Tongue Twisters Detection in Ukrainian by Using TDA

Iryna Yurchuk [1] and Olga Gurnik [2]

[1] *Taras Shevchenko National University of Kyiv, Bohdan Hawrylyshyn str. 24, Kyiv, UA-04116, Ukraine*
[2] *Separate Structural Unit "Vocational College of Engineering, Management and Land Management of National Aviation University", Metrobudivska str. 5-a, Kyiv, UA- 03065, Ukraine*

Abstract

The current stage of development of the digital world requires solving many problems. In particular, the analysis of texts occupies one of the important places. The scope of such analysis is very wide, from the semantic analysis of resumes to the psychological portrait of a social network user based on his posts.

The algorithm of tongue twisters detection in Ukrainian using the topological data analysis methods based on the consideration of a twister as a sample in the four-dimensional space with a construction simplicial structure on it, calculating its invariant and classifiers as machine learning methods to realize distinguishing a tongue twister from a simple narrative sentence is obtained by authors. A better rate of detection was obtained by using a support vector machine with a Gaussian RBF kernel.

Keywords

Ukrainian tongue twister, persistent homology, decision tree, support vector machine

1. Introduction

A tongue twister is a syntactically short, correct phrase spoken in any language with especially complicated articulation. The tongue twisters contain combinations of sounds that sound similar but have different phonemes and are difficult to pronounce. A tongue twister is naive in content, simple, even primitive text, which is built on intricate and difficult phrases and phrases. Often tongue twisters contain rhymes and alliteration. There are several aspects that contribute to the need to study tongue twisters as a cluster of texts in human speech:

- They can be called hard talkers. Working on tongue twisters is very good, very useful diction training. It is a mistake to think that it is necessary to achieve an extremely fast pace. It is not always so. When dealing with hard talkers, as in all work on diction, each speaker requires an individual approach.
- Speech therapists adhere to another name for this type of speech art - pure speaking. Phrases are used to practice pronunciation and diction. Phrases are ideal for differentiating and training automatic pronunciation. For the greater development of the speech apparatus, it is recommended to pronounce colloquialisms with a change in tempo and volume, i.e. quietly, loudly or in a whisper, slowly or quickly. But the main task when reading tongue twisters is to do it clearly and cleanly, with deep, full articulation, regardless of the volume and speed of speech. Politicians, actors and other public figures are well familiar with this type of literary creativity and the miracle that happens to their articulation, diction and manner of speech after a few hard sessions with a good speech therapist.
- Studying the connections between colloquialisms and the peculiarities of the language zone of the brain, scientists found out why colloquialisms are so difficult to pronounce. The reason is the close location of the groups of neurons needed to pronounce a set of sounds in colloquial speech.

COLINS-2023: 7th International Conference on Computational Linguistics and Intelligent Systems, April 20–21, 2023, Kharkiv, Ukraine
EMAIL: i.a.yurchuk@gmail.com (I. Yurchuk); olga.gurnick@gmail.com (O. Gurnik)
ORCID: 0000-0001-8206-3395 (I. Yurchuk); 0009-0008-4186-3044 (O. Gurnik)

Experts have identified two main mechanisms of the so-called “double attack” - a phenomenon when the speaker tries to pronounce two sounds at the same time. The first type of error is when a person tries to pronounce two sounds almost simultaneously, the second - when mixing sounds with a slight delay, and then a vowel sound appears between consonants.

Purpose of the work – to propose an algorithm and its realization for tongue twisters detection in the Ukrainian language that provides the understanding of text shape aspects with the possibility to implement it into machine learning.

The aim of research – to propose topological invariants, the calculation of which will be informative for understanding the nature of a tongue twister, to establish a set of data, the integration of which in a certain method of machine learning has the ability to distinguish a tongue twister from a simple narrative sentence.

Major research objectives are:

- using methods of topological data analysis (TDA), propose an away to encode the Ukrainian tongue twisters that allows to integrate it in machine learning;
- based on obtained results, propose an algorithm for detecting the tongue twisters among simple narrative sentences.

Practical tasks in which the algorithm of tongue twisters detection in Ukrainian can be used are the following:

- rating complexity of existing texts for children who have speech problems. We analyze the sentences of the text and check whether they have the feature of a tongue twister. The more tongue twisters there are, the more difficult for such children a text is.
- artificial generation of a special type of text in the Ukrainian language. It may be senseless, which is allowed by the nature of a tongue twister, but it is aimed at eliminating one or another problem, for example, improving the speech of announcers or developing it in younger children.
- artificial generation of tongue twisters will make it possible to enrich the existing dataset (which is currently consists of no more than 500 items for the Ukrainian language) in order to apply NLPT for tongue twisters detection.

