Exploring structural characteristics of Sheptytsky's pastoral letters through ML algorithms

Olena Levchenko[†] and Yurii Hulyk[*,†]
Lviv Polytechnic National University, S. Bandera Street, 12, Lviv, 79013, Ukraine

Abstract

The research work on analysing the structural features of Andrey Sheptytsky's pastoral letters using machine learning algorithms is a step towards understanding this outstanding spiritual leader. Andrei Sheptytsky was an archbishop and hegumen of the Greek Catholic Church. He was active during the First World War and the struggle for Ukrainian independence. One of the key tasks of the study is to understand the peculiarities of the structure of pastoral letters written by Andrei Sheptytsky. These letters contain important information about his thoughts, views, and influence on parishes and believers. Machine learning algorithms allow us to systematically analyse and identify the structural features of these letters. Machine learning algorithms, such as neural networks and deep learning models, can be applied to automatically analyse text. They can classify different aspects of the text, such as headlines, keywords, topics, and tone. Using these algorithms, we can modify the typical structural features of Andrei Sheptytsky's pastoral letters, such as their organisation, writing style, use of biblical references, and other elements that help to create a complete picture of his intellectual and spiritual guidance. The importance of this study lies in the fact that by analysing the structural features of the pastoral letters, we can understand Sheptytsky as a figure, his views on religious issues, political leadership, and social problems. This is promising for further understanding of Sheptytsky's influence on Ukrainian society. This study can also be useful for students, historians, religious scholars, and anyone interested in Ukrainian history and Sheptytsky as an important figure in the country's history. Understanding the structural features of his pastoral letters opens up new possibilities for analysing his activities and significance for the Ukrainian people. In conclusion, the study of the structural features of Andrey Sheptytsky's pastoral letters using machine learning algorithms is a factor in understanding his role as a spiritual leader and his contribution to the history of Ukraine. The data obtained from this study can be used for further study and analysis of Sheptytsky and his impact on society. This article explores the methods of text analysis based on clustering and thematic modelling. This method helps to identify hidden rules in the text and better understand their structure. The article also examines how the structural features of the pastoral letters are related to their historical context, as the structure of the letters changed over time, reflecting changes in social and political conditions.

Keywords

Pastoral letters, machine learning, clustering, thematic modelling, religious rhetoric, authorship.

1. Introduction

Andrey Sheptytsky, a prominent figure in the Ukrainian Greek Catholic Church, left a profound legacy through his pastoral letters, which remain significant both spiritually and historically.

CLW-2024: Computational Linguistics Workshop at 8th International Conference on Computational Linguistics and Intelligent Systems (CoLInS-2024), April 12–13, 2024, Lviv, Ukraine
* Corresponding author.
† These authors contributed equally.
olena.p.levchenko@lpnu.ua (O. Levchenko); yurii.v.hulyk@lpnu.ua (Y. Hulyk)
0000-0002-7395-3772 (O. Levchenko); 0009-0000-8030-1409 (Y. Hulyk)

These letters, penned during a tumultuous period in Eastern European history, offer valuable insights into the theological, cultural, and social issues of his time. While traditional methods of literary analysis have provided a solid foundation for understanding Sheptytsky's works, the advent of machine learning (ML) technologies opens new avenues for exploring their structural characteristics in a more nuanced and comprehensive manner.

This study aims to delve into the intricate patterns and stylistic features of Sheptytsky's pastoral letters by leveraging advanced ML algorithms. By applying natural language processing (NLP) techniques, we can uncover latent structures, thematic distributions, and rhetorical strategies that may not be immediately apparent through conventional analysis. The integration of ML not only enhances our understanding of Sheptytsky's epistolary style but also sets a precedent for the application of digital humanities in the study of religious and historical texts. Through this exploration, we aim to demonstrate how ML algorithms can enrich our appreciation of Sheptytsky's contributions and provide a template for future research in similar domains.

2. Forming a database for researching pastoral letters using AI

1.1. Description of the data collection process

The use of artificial intelligence can greatly facilitate the process of forming a pastoral letter research database by helping to collect, organise and analyse a large amount of information from these letters [9].

First of all, the use of AI allows for the automatic collection of pastoral letters from various sources such as websites, blogs, social media, etc. With the help of text processing algorithms, AI can quickly extract keywords, topics, and other important details from each letter.

Once the emails are collected, AI can help create a structured database where each email has its own place. It can easily find, sort, and filter emails by all parameters, such as data, author, subject, etc [6].

In addition, using AI machine learning algorithms, it can learn to recognise certain trends in pastoral emails. For example, it can detect certain topic groups that are important, or detect changes in the approaches or samples of letter writers [8].

