MATHEMATICAL MODELING OF THE LEXICAL-SEMANTIC FIELD OF ANDREY SHEPTYTSKYI'S PASTORAL LETTERS USING MACHINE LEARNING ALGORITHMS

Yurii Hulyk
Postgraduate Student, Lviv Polytechnic National University, Ukraine
e-mail: yurii.v.hulyk@lpnu.ua, orcid.org/0009-0000-8030-1409

Summary

This article explores the lexical and semantic field of Andrey Sheptytsky's pastoral letters using machine learning algorithms. To study the lexical and semantic field, such algorithms as Word2Vec, TF-IDF, CBOW, Bag-of-Words and Skip-Gram are used. The texts of this prominent figure are studied through the prism of modern methods of natural language processing, which allows for a more detailed identification of the peculiarities of their vocabulary and semantics. The article discusses the process of creating computer models for text analysis, as well as the use of machine learning algorithms for automatic processing and classification of lexical items in pastoral letters. The results obtained allow us to better understand the specifics of the author's language and mentality, as well as to identify patterns in the use of relevant words and expressions in his work. This work opens up new possibilities for the study of pastoral texts and deepening our understanding of their semantic properties.

Key words: pastoral letters, lexical and semantic field, machine learning, Word2Vec, text processing, NLP, TF-IDF, CBOW, Bag-of-Words, Skip-Gram.

DOI https://doi.org/10.23856/6204

1. Introduction

Pastoral letters reflect an important aspect of the spiritual, social and cultural heritage that is of significance to both religious communities and society as a whole. A special place among them is occupied by Andrey Sheptytsky's letters, which are notable not only for their high moral authority but also for their relevance and depth of wisdom.

In the world of modern research in the field of linguistics and natural language processing, there is a great need to use mathematical models and machine learning algorithms to analyse texts and identify their lexical and semantic features. However, to date, research on mathematical modelling of pastoral letters remains limited.

This article aims to fill this gap by applying machine learning algorithms to the analysis of Andrey Sheptytsky's pastoral letters. We aim to use modern methods of computational linguistics to reveal the internal structure and semantic properties of these texts. The presentation of mathematical models of the lexical and semantic field will help not only to better understand the content and context of the pastoral letters, but also to reveal their significance in shaping cultural discourse and moral and ethical values.

2. Text pre-processing methods

Text preprocessing is the task of transforming raw data into well-defined knowledge. The main operations of text pre-processing can be divided into the following stages:

Text preprocessing.

– **Tokenisation:** the process of breaking down a stream of textual content into words, terms, symbols, or other meaningful elements called tokens. Filtering (stop words) removes unnecessary information, including prepositions, articles, conjunctions, etc.

– **Lemmatisation process:** a process similar to root formation, but it determines the dictionary form of a word. This technique is used to reduce the length of words in a text. In text analysis techniques, the pre-processing stage plays an important role in converting root words into correct words for proper text analysis.

– The **process of identifying the roots** of certain words determines the origin of the word. Two main types of sources are used: inflectional and derivational. The most common algorithm for determining the origin of a word is Porter's algorithm.

Text transformation process

– **Text** transformation uses bag-of-words or vector spatial models. It performs the task of feature selection. At the same time, it reduces the dimensionality by removing redundant and irrelevant features. Sequences of words that occur frequently but have no meaning or significance in the collection of text documents are also removed. IAT has the following process.

– **Text clustering process:** measures similarity and groups similar texts.

– **Text abstraction (summarisation**) process: summarises the entire text of a document to its essence.

– **Text classification** process: automatically assigns multiple documents to different categories. It is a supervised learning method that classifies new documents based on input and output instances. The main goal of text classification is to automatically train classifiers based on supervised and unsupervised categories. For this purpose, statistical classification methods can be used, such as naive Bayesian classifier and nearest neighbour classifier, decision tree and support vector method *(Mogylna, 2022)*.

– To reduce the dimensionality of words, they are transformed into a **word form by** discarding endings, prefixes and suffixes. This way, the features are generalised and easier to classify.

