Theoretical and applied bases of creating a system for automatic phraseological units classification in English texts based on the artificial intelligence technologies usage

Mikolaj Karpinski[1,†], Liudmyla Borovyk[2,*,†], Serhii Lienkov[3,†], Vitalii Savchenko[4,†], Iryna Panibrat[5,†], and Vadym Sobko[6,†]

[1] *Institute of Security and Computer Science University of the National Education Commission, 2 Podchorazych str, Krakow, 30-084 Poland*

[2,5,6] *Bohdan Khmelnytskyi National Academy of the State Border Guard Service of Ukraine, Shevchenko str., 46, Khmelnytskyi, 29000, Ukraine*

[3] *Military Institute, Taras Shevchenko National University of Kyiv, Zdanovska str. 81a, Kyiv, 03189, Ukraine*

[4] *State University of Information and Communication Technologies, Solomianska str. 7, Kyiv, 03680, Ukraine*

Abstract

The paper sets and investigates the task of developing a system for automatic classification of phraseological units in English texts, designs the structure of the corresponding system and implements it in software. The paper proposes a hybrid method for automatic classification of phraseological units in English texts, the main idea of which is to use a rule-based method to identify and distinguish specific types of phraseological units and further apply machine learning methods to classify them based on semantic and syntactic properties. To implement the hybrid method, a system is proposed, the structure of which includes the following modules: Hybrid Soft; tokenization; tagging; base determination; division; corpus module; classification module. The Python language was used to develop a system for automatic classification of phraseological units in English texts that implements the hybrid method. The use of the proposed system allows not only to reduce the time required for processing phraseological units but also to establish additional regularities for the classification features by which phraseological units are classified.

Keywords

phraseological units, automatic classification, artificial intelligence, information technology, algorithm, method.

1. Introduction

The study of phraseology is an important part of linguistics as it helps to understand the complex structure of language and how it is used in communication. In recent years a lot of

CLW-2025: Computational Linguistics Workshop at 9th International Conference on Computational Linguistics and Intelligent Systems (CoLInS-2025), May 15–16, 2025, Kharkiv, Ukraine

[*] Corresponding author.

[†] These authors contributed equally.

mikolaj.karpinski@uken.krakow.pl (M. Karpinski); blv_nadpsu@ukr.net (L. Borovyk); lenkov_s@ukr.net (S. Lienkov); savitan@ukr.net (V. Savchenko); irynaborovyk2017@ukr.net (I. Panibrat); sgvadik@gmail.com (V. Sobko);

0000-0002-8846-332X (M. Karpinski); 0000-0003-2949-2187 (L. Borovyk); 0000-0001-7689-239X (S. Lienkov); 0000-0002-3014-131X (V. Savchenko); 0000-0002-3209-9119 (I. Panibrat); 0009-0004-6109-0262 (V. Sobko)

attention has been paid to the study of natural language processing in general and phraseology in particular. This is due to the improvement of research tools. In particular, the development of information technology and artificial intelligence has made it possible to develop systems capable of analyzing and understanding human speech. The computational capabilities of these tools increase the number of linguistic tasks that can be solved and deepen the level of their processing.

One of the key problems in natural language processing (NLP) is phrase recognition and classification. The need to automate this task is explained by the fact that many natural language processing applications such as text mining, information search, machine translation, and natural language generation, require preliminary phrase recognition and classification. Automatic classification of phraseological units in English texts is an important task in natural language processing which involves the identification and categorization of groups of words or phrases based on their semantic and syntactic properties.

The subject area of automatic phraseological unit (PhU) classification lies at the intersection of computational linguistics and natural language processing. It involves the development of algorithms and methods that can automatically identify and classify PhUs that are fixed or semi-fixed expressions in a language with figurative or idiomatic meanings that cannot be easily derived from the meanings of their individual words.

Thus, automatic phraseology classification is a complex task that requires a combination of linguistic knowledge, computing and algorithms.

Taking into account the shortcomings of existing approaches to the classification of phraseological units and the peculiarities of the functioning of software tools for the automatic classification of phraseological units, the task of developing an effective system for the automatic classification of phraseological units in English texts is becoming increasingly important. The effectiveness of the system implies its reliability, applicability to the processing of various sentence structures and different types of phraseological units, including fixed expressions, idioms and phrases, as well as minimization of the shortcomings typical of existing tools.

Therefore, the purpose of the article is to elaborate the theoretical and applied foundations for the development of an effective system for automatic classification of phraseological units in English texts based on the use of modern information technologies in general and artificial intelligence in particular.

2. Related works

Recently, the issue of classification of phraseological units in general and automatic classification in particular has been the subject of research by a number of philologists and specialists in the field of information technology [1-6]. As part of their research they analyzed various approaches, methods and developed software and hardware tools for automatic classification of PhUs.