1.2. Description of machine learning algorithms used for:

1.2.1. Automated categorisation of pastoral letters by author and date

The following machine learning algorithms can be used for automated categorisation of pastoral letters by author and date [5]:

1. Classification algorithm: this algorithm can be trained to recognise the authors of pastoral letters based on their content. The input to this algorithm can include the content of the letters and information about the author. Once trained, the algorithm will be able to classify new letters by author. For instance, the algorithm might analyze linguistic patterns, vocabulary usage, and writing style to discern unique characteristics associated with each author. By providing information about the author, such as past writing samples or biographical data, the algorithm can further refine its classification accuracy. This algorithm can be particularly useful in cases where the authorship of a letter is ambiguous or disputed, providing a data-driven approach to resolve such issues.

2. Clustering algorithm: This algorithm will help to group pastoral letters based on similarity of content. The input to this algorithm can include the content of the emails. The algorithm will divide the emails into groups, with each group having the following content. This allows you to quickly find emails from a particular author. The clustering algorithm employs techniques like k-means clustering or hierarchical clustering to organize the pastoral letters into distinct groups based on shared characteristics. By examining the content of the emails, the algorithm identifies patterns and similarities, grouping together letters with comparable themes, topics, or language usage. This functionality can facilitate efficient archival and retrieval of emails, enabling users to access related correspondence swiftly and conveniently.
3. Regression algorithm: This algorithm can learn to predict the dates of pastoral letters based on various factors, such as words or phrases that appear in the letter or information about the author. The input to this algorithm can include the content of the letters and information about the author. After training, the algorithm can convert the dates of any emails. Through regression analysis, the algorithm identifies correlations between textual features, author attributes, and the dates of pastoral letters. By analyzing historical data, the algorithm learns patterns and trends associated with the temporal aspects of letter writing. This predictive capability allows users to estimate the dates of previously undated letters or anticipate the timing of future correspondence. Moreover, the algorithm's ability to incorporate diverse factors enhances its accuracy and robustness, enabling nuanced predictions even in complex scenarios where multiple variables influence the dating of pastoral letters.

Overall, using these machine learning algorithms helps automate the classification of pastoral letters by author and date, saving time and making the process more efficient.

1.2.2. Identification and categorisation of structural elements

All of the machine learning algorithms that follow can be trained using a dataset consisting of pastoral letters with known structural elements and their classification or labelling.

Data preprocessing techniques such as feature extraction can be used to improve the results. Various machine learning algorithms can be used to detect and categorise structural elements in pastoral letters [2]: object algorithm, classification algorithm, semantic analysis algorithm, change detection algorithm.

The object algorithm learns to recognise and detect various structural elements such as headings, subheadings, paragraphs, lists, tables, etc. It uses text processing and image processing techniques to detect such objects based on their shape, size, or other features. By covering both text and image processing, this algorithm can effectively parse through diverse document formats, whether they are scanned images, PDFs, or plain text files. It employs advanced pattern recognition methods to identify structural elements within documents, allowing for comprehensive analysis and manipulation. Furthermore, the algorithm's ability to consider features like shape and size enhances its adaptability to different document layouts and formats, ensuring robust performance across a variety of scenarios.

Classification algorithm learns to recognise and categorise different types of structural elements in pastoral lists. For example, it can classify headings, subheadings, paragraphs, lists, etc. The input to this algorithm can include textual descriptions of the structural elements and their location on the page. Through machine learning techniques, the

classification algorithm distinguishes between various structural components within pastoral letters, assigning them to predefined categories or classes. By analyzing textual descriptions and spatial information, such as coordinates or bounding boxes, the algorithm accurately labels each element according to its type. This capability facilitates automated document processing and organization, enabling efficient retrieval and manipulation of specific content elements within pastoral letters.

Semantic analysis algorithm learns to understand the semantic relationships between the more structural elements in the pastoral letters. It analyses textual structural elements and identifies relationships, such as heading-subheading or table-heading relationships. This helps you understand the structure of the letter and identify key elements. Through semantic analysis, this algorithm goes beyond mere structural identification to interpret the meaning and context of textual elements within pastoral letters. By examining the relationships between headings, subheadings, and other components, the algorithm discerns hierarchical structures and thematic connections inherent in the document. This deeper understanding enhances comprehension and facilitates tasks such as information extraction and summarization, enabling users to extract meaningful insights from the content of pastoral letters more effectively.

Change detection algorithm is used to detect and track changes in structural elements in pastoral lists. It compares two versions of the email and identifies elements that have been added, deleted, or changed. This is useful for detecting changes in the structure of an email, for example, when it is updated or edited. The change detection algorithm employs algorithms like diffing or tree differencing to pinpoint alterations between different versions of pastoral letters. By comparing structural elements across documents, the algorithm identifies modifications such as additions, deletions, or revisions. This functionality is invaluable for version control and document management, enabling users to track the evolution of pastoral letters over time and ensuring the integrity and consistency of the document structure, particularly in collaborative or dynamic editing environments.