– To transform a word into a word form, either **lemmatisation or stemming** is used. Lemmatisation takes into account the morphology of the language, so it accurately determines the word stem. First, the part of speech is determined, and then rules are applied to cut off endings and suffixes depending on the part of speech. Stemming, on the other hand, does not require dictionaries, but does not guarantee a match with the true morphological basis of the word. This algorithm cuts off the beginning or end of a word using a list of prefixes and suffixes *(Malyha, 2022)*.

3. Study of statistical methods of analysing pastoral letters

Analysis of the frequency of words. In the pastoral letters of Andrei Sheptytsky, certain tendencies can be identified in terms of emphasis on certain aspects of the Christian faith and moral values. Some of the key words he uses indicate that he favours concepts that focus

on repentance, love of neighbour, mercy, and peace. Terms related to Christian theology, such as "grace," "salvation," and "repentance," can be found with high frequency in Sheptytsky's pastoral letters.

An analysis of the frequency of words in pastoral texts can provide valuable information about the main thematic emphases, key concepts and ideas contained therein. To conduct such an analysis, you must first divide the text into individual words (tokens) and then count the number of occurrences of each word. The most frequently occurring words may indicate the main themes or priorities of the author. You can also identify unique or specific terms that may be key to understanding the context of the text. By analysing word frequency, we can get a better idea of the structure and content of a text, as well as the important themes that run through it.

The main systems that solve a similar task are:

– Sztergak framework;

– The Humb system;

– Wingnus system;

– KP-Miner system.

These systems are designed to process text, remove unnecessary elements (punctuation, unimportant words) and determine a set of features that characterise keywords *(Aggarwal, 2012)*.

The effectiveness of the considered systems for the top 10 key phrases.

The use of statistical methods of analysis ensures high reliability of the results of the study of linguistic texts, clarifies the conclusions and reveals the relationships between their properties. These methods help to obtain objective data necessary for the organisation of linguistic observations and ensure the reliability of the conclusions.

Table 1

Comparison of the systems for intelligent text analysis

Command	Accuracy, %	Fullness, %	F-measurement, %
Sztergak	37.80	25.78	30.65
Humb	32.00	21.83	25.95
Wingnus	30.50	20.80	24.73
KP-Miner	28.60	19.51	23.20

In analysing Andrey Sheptytsky's pastoral letters, statistical methods will allow us to use concepts such as the mean, which is used in probability theory to determine the central value of a discrete set of numbers. The formula for calculating the mean is the sum of the values divided by their number and is used to identify the key characteristics of linguistic units in the pastoral letters.

The study of the statistical profile of Andrey Sheptytsky's pastoral letters allows us to use mathematical methods of analysis to understand the linguistic features and structure of the texts.

One of the most important parameters for characterising a sample mean is the standard error of deviation. This value indicates the dispersion of sample means around the overall mean. The formula for calculating the standard error of deviation includes the parameters mentioned in the previous paragraph about the measure of fluctuation in the mean frequency, such as σ – standard deviation, – number of occurrences of the variant in the trials, i – variant number, and – mean.

The analysis of the standard error of deviation is important in the study of Andrey Sheptytsky's pastoral letters, since this measure allows us to estimate the spread of the distribution of sample means around the overall mean in the texts, which contributes to a better understanding of their structure and features.

Thus, each element that was studied was first evaluated by its absolute frequency, and then a variation series was formed. For these series of variations, average values were calculated and standard formulas were used to determine statistical estimates of the average frequency.

4. Vectorisation

Vectorisation is a classical approach to transforming input data from its original format (e.g. text) into vectors of real numbers that can be intelligent machine learning models. This approach is used in NLP and is not new, as it has been used since the creation of computers.

In machine learning, vectorisation is a step in feature extraction. The idea is to extract certain characteristics of text to train a model by transforming text into numerical vectors.

There are many vectorisation techniques that we will soon look at, ranging from simple implementations using binary features of term frequency to more complex techniques that take into account context. Depending on the specific situation and model, any of these techniques can perform the required tasks:

Bag-of-words. The simplest of all existing techniques. It includes three operations:

– **Tokenisation:** first, the input text is split into tokens. The sentence is represented as a list of its constituent words, and this is done for all input sentences.