Several approaches have been proposed for the automatic classification of PhUs which can be generally divided into three categories: rule-based approach, statistics-based approach and machine learning approach. The following methods have also been used to classify phraseological units in the raw text: the use of constructed «local» grammars; the use of dictionaries; the use of statistical processes.

Table 1 provides a comparative assessment of these approaches by various criteria.

Table 1
Evaluation of different approaches for automatic classification of PhUs

Approach	Comparative characteristics of the approaches			
	The essence of the approach	Mathematical methods underlying the approach	Benefits of the approach	Disadvantages of the approach
Rule-based	Relying on predefined rules to identify and classify PhUs	Formal grammars, regular expressions	Flexibility, clarity, high accuracy	Limited scalability, error-prone, maintenance overhead, lack of adaptability
Statistics-based	Uses statistical measures of frequency and distribution of expressions to identify and classify PhUs	Probabilistic models, clustering, associative rules	Scalability, data management, automatic feature extraction, data processing speed, generalization	Lack of contextual understanding, data displacement, domain dependency
Machine learning-based	Uses a set of features derived from expressions and a labeled dataset to train a classifier that can automatically identify and classify new expressions	Decision trees, support vectors, deep neural networks	High accuracy, contextual understanding, reliability, scalability	Requires large amounts of training data, retooling, limited interpretation capabilities, lack of transparency

The analysis of the data in Table 1, the methods of classifying PhUs and a number of thematic scientific sources, in particular [1-6], allows us to conclude that there are currently no perfectly working approaches and methods that could ensure the identification and classification of PhUs in any English text without error and distortion.

It should also be noted that there are several basic software tools available for automatic classification of financial institutions:

Sketch Engine: Sketch Engine is a web-based corpus management and analysis tool that provides various functions for language processing including automatic classification of PhUs [7]. It uses statistical analysis and machine learning algorithms to identify and classify the PhUs in a corpus of text.

Linguistic Investigation and Word Count (LIWC): LIWC is a software tool that provides text analysis and language processing capabilities. It contains the function of identifying and classifying the PhUs in the text corpus based on the created rules and statistical analysis [8].

ConText: ConText is a software tool that provides natural language processing capabilities including automatic classification of PhUs. It uses machine learning algorithms to identify and classify PhUs in clinical text.

Natural Language Toolkit (NLTK) [9]: NLTK is a Python library that provides various functionalities for natural language processing including automatic classification of phraseological units. It contains a module for identifying and classifying phraseological units based on manual rules and statistical analysis [10].

PhraseDetective: PhraseDetective is a web-based software tool that provides automatic classification of the phraseological units in a corpus of text. It uses a combination of statistical analysis and machine learning algorithms to identify and classify the phraseological units.

In general, existing software tools for automatic classification of PhUs provide a number of functionalities but there is a need for further research and development in this area. This is due to the need to improve the accuracy and efficiency of software tools.

3. Materials and methods

In order to achieve this goal, it seems advisable to clearly formulate the task of developing a system for the automatic classification of PhUs in English texts, to design the structure of the relevant system and to implement it in software.

Problem Statement of the Development of a System for Automatic Classification of PhUs in English Texts

The input to the system should be an array of English text consisting of sentences, paragraphs or large text segments. The output of the system should be a list of PhUs found in the text representing the identified phrases, and classified according to certain features.

The system should process the input data and automatically generate the result.

The system should demonstrate the following key capabilities:

1. Comprehensive perception: the system should be able to process a large amount of text covering different genres, registers and linguistic contexts. It should be able to perceive the PhUs in complex sentence structures such as compound or complex subordinate clauses as well as interrogative and exclamatory sentences. The system should be able to classify a wide range of linguistic units including fixed expressions, idiomatic phrases, phrases and other lexical combinations.

2. Accurate identification and classification: the system should provide high accuracy in recognizing and classifying the PhUs in the text. It should use linguistic models, parsing, semantic information and contextual clues to distinguish the PhUs from ordinary language usage. The system should apply rule-based and machine learning techniques to improve the accuracy and memorability of the classification of the PhUs.

3. Scalability and efficiency: the system should be able to process large amounts of text efficiently. It should optimize computing resources to provide timely results even when processing large corpora or real-time text streams.

4. Further analysis: the system should produce a list of identified and classified PhUs as an output. The output should be suitable for further analysis such as language modeling, corpus linguistics research or other natural language processing tasks.

Designing the structure of the system for automatic classification of PhUs in English texts.

The theoretical basis of the developed system of automatic classification of PhUs in English texts should be an improved method of automatic classification of PhUs which would combine the strengths of rule-based and machine learning methods. In the following, we will call this method a hybrid method. The hybrid method should provide a more accurate classification of the PhUs in English texts.