3. Description of Andrey Sheptytsky's letters and their significance

Metropolitan Sheptytsky went down in the history of Ukrainian state-building as a truly remarkable and outstanding figure, capable of self-sacrifice, heroism and wandering for the happiness of his homeland and his people. While still Bishop Stanislav, he clearly outlined the tasks and priorities that the Ukrainian people should pursue. In particular, in his pastoral letter to the faithful "Christian Deeds" (2 August 1899), Sheptytsky advised teachers on how to educate conscious and patriotic youth who work in the interests of the fatherland: "Turn your desires into the basis of national wealth and power. Let small children learn to love their land, let children learn to work on the land, and let future generations take trade and commerce into their own hands, because a person without a society in which he trades is always weak; a society in which foreigners trade is poor."

One of the most famous pastoral letters of Andrei Sheptytsky is the Patriotic Letter, written in 1914 during the First World War. In this letter, the archbishop calls on Ukrainians to become patriots and unite to defend their rights and interests. He stressed the importance of national unity and support for Ukrainian soldiers in the war. This pastoral letter was extremely important for strengthening the Ukrainian national movement and national consciousness. Another well-known pastoral letter by Andrei Sheptytsky, "A Letter of Praise on the Ukrainian Language Question," was written in 1916. In this letter, the archbishop called for the protection, development and respect of the Ukrainian language.

He emphasised that language is an integral part of Ukrainian national identity and is important for transmitting values and history. This letter had a significant impact on the development of the Ukrainian language and culture, especially in support of the literary heritage. The next famous pastoral letter, the Letter on Unemployment and Poverty, was written in 1933 during the Great Famine. In this letter, St Andrew asks for help for the starving, expressing his sympathy and outrage at this terrible tragedy.

He called on the Christian community to help the victims and on the international community to provide international assistance. This pastoral letter became a symbol of moral protest against the genocide of the Ukrainian people. Andrey Sheptytsky's pastoral letters are of great social, cultural and religious significance in Ukraine. They contributed to the development of the Greek Catholic Church in Ukraine, strengthened national identity and supported the Ukrainian people in the most important historical moments. Andrey Sheptytsky left an unforgettable legacy for future generations with his pastoral letters, which are considered one of the most important religious texts in Ukrainian history [1].

4. Study of the structural features of letters

4.1. Identification of typical structural elements

Machine learning can be a useful tool for identifying common structural elements in pastoral letters. Machine learning algorithms can be trained on a dataset of pastoral letters with annotated structural elements.

Before starting to identify structural elements, it is necessary to build a machine learning model that can recognise different structural elements, such as headings, paragraphs, etc. In order to apply machine learning to the analysis of Andrey Sheptytsky's pastoral letters, it is necessary to prepare an appropriate data set. This dataset can include text files containing the pastoral letters, as well as labels indicating the typical structural elements of each letter developed by experts in a specialised field. In this model, the process analyses each pastoral letter and identifies its structural elements, such as headings, paragraphs, lists, etc.

Algorithms based on neural networks and statistical methods can scan and analyse large amounts of text, search for characteristic keywords and phrases, establish connections between them, and classify sentences according to their structure.

The results of a machine learning model allow us to draw several interesting conclusions. First, it is possible to use the structural elements used in Andrey Sheptytsky's pastoral letters, which allows us to understand his creative approach to writing letters. Secondly, it is possible to compare the structural elements used in different pastoral letters, which addresses changes in the style and content of the message over the years. The use of machine learning in such analyses opens up new possibilities for researching texts and revealing their internal structure [3].

4.2. Algorithm for analysing pastoral letters

One of the possible algorithms that can be applied to the analysis of Sheptytsky's pastoral letters is the "bag of words" - a model of natural language texts in which each document or text appears as an unordered set of words without information about the relationships between them. It can be represented in the form of a matrix, with each row representing a separate document and each column representing a separate unique word. The intersection of a row and a column contains the number of occurrences of the word in the corresponding document [11].

Alternatively, TF-IDF is a statistical measure used to assess how important a word is to a document in a collection or corpus. The importance increases in proportion to the number of times the word appears in the document but is offset by the frequency of the word in the corpus. Term Frequency (TF): This is a count of the frequency of a word in the current document. Since each document is different in length, it is possible that a term will appear more often in long documents than in short documents. Therefore, the frequency of a term is divided by the length of the document for normalisation where n_i is the number of times word t appears in the document, and the denominator is the number of all words in the document.

The Skip-Gram model takes a word as input and predicts the probability of each word in the dictionary to be adjacent to this input word. In other words, the Skip-Gram model predicts the context for the input word. To train a neural network, you need to represent words in numerical form. For this purpose, "one-hot-encoding" vectors are used, where the input word has a "1" in the position of the input word and "0" in the other positions. Thus, a one-hot vector is fed into the neural network, and the output is a vector of the dimension of the input vector, which contains the probabilities that each word from the dictionary will occur near the input word.

This neural network has one hidden layer. The input vector has dimension 1xV, where V is the number of words in the dictionary. The dimension of the hidden layer is VxE, where E is the hyperparameter responsible for the size of the vector representation of the word. The output from the hidden layer has a dimension of 1xE and is fed to the softmax layer. The dimension of the output layer is 1xV, where each value in the vector is an estimate of the probability of the target word at that position. After training, we get a vector representation of the word by multiplying the input vector by the weights of the hidden layer.