2. Related Works

In this section, we consider some existing works on analyzing text by its figurative shape, text artificiality detection, classification, visualization and methods of machine learning with help of topological data analysis (TDA). Below, the authors focus specifically on existing solutions using topological data analysis.

Note that TDA is a relatively new direction in data analysis, its main advantage is the ability to learn the space characteristics from the point of view of its geometry (for details, see Section 3.3). However, movement in this direction requires an understanding of the basic concepts and aspects of such a science as topology, which often complicates and slows down the process of obtaining new results in this area.

In today's world, social media produces a huge amount of content [1]. This content needs to be processed to identify interesting topics, but this cannot be done manually. Various automated approaches to topic detection have previously been proposed for use. Most of these methods use document clustering and packet detection. For the most part, these approaches represent text functions in standard n-dimensional Euclidean metric spaces. However, these methods have a drawback. It is difficult to determine the subject when filtering noisy documents directly. The authors propose using topology as a subject detection method based on Topological Data Analysis (TDA). This method transforms the Euclidean feature space into a topological space. In this case, the forms of noisy irrelevant documents are much easier to distinguish from thematically relevant documents. This topological space is organized into a network according to the connectivity of points, that is, documents. Therefore, we obtain competitive results by filtering data based on the size of connected components.

In [2], the authors proposed a method to compute the similarity between two users based on their profile images using topological data analysis on Twitter. There were used converting of their captions to numerical vectors usingWord2vec, calculating cosine distances of the caption vectors, and applying them to the topological data analysis approach using mapper. Using k-means, hierarchical clustering

and topological data analysis, the analysis of the popularity of social media images and correlation between image captions and the images' popularity are obtained.

The issue of distinguishing two texts is relevant, the distance between them can be used as a criterion, that is, a metric. The authors [3] provide a first approach to a metric distance between the literary style of these poets (Conceptismo and Culteranismo as the canon of the baroque Spanish literature) using TDA techniques. At the beginning, skipgram was used, having the high-dimensional representation of the words that compose the different sonnets of the dataset following with the Vietoris-Rips filtration computation. The cosine distance as a measure of the similarity between words by the angle of their vectors is used as a metric to compute the Vietoris-Rips filtration and applied in the word2vec algorithm.

For artificial text detection, the experimental setup also highlights the applicability of the features towards the TGM architecture, TGM's size and the decoding method. Notably, the TDA-based classifiers tend to be more robust towards unseen GPT style TGMs as opposed to the considered baseline detectors, see [4]. In addition, it is proposed three types of interpretable topological features that can be derived from the attention maps of any transformer-based LM and shown that the features capture surface and structural properties, lacking the semantic information.

By Wlodek Zadrozny, see [5], the possibility of capturing 'the notion of logical shape in text,' using TDA was researched. But the main result of this paper stated that there is no clear answer to the question:"Can we find a circle in a circular argument?"

Similarly to bag-of-words, there is persistence bag-of-words, see [6]. It is a novel and stable vectorized representation that enables the seamless integration with machine learning. Comprehensive experiments show that the new representation achieves state-of-the-art performance and beyond in much less time than alternative approaches.

Important aspects of text processing are text classification and visualization [7-9] by TDA. For example, see [7], the authors tried to identify the author of a given text based on its content. They considered Hafezian poems and poems of Ferdowsi. As result, it proved that it is possible to divide the number of Hafezian poems from poems of Ferdowsi in mixed data set.

3. Methods

In this section, vectorizing the words, dataset and main terms of persistent homology are considered.

We have to remark that there are several methods for obtaining vector representations for words: unsupervised learning algorithm (GloVe, see [10]), semi-supervised sequence learning (BERT, ELMo, ULMFiT, etc.), supervised learning algorithm (Word2Vec, etc) and construction of model (bag-of-words model, etc.). All of them are part of natural language processing (NLP), and the reader who is interested in these techniques in more detail should study the classic textbooks on this topic. A review of these techniques is not carried out within the scope of this article.

The main disadvantage of all is the fact that the larger the sample (dataset), the better the results. Moreover, the amount of training data is expressed in thousands of units. That is why the authors propose the approach described in this section.

3.1. Principles of letter coding

It is known that there are two classes of letters: consonants and vowels in Ukrainian. Every letter x corresponds to a vector $\vec{x}(x_1, x_2, x_3, x_4)$, where:

- x_1 be the ordinal number of the letter in the text;
- x_2 be the ordinal number of the word in the text which contains letter x;
- x_3 be the ordinal number of alphabetical ordered set of vowels. If a letter is consonant, x_3 is equal to a zero;
- x_4 be the ordinal number of alphabetical ordered set of consonants. If a letter is vowel, x_4 is equal to a zero.

For any tongue twister, there is a map into non negative real four-dimensional space $\mathbb{R}^4_+$.