The main idea of the hybrid method is to use a rule-based method to identify and distinguish specific types of PhUs and then apply machine learning methods to classify PhUs based on their semantic and syntactic properties. For example, a rule-based method can be used to identify nouns that are composed of a noun and an adjective and a machine learning algorithm can be used to classify these nouns based on their semantic similarity to other known PhUs.

There are several algorithms and technical tools that can be used to achieve this combination of rule-based and machine learning techniques. For example the Natural Language Toolkit (NLTK) in Python which contains tools for pattern matching, parsing, and feature extraction as well as algorithms for classification, clustering, and information search [11]. Another natural language processing tool can be spaCy which contains a rule-based matching system for detecting specific patterns in text as well as a machine learning line for training and evaluating custom models for classification and other tasks [12].

Feature extraction involves identifying the characteristics of the PhUs that can distinguish it from other types of expressions. These features may include frequency of occurrence, length of the expression, presence of certain words or parts of speech and other linguistic properties.

Classification involves the use of machine learning algorithms to group similar PHUs based on selected features.

The classification process can be supervised or unsupervised. In supervised learning, the algorithm is trained on a labeled dataset while in unsupervised learning, the algorithm identifies data patterns without prior knowledge of the categories [13]. There are several machine learning algorithms that can be used to automatically classify phraseology including supervised learning, unsupervised learning, and semi-supervised learning [14].

Supervised learning involves training a machine learning algorithm on a labeled dataset of PhUs and their corresponding categories. The algorithm learns to identify the characteristics and properties of different types of phrases and uses this knowledge to classify new phrases. For example a supervised learning algorithm can be trained on a dataset of idiomatic expressions and their corresponding categories (e.g., food-related idioms, weather-related idioms etc.) to accurately classify new examples of idiomatic expressions based on their semantic and syntactic properties [15].

Unsupervised learning involves training a machine learning algorithm on an unlabeled set of PhUs data allowing the algorithm to detect patterns and similarities in the data without any prior knowledge of the categories. This approach is useful when the categories of phrases are unknown, or when there are too many categories to be labeled manually. For example, the algorithm can be used to combine similar idiomatic expressions into groups based on their semantic and syntactic properties [16].

Semi-supervised learning involves a combination of supervised and unsupervised learning where the algorithm is trained on a small labeled data set and a large unlabeled data set. The algorithm uses the labeled dataset to learn the characteristics and properties of the categories and then applies this knowledge to the unlabeled dataset to identify similar instances. This approach can be useful when the labeled dataset is small or when labeling the entire dataset is not possible [17].

When developing an information system for automatic classification of PhUs we divide the structure of the information system into three main components: data pre-processing; feature extraction; and classification.

For example, let's imagine that we have a set of PhUs that have been labeled as idioms or phrases. To classify a new PhUs as an idiom or a phrase we can apply the k-nearest neighbor algorithm (k-NN). Its characteristics such as frequency and length, are established, and its k nearest neighbors in the dataset are determined. If most of the neighbors are labeled as idioms, the PhU can be classified as an idiom.

Data preprocessing involves cleaning and organizing data before it is divided into feature extraction and classification algorithms. This process includes removing irrelevant information such as stop words and converting the text into a format that can be easily processed by the algorithms. Feature extraction involves identifying the characteristics of phrases that can distinguish them from other types of expressions. These features may include frequency of use, length of the expression, presence of certain words or parts of speech and other linguistic properties. Mathematical formulas can be used to extract features from the data.

Other linguistic properties such as the presence of certain words or parts of speech can be determined using NLP methods [18]. Classification involves grouping similar phrases based on the features identified during the feature extraction process. Mathematical algorithms can be used to classify phrases based on their features.

The bag-of-words (BOW) model and the word embedding model are two popular methods used for feature extraction. In the BOW model each phraseological unit is represented as a vector of word frequencies. The number of times each word appears in the unit is counted, and the resulting vector has a dimension corresponding to the number of unique words in the corpus [19]. For example the phraseological units «A piece of cake», «Break a leg» and «Hit the sack» will be represented as follows:

[1, 1, 1, 0, 0, 0, 0]

[0, 1, 0, 1, 0, 0, 0]

[0, 1, 0, 0, 0, 1, 0]

Each dimension of the vector represents the frequency of the corresponding word in the phraseology.

The word embedding model represents each word as a vector in a multidimensional space. The vector representation of each PhU is created by averaging the vectors of the words that make up the unit. For example the vector representation of the words of the phraseological units «A piece of cake», «Break a leg» and «Hit the sack» will be as follows:

[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7]

[0.5, 0.6, 0.7, 0.8, 0.9, 0.1, 0.2]

[0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3]

Each dimension of the vector represents the average value of the corresponding vector of words in the phrase.