These algorithms can analyse Sheptytsky's letters and classify them according to their positive, negative, or objective sentiment. The results of the analysis of Sheptytsky's pastoral letters can be useful not only for studying Sheptytsky's spiritual thought, but also for understanding his reaction to contemporary problems and trends. For example, artificial intelligence can detect changes in sentiment between popular periods or connections between certain topics and emotional reactions through lexical context analysis. Obviously, the analysis and sentiment of Andrei Sheptytsky's pastoral letters can be of great importance for understanding the spiritual and social transformations he led [10].

4.3. Practical use of the bag of words algorithm with Phyton

An example for studying this algorithm is the letter «Повинуйтеся своїй духовній владі» [7]:

«А тепер: послідна моя рада й послідне прохання до Вас:

Держіться своїх Духовних Отців! Слухайте і любіть їх! І ніколи не дайтеся звести людям, які схотіли б Вас від св. Церкви відірвати.

Памятайте, що Бог не є вітцем того чоловіка, якого матірю не є св. Церква. Тому будьте вірними дітьми св. кат. Церкви!

До своєї смерти заховайте вірність Святішому Вітцеві Папі Римському! Будьте вірними та повинуйтеся у всьому теж й епископові, якого Вам дам!

Хто небудь ним зістане, – то будьте певні, що даний він Вам буде за пастиря лишень з огляду на Ваше добро. Приймітьже його з любовю та довірям, так, як коли приймали б Ви самого Христа – Спасителя.»

The code is used to create the algorithm:

```
from collections import Counter
def bag_of_words(text):
words = text.split()
word_counts = Counter(words)
return word_counts
text = "This is a simple example of bag of words model"
bow_model = bag_of_words(text)
print(bow_model)
```

Let's analyse this code step by step [12]:

1. from collections import Counter: This line imports the Counter class from the collections module. Counter is used to count the number of occurrences of elements in a sequence (in this case, words in a text).
2. def bag_of_words(text): This is the definition of the bag_of_words function, which takes one argument - text.
3. words = text.split(): This line splits the text into words using the split() method. By default, split() splits a string of whitespace and returns a list of words.
4. word_counts = Counter(words): This line creates a Counter object that counts the number of occurrences of each word in the word list.
5. return word_counts: The function returns a Counter object containing the counted number of times each word appears in the text.
6. text = "This is a simple example of bag of words model "bag_of_words: This is just a string of text on which we can test the function
7. bow_model = bag_of_words(text): This line calls the bag_of_words function, passing it the string text. The result of the function (the Counter object) is stored in the bow_model variable.
8. print(bow_model): The result is a Counter object containing the number of occurrences of each word in the text string.

This code splits the text into individual words, counts the number of occurrences of each word using "Counter", and then returns a dictionary where the keys are words and the values are the number of times they occur in the text. Figure 1 shows the code for the "bag of words" algorithm, and Figure 2 shows the result of the counted words.

4.4. Assessment of the quality of algorithms

Based on the image, we have the following numbers:

- The number "1" occurs 81 times.
- The number "2" occurs 14 times.
- The number "3" occurs 2 times.

```
з  колекцій  імпорт  Лічильник

def  create_bag_of_words ( текст ) :

    слова  =  текст . розділити ( )
    bag_of_words  =  Лічильник ( слів )
    повернути  bag_of_word

text  =  "А тепер: дослідна моя рада й слідне прохання
    до Вас: Держіться своїх Духовних Отців! Слухайте і
    любіть їх! Я ніколи не давайте звести людей, які
    шотіліб Вас від св. Церкви відірвати.Пам'ятайте, що
    Бог не є вітцем того чоловіка, якого матірю не є св
    . Церква. Тому будьте вірними дітьми св. кат.
    Церкви !До своєї смерті захищайте вірність
    Святішому Вітцеві Папі Римському! Будьте вірними та
    повинуйтеся у всьому теж й єпископові, якого Вам
    дам!Хто небудь ним зістане, - то будьте певні, що
    даний він Вам буде за пастиря лишень з огляду на
    Ваше добро. Приймітьже його з любов'ю та довірям,
    так, як коли приймалиб Ви самого Христа - Спасителя
    . Бож він його слугою і заступником..."
bag_of_words  =  create_bag_of_words ( текст )
надрукувати ( bag_of_words )
```