Since the mapping is carried out in a four-dimensional space, any visualization is complicated by the human perception so it is necessary to reduce the dimensions.

In Fig. 1 and Fig. 2, there are three projections in different spaces of the same tongue twister “Babyn bib rozcviv u doshch, bude babi bib u borshch”. Common to all of them is the fact that the points are grouped into certain clusters, which in the future can ensure the presence of a certain cycle in passing from point (letter) to point (letter) in the space that is coded the tongue twister.

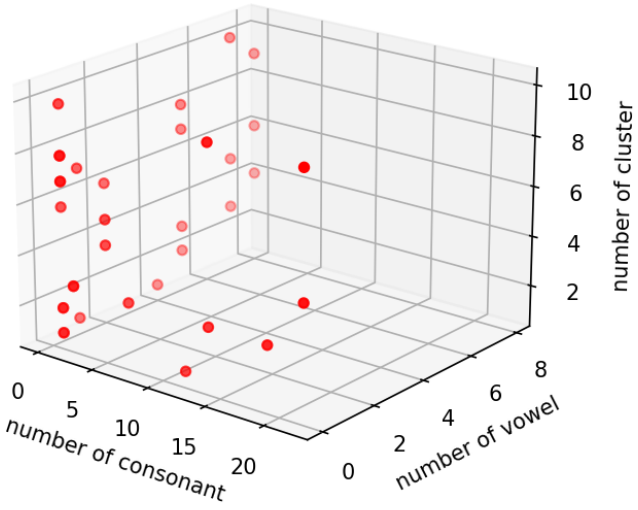

Figure 1: The tongue twister “Babyn bib rozcviv u doshch, bude babi bib u borshch” in the three-dimensional space $X_2X_3X_4$: there are ten words (clusters) in it; the set of points is densely concentrated in a certain part

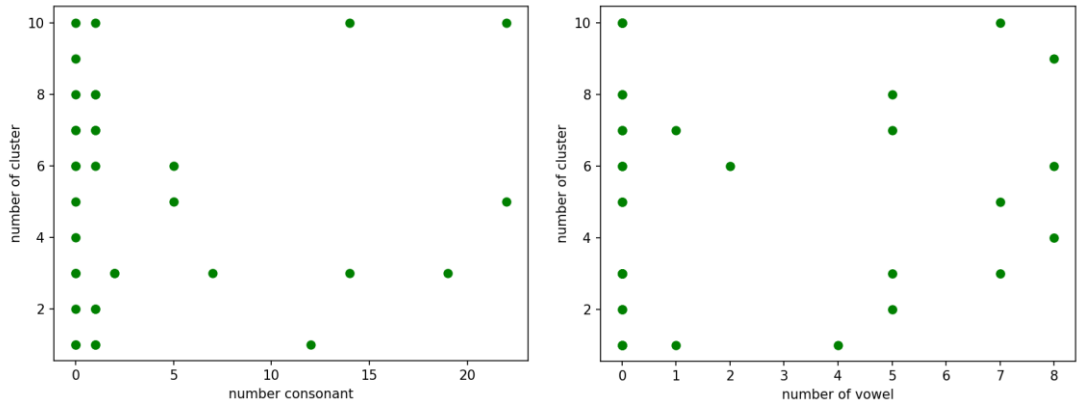

Figure 2: The tongue twister “Babyn bib rozcviv u doshch, bude babi bib u borshch”: it is presented in the two-dimensional spaces X_3X_2 [left] and X_3X_4 [right].

3.2. A Dataset

For research, we propose a dataset that consists of 100 Ukrainian tongue twisters, see [11], and non-twisters. We have to remark that there is a repository of textual data in the Ukrainian language named the UberText corpus. The authors do not use it seeing the texts are preprocessed for searching and their vectorizations are based on methods (LexVec and GloVe) that do not take into account the shape of the structure and their success depends on dataset dimension.

In Fig. 3, there are two histograms of dataset parts used in this research. These histograms show that most tongue twisters have up to 50 letters. They are short and aimed at the quick pronunciation of sounds.