After feature extraction, the next step is classification. One example of classifying PhUs by clear features is to distinguish between idioms and phrases by their structure and predictability. For example the idiom «kick the bucket» means «to die» and cannot be understood by looking up the meaning of «kick» or «bucket» in a dictionary. Idioms often have a metaphorical or figurative meaning that is not related to their literal meaning [20]. On the other hand, phrases themselves. They can be predicted to some extent based on the meaning of individual words.

For example, let's look at the classification of phraseological units by four features:

1) Structure. For example, the phrase «kick the bucket» is an idiom which means that it has a fixed structure and cannot be understood based on the meaning of its individual words.

2) Semantic connection. For example, the phrase «strong coffee» is a collocation, i.e. it consists of words that often occur together due to semantic similarity.

3) Function. For example, the phrase «on the other hand» is a discourse marker, which means that it is used to indicate a contrast or an alternative point of view.

4) Origin. For example, the phrase «faux pas» is a loanword which means that it was borrowed from the French language and is commonly used in English to refer to a social mistake or blunder [21].

Several methods can be used to classify PhUs including k-nearest neighbors, decision trees, support vector machines (SVMs) and neural networks. The choice of a classification method depends on the size of the data set, the complexity of the PhUs and the desired classification accuracy.

In the k-nearest neighbors method, a phraseology is classified based on the class label of its k nearest neighbors in the feature space.

The decision tree method creates a tree that represents decision rules for assigning class labels to phraseological units.

The SVM uses a hyperplane to divide phrases into different classes based on their feature representations.

In neural networks a deep learning model is trained on the feature representations of phrases to predict their class membership [22].

The development of an information system for automatic classification of PhUs has certain difficulties. One of the biggest challenges is the diversity and complexity of PhUs in different languages and cultures. In addition, the accuracy of the classification task strongly depends on the quality of the training data and the choice of feature extraction and classification methods. For example, the BOW model is simple and effective but it does not take into account the semantic relations between words in phraseological units. In contrast, the word embedding model takes into account the semantic relations between words but it requires a large amount of training data and can be computationally expensive. Therefore, the choice of a feature extraction method should be based on the characteristics of the dataset and the available computing resources. Likewise, the choice of classification method depends on the size of the dataset, the complexity of the feature and the desired classification accuracy. For example, the k-nearest neighbors method is simple and easy to implement but it may not work properly when the dataset is large and the number of classes is high. In contrast, the neural network method is more complex and computationally expensive but it can achieve high accuracy even with large and complex datasets [23].

The conducted analysis and research allow us to propose the author's information system for implementing the hybrid method for categorizing phrases:

Data collection: the system collects data that includes a set of texts or documents to be categorized. The data may also include phraseological units and related categories that can be used as training data for a machine learning model.

Pre-processing: The system pre-processes the data to prepare it for classification. This includes tasks such as tokenization, stop word removal, stemming, and normalization.

Rule-based method: the system applies a method to identify and classify phraseological units in the text. This can be done by creating a set of rules that match certain patterns or sequences of words that correspond to phrases [24].

Machine learning-based method: the system uses a method to classify the text. This can be done by training a classification model on training data that includes phraseology and related categories. The machine learning model can then be used to automatically classify the phraseology in the text.

Hybrid Method: The system combines the results of a rule-based method and a machine learning-based method to improve classification accuracy. This can be done by applying the machine learning model only to idioms that were not classified by the rule-based method or by using the rule-based method to refine the output of the machine learning model.

Evaluation: The system evaluates classification performance using metrics such as precision, recall, and F1-score. This can be done by comparing the system's output with a set of manually classified texts.

Output: The system outputs classification results which can be used for various purposes, such as information retrieval, text analysis, or sentiment analysis [25].

4. Experiment

Classification of phraseological units using HS software can be implemented with the following sequence of steps:

1) HS Installation: Installation of the HS library in the system. HS can be installed using pip, the Python package installer. Next, the command line interface is opened, and the command «pip install HS» is executed.

2) Import HS and Data Preprocessing: Import the necessary HS modules and packages into the Python code. In addition the PhUs dataset is preprocessed to ensure consistent formatting and remove any irrelevant information. This may include removing punctuation, converting to lowercase or applying stemming.

3) Feature Extraction: Defining the features that will be used for classification. In the case of phraseological units features can be based on usage frequency, part-of-speech tags, or other linguistic characteristics.

4). Dataset Split: Split the preprocessed dataset into training and testing sets. The training set will be used to train the classification model while the testing set will be used to evaluate the model's performance.

5). Classification Algorithm Selection: Choose a classification algorithm from the available options in the HS software: the Naive Bayes algorithm and the Named Entity Recognition (NER) algorithm.

Classifier Training: Use the training set to train the selected classification algorithm. Provide the features extracted from the training data along with the corresponding labels (categories).

6). Model Evaluation: Test the trained classifier on the test set. Then measure accuracy or other relevant performance metrics to assess how well the model can classify PUs.