Figure 1: "Bag of words" algorithm.

```
Counter({'не': 3, 'св.': 3, 'й': 2, 'і': 2, 'Церкви': 2, 'що': 2, 'є': 2, 'якого': 2, 'будьте': 2, 'вірними': 2, 'та': 2,
'Вам': 2, '-': 2, 'він': 2, 'з': 2, 'його': 2, 'А': 1, 'тепер:': 1, 'послідна': 1, 'моя': 1, 'рада': 1, 'послідне': 1,
'прохання': 1, 'до': 1, 'Вас:Держіться': 1, 'своїх': 1, 'Духовних': 1, 'Отців!': 1, 'Слухайте': 1, 'любіть': 1, 'їх!
1, 'І': 1, 'ніколи': 1, 'дайтеся': 1, 'звести': 1, 'людям,': 1, 'які': 1, 'схотілиб': 1, 'Вас': 1, 'від': 1,
'відірвати.Памятайте,': 1, 'Бог': 1, 'вітцем': 1, 'того': 1, 'чоловіка,': 1, 'матірю': 1, 'Церква.': 1, 'Тому': 1,
'дітьми': 1, 'кат.': 1, '!До': 1, 'своєї': 1, 'смерти': 1, 'заховайте': 1, 'вірність': 1, 'Святішому': 1, 'Вітцеві': 1,
'Папі': 1, 'Римському!': 1, 'Будьте': 1, 'повинуйтеся': 1, 'у': 1, 'всьому': 1, 'теж': 1, 'епископові,': 1, 'дам!
Хто': 1, 'небудь': 1, 'ним': 1, 'зістане,': 1, 'то': 1, 'певні,': 1, 'даний': 1, 'буде': 1, 'за': 1, 'пастиря': 1,
'лишень': 1, 'огляду': 1, 'на': 1, 'Ваше': 1, 'добро.': 1, 'Приймітьже': 1, 'любовю': 1, 'довірям,': 1, 'так,': 1,
'як': 1, 'коли': 1, 'приймалиб': 1, 'Ви': 1, 'самого': 1, 'Христа': 1, 'Спасителя.': 1, 'Бож': 1, 'слугою': 1,
'заступником...': 1})
```

Figure 2: Result of the counted words.

Based on this, it can be noted that these numbers have the following values:

- TP (True Positives) = 3 (that is, objects that were correctly classified as positive occur 3 times)
- TN (True Negatives) = 2 (that is, objects that were correctly classified as negative occur twice)
- FP (False Positives) = 2 (that is, objects that were incorrectly classified as positive occur 2 times)
- FN (False Negatives) = 1 (that is, objects that were incorrectly classified as negative occur 1 time)

After developing and training models, the next step is to evaluate their performance. A typical measure of classifier quality is accuracy, which for a binary task can be easily obtained from the error matrix (Figure 3). This indicator can be calculated as follows:

$$Accuracy = TP + TN / P + N = (TP + TN) / (TP + FN + TN + FP) = 3+2 / 3+1+2+2 = 5 / 8 = 0,625 * 100\% = 62,5\ \% \quad (1)$$

where TP is a well-defined positive class; TN is a well-defined negative class; FP is a false positive class; FN is a false negative class. Thus, the accuracy of the model is 62.5%. This means that the model correctly classifies 62.5% of all cases. In other words, 62.5% of all model predictions are correct.

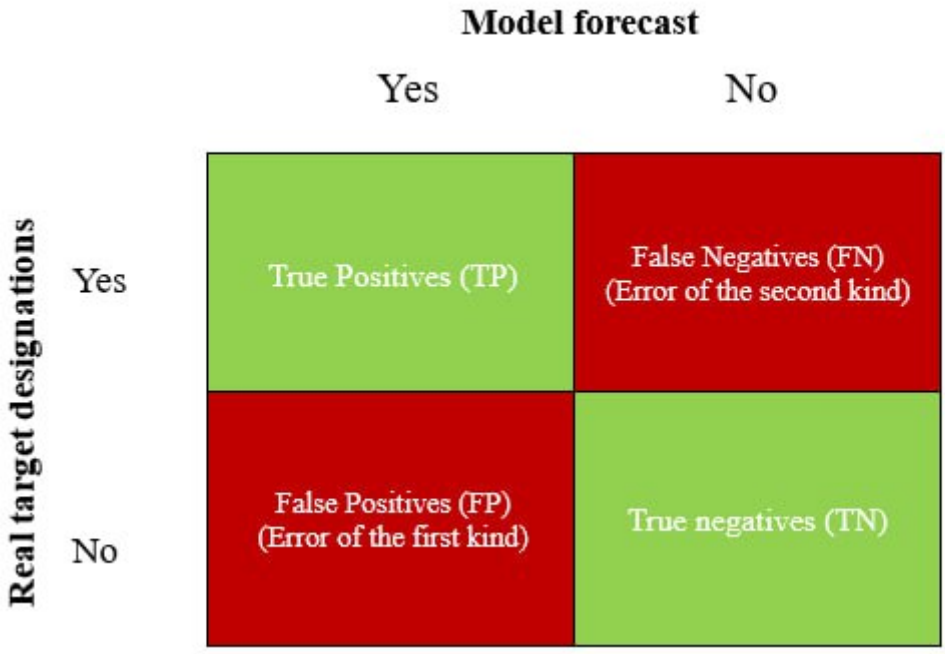

Figure 3: Error matrix.

Empirical evidence shows that accuracy is highly dependent on the balance of the data. Using this metric can lead to incorrect conclusions in the case of a strong imbalance of classes. For instance, suppose we have a dataset consisting of 90% negative instances and 10% positive instances. A classifier that always predicts negative will achieve 90% accuracy, which might seem high but is actually not helpful for identifying positive instances.