– **Vocabulary creation**: only unique words are selected from all the resulting tokenised words, which are then sorted in alphabetical order.

– **Vector** creation: finally, a sparse matrix is created from the word frequencies of the resulting dictionary as input. In this sparse matrix, each row represents a sentence vector whose length (number of columns in the matrix) is equal to the size of the dictionary.

TF-IDF (Term Frequency-Inverse Document Frequency) is a numerical statistic that reflects the importance of a word in a document. Although this methodology is also based on frequency, like the word bag, it uses more complex calculations.

How is TF-IDF better than word bag?

In the "bag of words" method, we saw how the vectorisation was reduced to the frequency of words from the dictionary for a given document. As a result, articles, prepositions, and conjunctions, which are not as important to the overall meaning of a sentence, are given the same weight as, for example, adjectives.

TF-IDF solves this problem, so that words that are repeated too often do not drown out other less frequent but important words.

The algorithm consists of two parts:

– TF stands for Term Frequency, which is a normalised frequency index. It is calculated using the following formula:

– TF = Frequency of word in document / Total number of words in the document, i.e. TF = Frequency of term in the document / Total number of words in the document. Thus, you can assume that this number will always be ≤ 1 and with this in mind, you can already estimate how often a word occurs in the context of all the words in the document *(Abhishek, 2023)*.

Word2Vec uses a simple neural network with one hidden layer to train the weights. Unlike most other machine learning models, we are not interested in the prediction of this

neural network, but rather in the weights of the hidden layer, which we will train. The input vector multiplied by these weights is a vector representation of the word. Two algorithms are used to create Word2Vec representations: **Skip-Gram** and **CBOW (Continuous Bag-of-Words)**. Let's take a closer look at these algorithms:

1. The Skip-Gram model receives a word as input and predicts the probability of each word in the dictionary to be adjacent to the 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. To do this, 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 input vector's dimension, which contains the probability for each word in the dictionary to be adjacent to the input word. The neural network has one hidden layer. The dimension of the input vector is 1xV, where V is the number of words in the dictionary. The dimension of the hidden layer is VxE, where E is a 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 into 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 can obtain a vector representation of the word by multiplying the input vector by the weights of the hidden layer.

2. The Continuous Bag-of-Words model differs from the Skip-Gram model. It predicts the probability of each word appearing in the dictionary, taking into account the context of the words. The sizes of the hidden and output layers remain the same, but the dimensionality of the input layer and the calculation of the hidden layer activation functions change. If we have 4 context words for one target word, we have 4 1xV input vectors that are multiplied by the VxE hidden layer to get 1xE vectors. These vectors are averaged to get the final activation, which is fed into the softmax layer.

Support vector machine is used to find the hyperplane that best divides a given sample into two classes. This method can be applied to both binary classification and multiclass classification using one-vs-all and one-vs-one strategies.

In this method, we have a sample of elements $x_i \in R_n$ that have assigned classes $y_i \in \{-1,1\}$. The sample objects are represented by points. Support vectors are the data points that are closest to the hyperplane. They are critical elements of the dataset, as their removal changes the position of the hyperplane.

In a simple binary classification problem, when the sample is linearly separated, the hyperplane can be represented as a line that separates the two classes. The further the data is from the hyperplane, the more accurately it is classified.