7). Using the Model for Prediction: Once the model is trained and evaluated, it can be used to predict the categories of new, unseen phraseological units. Provide the extracted features of the new data to the classifier and obtain the predicted categories.

An example code snippet demonstrating the implementation of the above steps using HS software is shown in Figure 5.

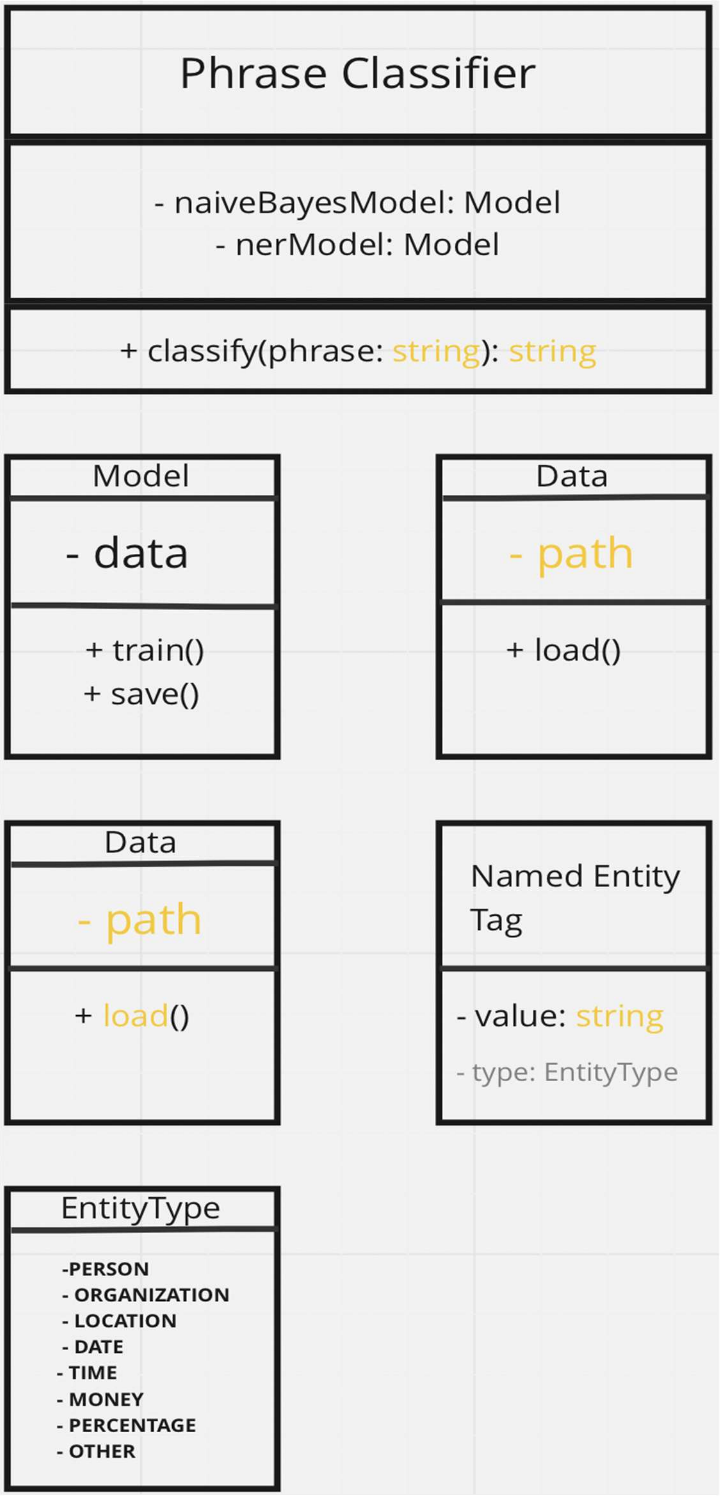
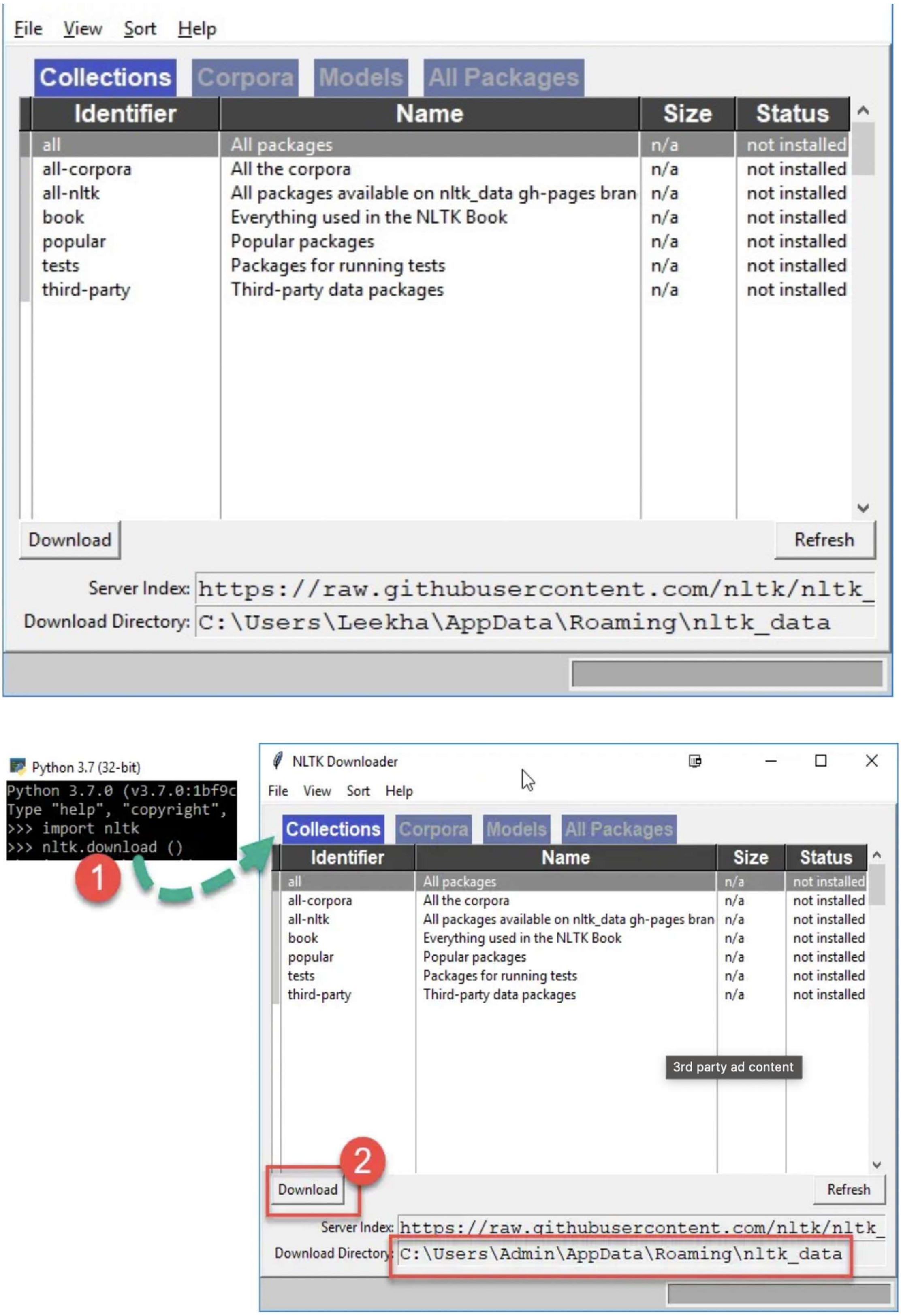