In the case of unbalanced data, it is important to check how well the classifier classifies only a part of the data, for example, positive or negative classes. To do this, you can use the sensitivity and recall metrics. Sensitivity, also known as true positive rate or recall, measures the proportion of actual positive cases that are correctly identified by the classifier. It is calculated using the formula:

$$Precision = TP / (TP + FP) = 3 / 3+2 = 3 / 5 = 0,6 * 100\% = 60\% \quad (2)$$

Precision = 60% means that of all positive class predictions, 60% are correct. In other words, if the model predicted 5 positive cases, only 3 of them turned out to be true positives (true positives) and 2 were false positives (false positives).

The completeness metric is used to avoid false negative classifications. It is calculated using the formula:

$$Recall = TP / P = TP / (TP + FN) = 3 / 3+1 = 3 / 4 = 0,75 * 100\% = 75\% \quad (3)$$

This means that the model correctly identifies 75% of all positive samples. In other words, out of all the cases where there should have been positive results (TP + FN = 4), the model correctly detected 3 of them (TP = 3), which is 75%.

Another indicator, which is the harmonic mean of the two previous estimates, is the F-measure. This is a general measure of model accuracy that combines sensitivity and completeness. In other words, a good F_1 means that you have a low number of false positives and false negatives. F_1 is considered perfect when it equals 1 and catastrophic when it equals 0 [13]:

$$F1 = 2 * Precision * Recall / Precision + Recall = 2 * 0,6 * 0,75 / 0,6 + 0,75 = 0,9 / 1,35 = = 0, 66 * 100\% = 66\% \quad (4)$$

So, the meaning $F1$-measure is approximately 66%. This means that the model has a fairly good balance between accuracy and completeness, indicating a relatively low number of both false positives and false negatives. While accuracy provides a general measure of classifier performance, it may not be suitable for assessing algorithms on imbalanced datasets. Sensitivity, completeness, and the F-measure offer more nuanced insights into the classifier's ability to correctly identify positive and negative instances, making them valuable metrics for evaluating algorithm performance in real-world applications.

5. Prospects opened by the use of AI for the study of Andrey Sheptytsky's letters

AI can help to decipher and analyze these letters, allowing us to better understand the thoughts, ideas, and views of this prominent figure. The main use of AI lies in its ability to automatically recognize and translate text. This is especially useful for letters written in other languages, as we can access important information without the need to know multiple languages. Additionally, AI can facilitate the comparison of Sheptytsky's pastoral letters across different time periods, enabling scholars to trace the evolution of his ideas and perspectives over time. By systematically analyzing changes in language use, thematic emphasis, and rhetorical strategies across various epochs, researchers can discern patterns of intellectual development and ideological shifts within Sheptytsky's corpus.

AI can also detect common themes, trends, or interesting phrases that will help in drawing a complete map of Andrey Sheptytsky's thoughts. Another aspect in which AI can be useful is analyzing the emotional tone of letters. Thanks to a special algorithm, AI can determine the emotional state of the author and highlight keywords or phrases that express certain emotions. This will allow us to gain a deeper understanding of Andrey Sheptytsky's feelings in different situations and explore his reaction to certain events. Furthermore, AI can compare Sheptytsky's writing style and statistical parameters to other religious texts of different authors, shedding light on his unique literary voice and theological contributions. Such comparative analyses can reveal Sheptytsky's distinct theological emphases, rhetorical strategies, and ideological affinities within the broader context of religious discourse.

Moreover, AI's capacity for linguistic analysis enables the identification and interpretation of biblical citations within Sheptytsky's letters. By employing a custom-trained ML model, AI can discern citations from the Bible embedded within the text, classify them according to their thematic relevance, and provide insights into Sheptytsky's theological interpretations and scriptural influences. This capability facilitates a nuanced understanding of Sheptytsky's theological methodology, illuminating the sources and contexts that informed his religious thought.

In addition, AI can analyze specific names, events, or places in Sheptytsky's letters, exploring their connections and providing specific suggestions for each of them. This capability allows us to see a broader picture of Sheptytsky as a person, to exploit his influence on society and his attitude to various issues. Furthermore, AI can facilitate interdisciplinary research by integrating textual analysis with other forms of data, such as historical records or sociopolitical context. By synthesizing diverse sources of information, AI enables a comprehensive examination of Sheptytsky's letters within their broader historical, cultural, and intellectual milieu.

6. Limitations and possible losses of the study

The study of the structural features of Andrey Sheptytsky's pastoral letters using machine learning algorithms may be subject to limitations due to the lack of relevant data or the limitations of processing large amounts of text. Losses may also occur due to problems with the accuracy of algorithms for detecting complex structural features that may be contained in the lists. Therefore, it is important to look for possible ways to overcome the limitations when conducting research [4].

Let's look at the possible limitations and losses in more detail.