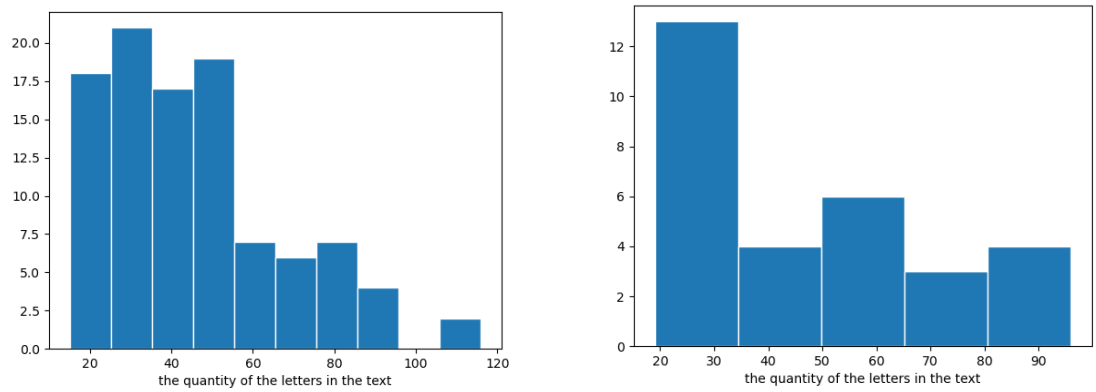

Figure 3: A histogram of the tongue twisters dataset [left], where an axis X is the quantity of the letters in the text and a histogram of the non-twisters dataset [right], where an axis X is the quantity of the letters in the text

This dataset contains 130 short texts, most of which are sentences. The non-twisters were chosen arbitrarily, they are meaningful and are parts of some stories.

3.3. Persistent homologies

Let consider the terms of computational topology. The exact mathematical definitions are in [12,13] The topological invariant of a space the first Betti number (β_1=rank $H_1^{i,j}$) is the amount of cycles of the space. For calculating this invariant we used *the first persistent homology* $H_1^{i,j}$, which is Im $f_1^{i,j}$ for 0≤ i<j ≤k+1, where $f_1^{i,j}: H_1^i \to H_1^j$, $i<j$, be a map. On other words, $H_1^{i,j}=Z_1^i/(B_1^j \cap Z_1^i)$, where Z_1^i is *1*-cycles of C_{ε_i} and B_1^j is *1*-boundaries of C_{ε_j} (a set $\{C_{\varepsilon_i}\}_{i=1}^k$ of Vietoris-Rips complexes is the filtration for any finite set $\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_k\}$, where $\varepsilon_i<\varepsilon_j$, $i<j$.). There is a method of their calculation based on the matrices algebra, the persistence barcode and the persistence diagrams. The 1-*cycle* is a 1-chain with empty boundary. The group of 1-cycles is the kernel of the 1-th boundary homomorphism, Z_1=ker ∂_1. The *1-boundary* is a 1-chain that is the boundary of a 2-chain. The group of *1*-boundaries is the image of the 2-nd boundary homomorphism, B_1=Im ∂_{1+1}. A *1-chain* is a formal sum of 1-simplices in a simplicial complex *K* and its standard notation is $c=\sum a_i\sigma_i$, where σ_i is *p*-simplex in *K* and a_i is either 1 or 0.

For next calculation, we used the GUDHI library, which is a generic open source C++ library with Python interface, for Topological Data Analysis (TDA) and Higher Dimensional Geometry Understanding, see [14].

In Fig. 4, there are two diagrams that are constructed for the tongue twister “Babyn bib rozcviv u doshch, bude babi bib u borshch”. According to topological aspects, the space that has $\beta_0 = 1$, $\beta_1 = 1$ and $\beta_2 = 0$ (Betti numbers), see [12], is topologically equivalent to a circle.

So, if the maximum length of an edge of constructed Vietoris-Rips complex is equal to 0.6 and the minimum persistence is equal to 0.36 then “Babyn bib rozcviv u doshch, bude babi bib u borshch” can be considered as a circle.

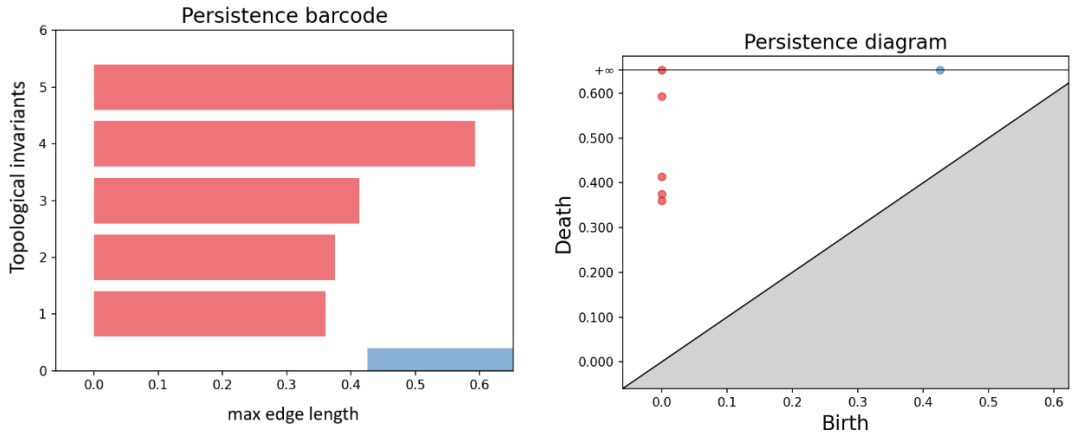

Figure 4: Persistence barcode [left] and persistence diagram [right] of the tongue twister "Babyn bib rozcviv u doshch, bude babi bib u borshch" that is presented as a circle - $\beta_0 = 1$, $\beta_1 = 1$ and $\beta_2 = 0$ (max edge length=0.6 and min persistence=0.36)