The best hyperplane is the one with the maximum distance $1/\|w\|$ from each class to the other. Here, w is the normal vector to the separating hyperplane, which can be written as a set of points x satisfying the equation:

$$wx - b = 0,$$

where b is an auxiliary parameter. If the training set is linearly separable, you can choose two parallel hyperplanes so that they divide this set into two classes. The area between them is called the gap, the margin. These planes are described by the equations:

$$wx - b = 1,$$
$$wx - b = -1,$$
$$\|w\|^2 \rightarrow min$$
$$y_i (wx_i - b) \geq 1, \text{ для } 1 \leq I \leq n$$

By minimising the distance $\|w\|$ and at the same time excluding data from falling into the gap, we obtain a minimisation problem:

Such a problem is considered equivalent to finding the saddle point of the Langrange function, and is reduced to a quadratic programming problem where only binary variables λ_i are present. Having solved this problem, it is possible to express w and b by the following formulas, respectively:

$$w = \sum {}^{n}{}_{i=1} \lambda_i c_i x_i,$$
$$b = w \cdot i, x_i - c_i, \lambda_i > 0.$$

If the sample is linearly inseparable, then the vectors are mapped to a higher dimensional space by replacing the scalar product with one of the nonlinear kernel functions (x_i, x) in the formula. After that, the best separating hyperplane is constructed.

5. Conclusions

The pastoral letters of Andrei Sheptytsky are an important part of the spiritual, social, and cultural heritage that is important for religious communities and society as a whole. The use of mathematical models and machine learning algorithms to analyse these texts can help reveal their internal structure and semantic properties. This will allow for a better understanding of the content and context of pastoral letters, as well as reveal their significance in shaping cultural discourse and moral and ethical values.

Text pre-processing, such as tokenisation, lemmatisation and word root detection, plays an important role in preparing the text for further analysis. Text transformation, such as vector model transformation and clustering, helps to identify similarities between texts and classify them. To reduce the dimensionality of words, lemmatisation or stemming methods are used to generalise features and facilitate classification.

Analysing the frequency of words in Andrey Sheptytsky's pastoral letters, we can conclude that he prefers concepts that are aimed at repentance, love of neighbour, mercy, and peace. Terms related to Christian theology, such as "grace," "salvation," and "repentance," occur with high frequency in Sheptytsky's pastoral letters. Analysing the frequency of words in pastoral texts allows us to gain a better understanding of the structure and content of the text, as well as the important themes that permeate it. Various systems such as Sztergak, Humb, Wingnus, and KP-Miner can be used to perform this analysis, helping to process the text, remove redundant elements, and identify keywords. The use of statistical methods of analysis ensures high reliability of the results of the study of linguistic texts and helps to obtain objective data for linguistic observations. The study of the statistical profile of Andrey Sheptytsky's pastoral letters allows us to use mathematical methods of analysis to understand the linguistic features and structure of the texts. The analysis of the standard error of deviation is important in the study of pastoral letters, as this measure allows us to estimate the spread of the distribution of sample means around the overall mean in the texts, which contributes to a better understanding of their structure and features.

In machine learning, text vectorisation is an important step in extracting features for model training. This process involves converting text into numerical vectors so that the model can work with them. There are various vectorisation techniques, such as bag-of-words and TF-IDF, which help to take into account the importance of words in the text. Additionally, Word2Vec uses neural networks to create vector representations of words, which allows for

contextualisation. The support vector method is used to classify data by finding the optimal hyperplane that will divide it into classes.

Overall, the study shows that machine learning algorithms can be useful in analysing texts with religious and social content, which will contribute to a better understanding of cultural heritage.

References

1. Mogylna M. V., Dubrovin V. I. (2022) Intelektualnyi analiz tekstu: zastosuvannia ta bezkoshtovni prohramni zasoby [Intelligent text analysis: applications and free software tools]. Prykladni pytannia matematychnoho modeliuvannia, vol. 5, no. 2. pp. 41–49.

2. Malyha I. E., Shmatkov S. I. (2022) Machine learning methods for solving semantics and context problems in processing textual data. Visnyk V.N. Karazina Kharkiv National University, seriia "Matematychne modeliuvannia. Informatsiini tekhnolohii. Avtomatyzovane upravlinnia systemy», vol. 56, pp. 35–42.

3. Aggarwal C. C., Zhai C. (2012) Mining Text Data. New York: Springer Science+Business Media. pp. 527.

4. Abhishek J. (2023) Vectorization Techniques in NLP. Retrieved from: https://neptune.ai/blog/vectorization-techniques-in-nlp-guide (accessed 1 March 2024).