Figure 2: UML diagram of the main components and their relationships in the hybrid algorithm



Identifier	Name	Size	Status
all	All packages	n/a	not installed
all-corpora	All the corpora	n/a	not installed
all-nltk	All packages available on nltk_data gh-pages bran	n/a	not installed
book	Everything used in the NLTK Book	n/a	not installed
popular	Popular packages	n/a	not installed
tests	Packages for running tests	n/a	not installed
third-party	Third-party data packages	n/a	not installed

Server Index: https://raw.githubusercontent.com/nltk/nltk_

Download Directory: C:\Users\Leekha\AppData\Roaming\nltk_data

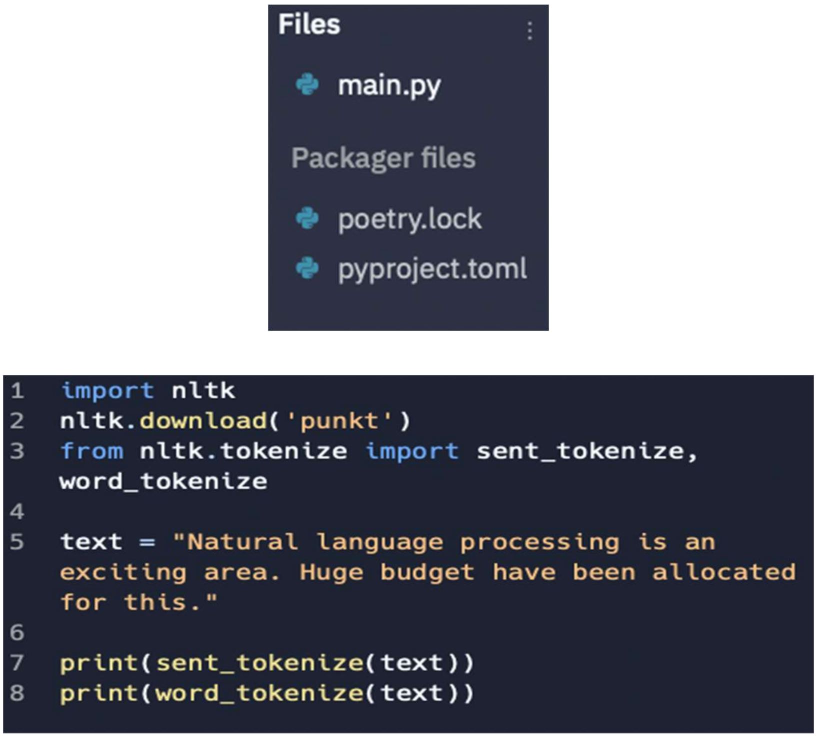
```
Python 3.7.0 (v3.7.0:1bf9c
Type "help", "copyright",
>>> import nltk
>>> nltk.download ()
```