The authors have to remark that the advantages of this approach over well-known competitors are the following:

- A text is considered as a figure (discrete structure) into a space. There is a possibility to follow connectivity components (separated pieces) and the existence of cycles (paths whose beginning and end coincide). Whereas GloVe generates a set of vectors without superstructures and structures.
- This approach does not require training and the presence of many hyperparameters for training as, for example, Word2Vec. The only parameter during construction is the selection of the value of the maximum length of the rib during a construction of Vietoris-Rips complex.

4. Algorithm and Experiments

The authors propose the following algorithm:

Step 1. Every letter of the text is coded according to Sec. 3.1. If the text consists of N symbols (letters) $\{s_1, s_2, \dots, s_N\}$, then it corresponds to a set $K = \{(s_1^1, s_2^1, s_3^1, s_4^1), \dots, (s_1^N, s_2^N, s_3^N, s_4^N)\}$. In other words, the text is considered as a cloud of the points in $\mathbb{R}_+^4$. We also normalize it by standard function and map a cloud of the points into I^4, where $I^4 = [0;1]^4$ be a four-dimensional unit cube. Let denote it by $\widetilde{K}$.

Step 2. To construct on $\widetilde{K}$ the filtration C_{ε_j} by Vietoris-Rips complexes, where $\{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_k\}$ is a finite set, ε_i<ε_j, i<j, and computing β_1=rank $H_1^{i,j}$ for every fixed ε_i, $i = \overline{1,k}$. So, then we obtain a set $\{\beta_1^1, \beta_2^1, \dots, \beta_k^1\}$.

Step 3. For every text of dataset, we apply the previous steps and obtain a set M which consists of vectors with $k-$ coordinates. For set M some methods of classification can be used.

In Fig.5, there is a pipeline of an algorithm. A coding corresponds to Step 1, Computation β_1 – Step 2 and Classifier – Step 3. We remark the following:

- The output of *Coding* is a cloud of the points into I^4, where $I^4 = [0;1]^4$ be a four-dimensional unit cube.
- The output of *Computation* β_1 is N numbers of ordered sets of k positive numbers.
- The output of *Classifier* is "Yes or No" answer to "Is this text a tongue twister?"

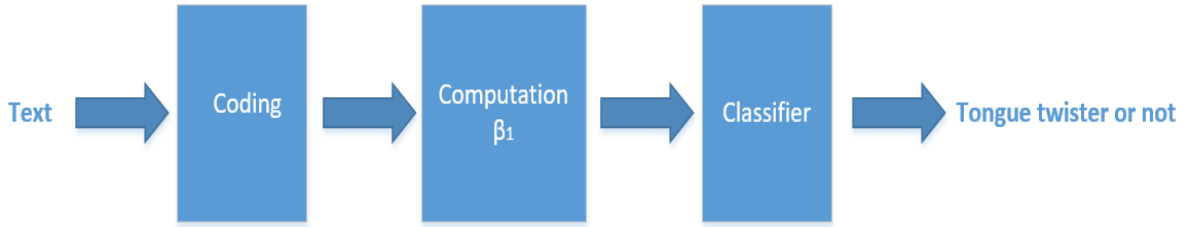

Figure 5: A pipeline of the algorithm

Let denote by D_1 and D_2 two datasets that consist of 100 Ukrainian tongue twisters, 20 non-twisters and 30 non-twisters, respectively. So, $|D_1| = 120$ texts and $|D_2| = 130$ texts.

The authors conducted two experiments. A common for both of them is steps 1 and 2. For both we use datasets D_1 and D_2. In Step 2, we use [0.2, 0.3, …,0.8] for a filtration.

In Step 3, experiment 1 used the Decision Tree Classifier and experiment 2 used the Support Vector Machine Classifier with Gaussian RBF kernel.

We have to remark that both classifiers are implemented using Scikit Learn (one of the most widely used Python packages for Data Science and Machine Learning).

5. Results and discussions

We have to remark that the structure of the filtrations does not depend on the *min_persistence* parameter.

To construct a classifier we use the parameter *test_size* = [0.1,0.2, …, 0.9]. In Table 1, the accuracies of considered classifiers are presented with values of considered parameters.