Limitations:

- Insufficient accuracy: Machine learning algorithms may struggle to accurately classify texts due to the complexity and variability of language. For example, nuances in language, subtle shifts in tone, or ambiguous phrasing can pose challenges for algorithms, leading to misclassifications. For instance, a classification algorithm may misclassify a paragraph discussing theological doctrine as a standard narrative passage, impacting the accuracy of structural analysis.
- The need for a large amount of data: Machine learning algorithms require extensive training data to generalize effectively across diverse texts. In the case of Sheptytsky's pastoral letters, if the dataset is limited or lacks diversity, the algorithm may fail to capture the full range of structural features present in his correspondence. For example, a clustering algorithm trained on a small subset of Sheptytsky's letters may struggle to accurately group them based on thematic similarities or linguistic patterns.
- Difficulty of interpretation: The outputs of machine learning algorithms can be challenging to interpret, especially for non-experts. This difficulty arises from the complexity of the underlying algorithms and the abstract nature of their decision-making processes. For instance, if a classification algorithm categorizes a paragraph as a subheading without providing clear justification, researchers may find it challenging to understand the rationale behind the classification decision.
- Loss of context: Machine learning algorithms may lack the contextual understanding necessary to accurately analyze texts. Without considering the broader context in which Sheptytsky's letters were written—such as historical events, cultural norms, or personal circumstances—the algorithms may misinterpret or overlook significant structural features. For example, an algorithm may misclassify a passage discussing a specific historical event as a general narrative, failing to recognize its contextual significance within the letter.

Possible losses:

- Loss of information: Machine learning algorithms may overlook nuanced or subtle aspects of the text, resulting in a loss of valuable information. For example, if an algorithm focuses solely on identifying structural elements such as headings and paragraphs, it may fail to capture the underlying thematic or rhetorical nuances present in Sheptytsky's letters, diminishing the richness of the analysis.
- Loss of nuance: Machine learning algorithms may oversimplify complex textual features, leading to a loss of nuance in the analysis. For instance, if an algorithm categorizes all instances of biblical references as a single structural element, it may overlook variations in citation styles, theological interpretations, or contextual meanings present in Sheptytsky's letters, obscuring important nuances in his writing.
- Loss of the human element: Machine learning algorithms cannot account for the subjective or intuitive aspects of textual analysis, potentially overlooking the human elements inherent in Sheptytsky's letters. For example, if an algorithm fails to recognize the subtle shifts in tone or emotion conveyed through Sheptytsky's language, it may miss important

insights into his personality, beliefs, or motivations, diminishing the holistic understanding of his correspondence.

To overcome these limitations, several strategies can be employed. Firstly, utilizing a combination of algorithms can significantly enhance classification accuracy. By leveraging various machine learning algorithms in tandem, the strengths of each can complement and compensate for the weaknesses of others. For example, ensemble methods such as random forests or gradient boosting combine multiple weak classifiers to create a stronger overall model. By aggregating the predictions of individual algorithms, ensemble methods often outperform any single algorithm alone, providing more robust and reliable results.

Secondly, meticulous attention to parameter selection is crucial. Fine-tuning the parameters of machine learning algorithms ensures optimal performance and better adaptation to the specific characteristics of the data. This involves systematically exploring different parameter values through techniques like grid search or random search and selecting the combination that yields the best performance on a validation set. For instance, adjusting the regularization parameter in a logistic regression model or tuning the depth of a decision tree can significantly impact the model's performance and generalization ability.

Moreover, employing high-quality, representative data is essential. By using datasets that accurately reflect the underlying distribution and diversity of the target domain, machine learning models can be trained more effectively and produce more reliable results. This may involve collecting new data, augmenting existing datasets, or carefully curating samples to ensure balanced representation of different classes or categories. For example, in sentiment analysis tasks, ensuring a diverse range of opinions and sentiments in the training data helps the model generalize better to unseen instances and improves its ability to capture nuances in language use.

Furthermore, contextual considerations play a pivotal role in interpreting the outcomes of machine learning algorithms. Taking into account the context in which texts were authored aids in understanding and contextualizing the predictions and classifications made by the models. For instance, historical or cultural factors may influence the language, themes, and stylistic conventions present in Sheptytsky's letters. By considering these contextual nuances, researchers can better interpret algorithmic findings and draw more accurate conclusions about the structural features of the texts.

Lastly, involving experts in literary studies and theology can provide invaluable insights. Their domain expertise can offer nuanced interpretations of the results, particularly in contexts where cultural or historical contexts are pertinent to the analysis. Collaboration between machine learning practitioners and domain experts facilitates a more comprehensive and nuanced understanding of the data and its implications. For example, literary scholars can provide insights into the stylistic conventions and thematic motifs prevalent in Sheptytsky's letters, while theologians can offer interpretations of the religious and philosophical themes embedded within the texts. By integrating diverse perspectives and expertise, researchers can enrich their analysis and gain deeper insights into Sheptytsky's literary and theological contributions.