Identifier	Name	Size	Status
all	All packages	n/a	not installed
all-corpora	All the corpora	n/a	not installed
all-nltk	All packages available on nltk_data gh-pages bran	n/a	not installed
book	Everything used in the NLTK Book	n/a	not installed
popular	Popular packages	n/a	not installed
tests	Packages for running tests	n/a	not installed
third-party	Third-party data packages	n/a	not installed

Server Index: https://raw.githubusercontent.com/nltk/nltk_

Download Directory: C:\Users\Admin\AppData\Roaming\nltk_data

Figure 3: Software interface



```
import nltk
nltk.download('punkt')
from nltk.tokenize import sent_tokenize, word_tokenize

text = "Natural language processing is an exciting area. Huge budget have been allocated for this."

print(sent_tokenize(text))
print(word_tokenize(text))
```

Figure 4: Tokenization of code in HS software

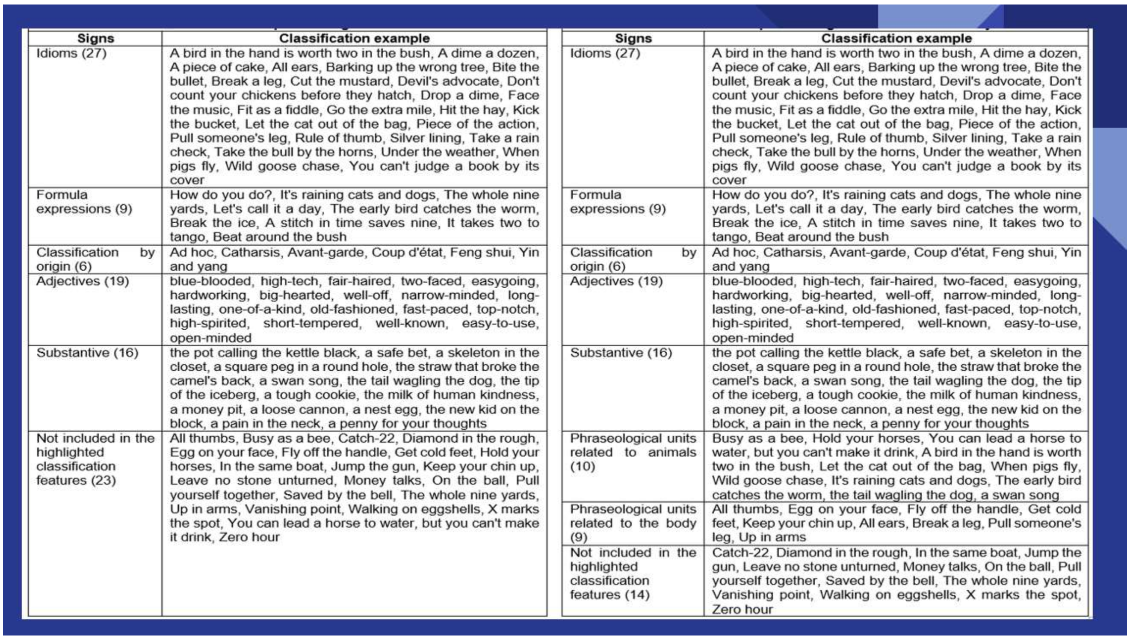
```
import hs
from hs.corpus import stopwords
from hs.tokenize import word_tokenize

# Preprocess the dataset
phraseological_units = [
    ("All thumbs", "Category1"),
    ("Busy as a bee", "Category2"),
    # ... Add more phraseological units with their corresponding categories
]

# Remove stopwords and perform tokenization
stop_words = set(stopwords.words('english'))
tokenized_units = []
for unit, category in phraseological_units:
    words = word_tokenize(unit)
    filtered_words = [word for word in words if word.lower() not in stop_words]
    tokenized_units.append((filtered_words, category))

# Extract features
all_words = hs.FreqDist(word.lower() for unit, _ in tokenized_units for word in unit)
word_features = list(all_words.keys())[:50]  # Use top 50 frequent words as features

def extract_features(unit):
    unit_words = set(unit)
    features = {}
    for word in word_features:
        features[word] = (word in unit_words)
    return features

# Split dataset into training and test sets
split_ratio = 0.8
split_index = int(len(tokenized_units) * split_ratio)
training_set = [(extract_features(unit), category) for unit, category in tokenized_units[:split_index]]
test_set = [(extract_features(unit), category) for unit, category in tokenized_units[split_index:]]

# Train the Naive Bayes classifier
classifier = hs.NaiveBayesClassifier.train(training_set)

# Evaluate the classifier
accuracy = hs
```