Table 1

Accuracy of Decision Tree and SVM classifiers with values of considered parameters *test_size* and *random_state*

test_size	10%	20%	30%	40%	50%	60%	70%	80%	90%	random_state
					Dataset D_1					
Decision Tree	0.77	0.68	0.62	0.76	0.64	0.62	0.45	0.66	0.63	0
SVM	0.92	0.72	0.73	0.82	0.82	0.79	0.84	0.79	0.82	0
Decision Tree	0.92	0.68	0.68	0.67	0.66	0.68	0.69	0.72	0.71	1
SVM	0.92	0.68	0.73	0.76	0.77	0.79	0.79	0.78	0.81	1
					Dataset D_2					
Decision Tree	0.86	0.74	0.68	0.6	0.68	0.62	0.63	0.53	0.52	0
SVM	0.79	0.78	0.75	0.77	0.76	0.79	0.78	0.8	0.74	0
Decision Tree	0.43	0.59	0.6	0.6	0.62	0.68	0.61	0.64	0.67	1
SVM	0.57	0.67	0.68	0.68	0.73	0.73	0.73	0.72	0.74	1

Based on the obtained results, we note the following:

- The best indicative tongue twisters detection accuracy is obtained with the following parameters – Dataset 1 with *test size*=10% for both classifiers.
- Tongue twisters detection is stable with an above average score with the following parameters – Dataset 1 by SVM classifier.

- Tongue twisters detection is stable with below average score with the following parameters – Dataset 2 with *random sate* =1 by Decision tree classifier.
- Tongue twisters detection turned out to be the most unstable with the following parameters – Dataset 2 with *random sate* =0 by Decision tree classifier.

In Fig. 6 and Fig.7, there are two trees for classifying twisters. Both of them are realized on Dataset 2 with *test size*=10%.

However, some differences are noticeable:

- For a tree in Fig. 6, there is a parameter *max_depth*=5. It regularized the model and reduced the risk of overfitting accordingly. This effect is clearly visible in the accuracy value, it is higher than in the solution in Fig.7.
- In Fig. 7, the proportion of nodes in which the Gini indicator is close to 0.5 is bigger, which is not a good indicator for Decision tree classifier.
- In Fig. 7, a tree classifier contains many nodes with 1 text.
- Both trees contain a branch with length less than 20 letters. Moreover, the similarity of parameter values for their longer classification of such short tongue twisters 80 and 40(45) should be noted. A small tree (Fig. 6) has only two threshold values for the number of characters that are 35 and 50. Whereas, a large tree (Fig. 7) generates new nodes with a step of 2-3 characters in the text, which makes it very difficult for understanding.

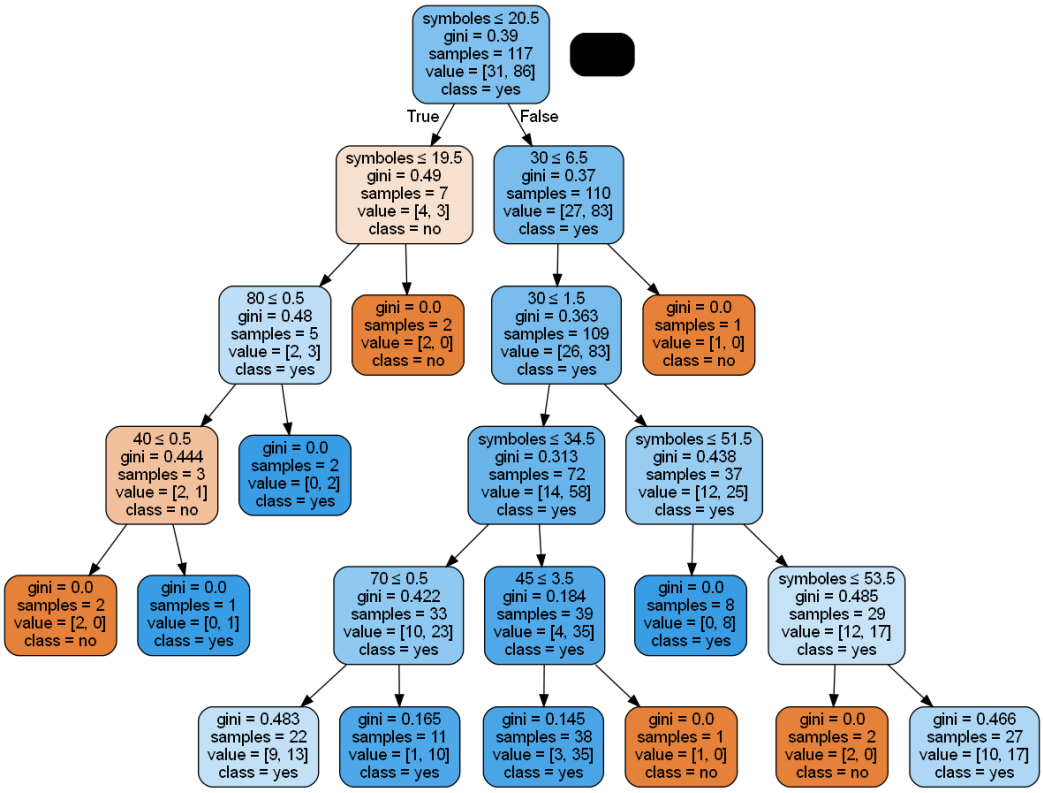

Figure 6: A decision tree, which is constructed with the following parameters: data set =130, max_depth=5, test_size = 0.1, random_state = 0, accuracy = 0.86

