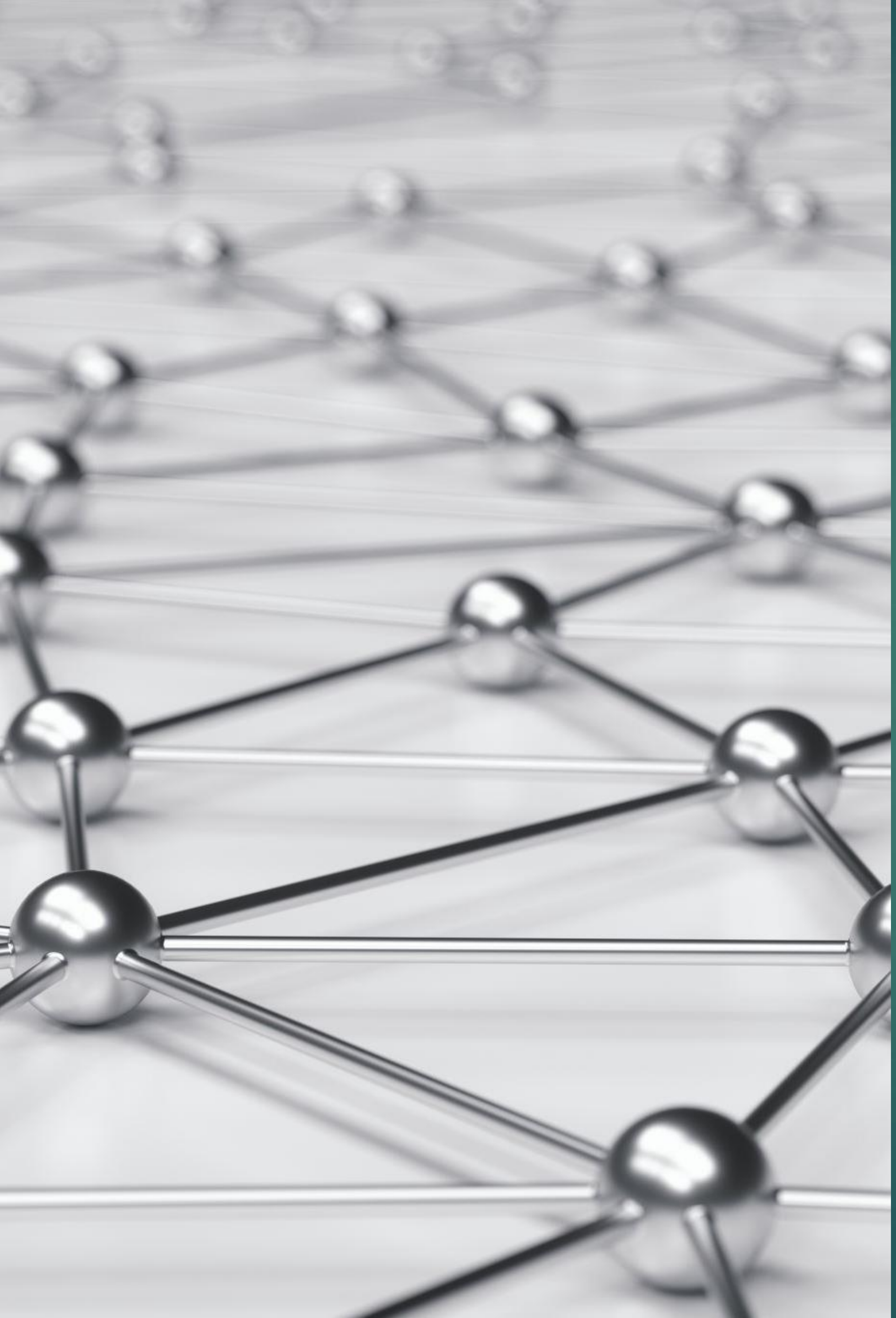
APPLICATION

- Literature review automation
- Article abstract generation
- Reduction of human information processing time
- Summary readability improvement

CONCLUSION

- The extractive summary was smoothed with Large Language Model (LLM) fine-tuning approach.
- We observe the increase in coherence measure in the generated text relative to the input text.
- Extractive methods still have potential for development.

Current projects



Decision support system in space industry

- Topic trend analysis
- Text summarization
- Query focused multi-document text summarization
- Automatic ontology generation
- Sentiment analysis
- Question answering

Analytical Data Base for Energy sector

- Industrial indicators analysis and report generation
- Machine based investing advising
- Oil company ranking system
- News sentiment analysis, keyword extraction and trend analysis
- Query focused multi-document text summarization

GreedSum: Automatic Text Summarizer

- GreedSum is an automatic text summarization algorithm described in I. Akhmetov, A. Gelbukh and R. Mussabayev, "Greedy Optimization Method for Extractive Summarization of Scientific Articles," in IEEE Access, vol. 9, pp. 168141-168153, 2021, doi: 10.1109/ACCESS.2021.3136302. To summarize a text, we utilize a greedy algorithm that selects the most important sentences which are ranked according to their value in content of the most important words in the text. It also cares to avoid redundancy, looking not to include sentences that are very similar to the ones already selected.
- Products
  - Web app: web realization of the algorithm on the Pythonanywhere platform.
  - Chrome extension: allows you to summarize webpages, highlighting the most important sentences in the text and allowing you to save the extract.
  - Word Extension: this one is yet to come...

Thank you for your attention!

- Contacts:

Iskander Akhmetov

Data Science laboratory coordinator, Assistant professor

School of Information Technology and Engineering

Kazakh-British Technical University

i.akhmetov@kbtu.kz

+7 (701) 909 98 91