Conclusions

To sum up, the use of machine learning algorithms helps to identify and categorise structural elements in pastoral letters. The object algorithm allows you to recognise headings, subheadings, paragraphs, lists, tables, etc. The classification algorithm helps to categorise these

elements by their type. The semantic analysis algorithm understands the semantic relationships between elements, such as heading-subheading or table-heading relationships. A change detection algorithm helps to identify and track changes in structural elements. The use of these algorithms allows automating the process of identifying and categorising structural elements in pastoral letters, making the process more efficient and effective.

In the course of the study, we determined that one of the possible algorithms that can be applied to the analysis of Sheptytsky's pastoral letters is a "bag of words" - a model of natural language texts in which each document or text looks like an unordered set of words without information about the connections between them. It can be represented in the form of a matrix, where each row corresponds to a separate document and each column corresponds to a separate unique word. The intersection of a row and a column contains the number of occurrences of the word in the corresponding document.

Another possible algorithm is TF-IDF, which is used to estimate the importance of a word for a document in a collection or corpus. The importance increases in proportion to the number of times the word appears in the document but is offset by the frequency of the word in the corpus. Term Frequency (TF) is a calculation of the frequency of a word in the current document, and the term frequency is divided by the length of the document for normalisation.

The Skip-Gram model predicts the probability of each word in the dictionary to be adjacent to the input word. To train a neural network, words are represented in numerical form using "one-hot-encoding" vectors. The neural network has one hidden layer, and the output is a vector of the dimension of the input vector, which contains the probabilities that each word in the dictionary will occur near the input word.

These algorithms can be applied to the analysis of Sheptytsky's pastoral letters and classify them according to their positive, negative, or objective sentiment.

In a practical application of the Phyton-based "bag of words", one can see how the algorithm code breaks the text into individual words, counts the number of occurrences of each word using "Counter", and then returns a dictionary where the keys are the words and the values are the number of occurrences of the words in the text.

Considering the prospects and possible limitations and losses, we can note that among the prospects is the ability of AI to help analyse and decipher the pastoral letters of Andrei Sheptytsky, which allows us to better understand his thoughts and feelings. AI can automatically recognise and translate text, as well as identify common themes and emotional tone of letters. However, the study may be limited by insufficient data, difficulty in interpreting the results, and loss of context. To overcome these limitations, it is important to use a combination of algorithms, carefully select parameters, use high-quality data, and involve experts. Overall, the use of AI to research Andrey Sheptytsky's letters offers great potential for a deeper understanding of his personality and contribution to Ukrainian history and culture.

References

[1] Basarab V. I., State-building concept and national-patriotic ideas of Metropolitan Andrey Sheptytsky, Master's thesis, Uzhhorod National University, Uzhhorod, Ukraine, 2019.

[2] H. Zhu, L. Lei, “The research trends of text classification studies (2000–2020): a bibliometric analysis,” SAGE Open, vol. 12, no. 2, 2022, pp. 1–16.

[3] M. Agarwal and A. Saxena, “An overview of natural language processing,” International Journal for Research in Applied Science & Engineering Technology (IJRASET), vol. 7, no. 5, 2019, pp. 2811–2813.

[4] M. Eder, Mind your corpus: systematic errors in authorship attribution, in: Literary and Linguistic Computing, 28, 2013, pp. 603-614.

[5] M. Koppel, J. Schler, S. Argamon, Computational methods in authorship attribution, in: Journal of the American Society for Information Science and Technology, 60, 2009, pp. 9-26.

[6] Maerz, S. F., & Schneider, C. Q. . Comparing public communication in democracies and autocracies: Automated text analyses of speeches by heads of government. Quality & Quantity, 54, 2020, pp. 517–545.

[7] Metropolitan A. Sheptytsky, Works of Metropolitan Andrey Sheptytsky. Pastoral Letters to the Clergy and Faithful of the Stanislaviv Eparchy (1899-1904), Lviv - 1935, pp. 228.

[8] P. Sisyak, "Artificial Intelligence - Revolution, Hope or Utopia?", 2016. URL: https://www.imena.ua/blog/ai-revolution/.

[9] P. Yuola, Attribution of authorship, in: Fundamentals and trends in information retrieval, 1, 2006, pp. 233-334 .

[10] Rudzevych A.-M. P., Machine learning methods in sentiment analysis of textual information, Master's thesis, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine, 2020.

[11] Welbers, K., van Atteveldt, W., Kleinnijenhuis, J., Ruigrok, N., & Schaper, J.. News selection criteria in the digital age: Professional norms versus online audience metrics. Journalism, 17(8), 2016, pp. 1037–1053.

[12] Y. Zhang, J. Rong and Z. Zhi-Hua, “Understanding bag-of-words model: A statistical frame-work,” International Journal of Machine Learning and Cybernetics, vol. 1, no. 1, 2010, pp. 43–52.

[13] Yavary, A., Sajedi, H., & Abadeh, M. S. . Information verification improvement by textual entailment methods. SN Applied Sciences, 1, 2019, pp. 1048.