The authors remark that the application of the classical statistical approach with the formulation of statistical hypotheses on this sample and determination of the level of significance is complicated. Since the sample can not be expanded to increase the level of significance, due to the presence of established expressions in the language that are tongue-in-cheek, it can only be expanded with the help of artificial methods. For example, if you apply to this sample two-sample t-test to estimate the sample sizes of an

experimental group and a control group that are of equal size, then statistical power is equal to 0.25. On the other hand, using the same assumptions, the presence of more than 1000 instances in the sample does not guarantee 0.99, since models can have hundreds of thousands of data, but cannot have high training.

That is why the selection of machine learning models and data mining is an art starting from the formation of a dataset and ending with the selection of models and their parameters. There are no theorems and statements that would clearly establish how and on what to teach the module in order to obtain a high level of recognition.

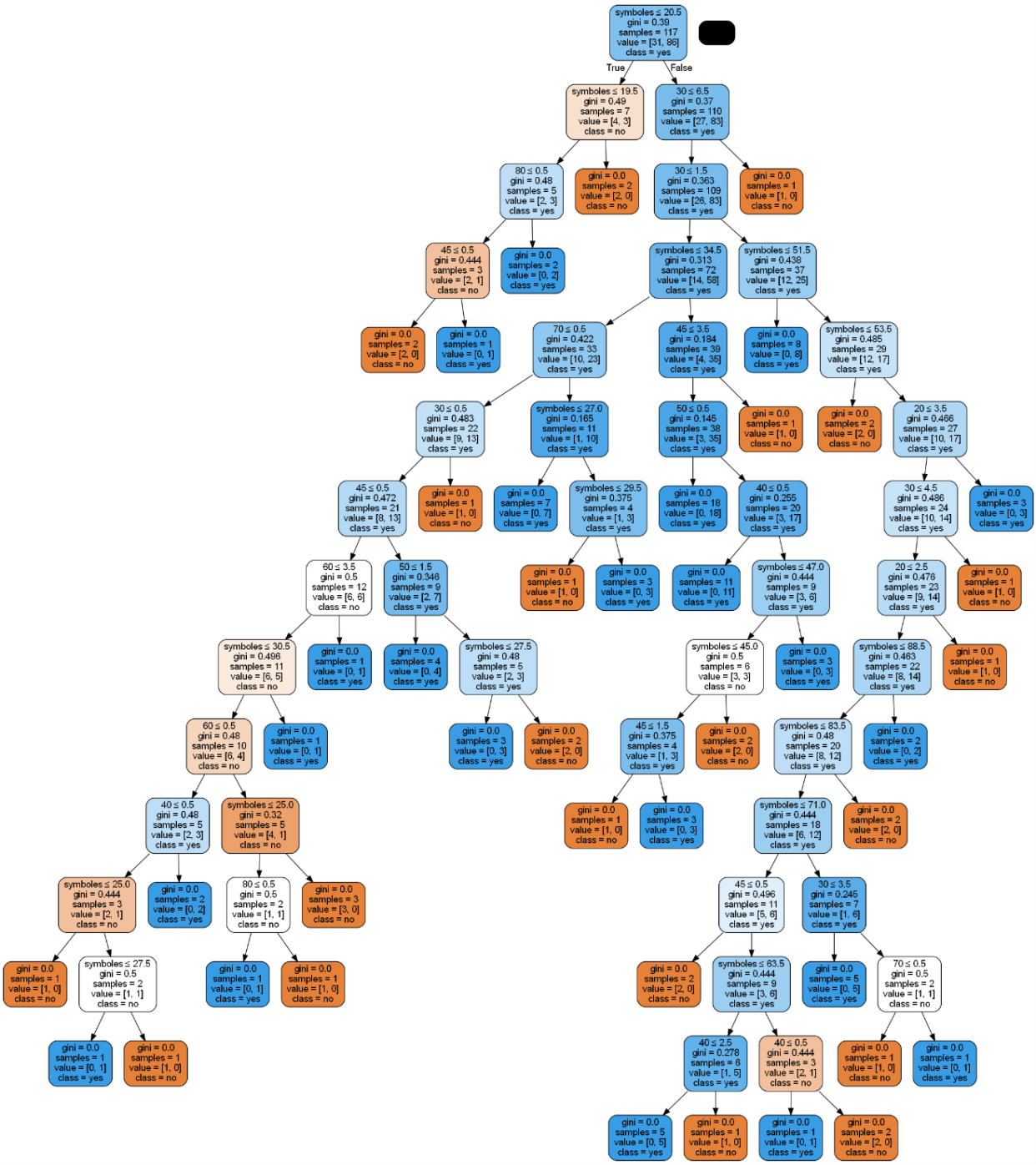

Figure 7: A decision tree, which is constructed with the following parameters: data set =130, test_size = 0.1, random_state = 0, accuracy = 0.71

6. Conclusion

An algorithm of tongue twisters detection in Ukrainian based on the consideration of a text as a sample of points in four-dimensional space, calculation of Betti numbers as a topological invariant of the existence of one-dimensional circles in it and implementation it into machine learning is proposed. This approach makes it possible to see behind the text not only a set of points, but also the possibility of generating a structure in space with a longer calculation of its invariants, the values of which make it possible to see its geometry (a sentence is as "circle", "sphere" or "torus").

In addition, there were realized two methods of machine learning as classifiers for detecting tongue twisters among simple narrative sentences. According to research, more accuracy is provided by Support Vector Machine Classifier with Gaussian RBF kernel.

In further research, there are several ways to improve the accuracy of detection, as well as to improve the process of encoding words or letters taking into account the analysis of the sounds, as well as the complexity in the pronunciation of sounds and calculating of topological invariants that contain cyclomaticity of higher orders.

7. References

[1] P. Torres-Tramón, H. Hromic, B. Heravi, Topic detection in twitter using topology data analysis, in: Proceedings of 15th International Conference, ICWE 2015 Workshops, NLPIT, PEWET, SoWEMine, Rotterdam, The Netherlands, 2016, pp. 186-197. doi: 10.1007/978-3-319-24800-4_16.

[2] K. Almgren, Employing topological data analysis on social networks data to improve information diffusion (Computer Science), Ph.D. thesis, the school of engineering university of Bridgeport, Connecticut, USA, 2018.

[3] E. Paluzo-Hidalgo, R. Gonzalez-Diaz, M. A. Gutierrez-Naranjo, Towards a philological metric through a topological data analysis approach, ArXiv, 2019, abs/1912.09253.

[4] L. Kushnareva, D. Cherniavskii, V. Mikhailov, E. Artemova, S. Barannikov, A. Bernstein, I. Piontkovskaya, D. Piontkovski, E. Burnaev, Artificial text detection via examining the topology of attention maps, in: Proceedings of Conference on Empirical Methods in Natural Language Processing, Punta Cana, Dominican Republic, 2021, pp. 635–649. doi: 10.18653/v1/2021.emnlp-main.50.

[5] W. Zadrozny, A note on argumentative topology: circularity and syllogisms as unsolved problems, arXiv - CS - Computation and Language, 2021. doi:arxiv-2102.03874.

[6] B. Zielinski, M. Lipinski, M. Juda, M. Zeppelzauer, P. Dłotko, Persistence bag-of-words for topological data analysis, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, 2019, pp. 4489-4495. doi: 10.24963/ijcai.2019/624.

[7] N. Elyasi, M. H. Moghadam, An introduction to a new text classification and visualization for natural language processing using topological data analysis, 2019, arXiv. URL: https://arxiv.org/abs/1906.01726.

[8] B. Yue, Topological data analysis of two cases: text classification and business customer relationship management, J. of Physics, 1550(2020). doi:10.1088/1742-6596/1550/3/032081.

[9] Sh. Gholizadeh, K. Savle, A. Seyeditabari, W. Zadrozny, Topological data analysis in text classification: extracting features with additive information, arXiv, 2020. URL: https:// https://arxiv.org/abs/2003.13138.

[10] J. Pennington, R. Socher, C. D. Manning, GloVe: Global Vectors for Word Representation. URL: https://nlp.stanford.edu/projects/glove/.

[11] Top 100 Ukrainian tongue twisters by Leopolis published 17.11.2021 URL: https://lviv1256.com/lists/top-100-ukrajinskyh-skoromovok/.

[12] G. Carlsson, Topology and data, Bull.Amer.Math.Soc, 46(2) (2009): 255–308.

[13] I. Yurchuk, Digital image segmentation based on the persistent homologies, in: Proceedings of the 1st International Workshop on Information-Communication Technologies and Embedded Systems, ICTES, Mykolaiv, Ukraine, 2019, pp. 226-232.

[14] The GUDHI library. URL:https://gudhi.inria.fr/