Figure 5: Code snippet demonstrating the operation of HS software

5. Discussion

Figure 6 shows the results of applying the existing and proposed software to solve the research problem. Sketch Engine software was used as the existing software. 100 phraseological units were selected for classification. The classification results were as follows. As a result of the Sketch Engine software's work, 27 PhUs were assigned to group 1, 9 to group 2, 6 to group 3, 19 to group 4, 16 to group 5, and 23 PhUs were not classified according to a specific feature. The automatic classification time was 27 seconds.

The classification results of the same PhUs by the author's Hybrid Soft software were as follows: 27 PhUs were assigned to group 1, 9 to group 2, 6 to group 3, 19 to group 4, 16 to group 5, and 14 PhUs did not fall into a specific group. Moreover, thanks to the author's method, the algorithm was able to identify 2 additional features and classify another 19 PhUs into the newly created 2 groups. The automatic classification time was 19 seconds.



Signs	Classification example
Idioms (27)	A bird in the hand is worth two in the bush, A dime a dozen, A piece of cake, All ears, Barking up the wrong tree, Bite the bullet, Break a leg, Cut the mustard, Devil's advocate, Don't count your chickens before they hatch, Drop a dime, Face the music, Fit as a fiddle, Go the extra mile, Hit the hay, Kick the bucket, Let the cat out of the bag, Piece of the action, Pull someone's leg, Rule of thumb, Silver lining, Take a rain check, Take the bull by the horns, Under the weather, When pigs fly, Wild goose chase, You can't judge a book by its cover
Formula expressions (9)	How do you do?, It's raining cats and dogs, The whole nine yards, Let's call it a day, The early bird catches the worm, Break the ice, A stitch in time saves nine, It takes two to tango, Beat around the bush
Classification by origin (6)	Ad hoc, Catharsis, Avant-garde, Coup d'état, Feng shui, Yin and yang
Adjectives (19)	blue-blooded, high-tech, fair-haired, two-faced, easygoing, hardworking, big-hearted, well-off, narrow-minded, long-lasting, one-of-a-kind, old-fashioned, fast-paced, top-notch, high-spirited, short-tempered, well-known, easy-to-use, open-minded
Substantive (16)	the pot calling the kettle black, a safe bet, a skeleton in the closet, a square peg in a round hole, the straw that broke the camel's back, a swan song, the tail wagling the dog, the tip of the iceberg, a tough cookie, the milk of human kindness, a money pit, a loose cannon, a nest egg, the new kid on the block, a pain in the neck, a penny for your thoughts
Not included in the highlighted classification features (23)	All thumbs, Busy as a bee, Catch-22, Diamond in the rough, Egg on your face, Fly off the handle, Get cold feet, Hold your horses, In the same boat, Jump the gun, Keep your chin up, Leave no stone unturned, Money talks, On the ball, Pull yourself together, Saved by the bell, The whole nine yards, Up in arms, Vanishing point, Walking on eggshells, X marks the spot, You can lead a horse to water, but you can't make it drink, Zero hour

Signs	Classification example
Idioms (27)	A bird in the hand is worth two in the bush, A dime a dozen, A piece of cake, All ears, Barking up the wrong tree, Bite the bullet, Break a leg, Cut the mustard, Devil's advocate, Don't count your chickens before they hatch, Drop a dime, Face the music, Fit as a fiddle, Go the extra mile, Hit the hay, Kick the bucket, Let the cat out of the bag, Piece of the action, Pull someone's leg, Rule of thumb, Silver lining, Take a rain check, Take the bull by the horns, Under the weather, When pigs fly, Wild goose chase, You can't judge a book by its cover
Formula expressions (9)	How do you do?, It's raining cats and dogs, The whole nine yards, Let's call it a day, The early bird catches the worm, Break the ice, A stitch in time saves nine, It takes two to tango, Beat around the bush
Classification by origin (6)	Ad hoc, Catharsis, Avant-garde, Coup d'état, Feng shui, Yin and yang
Adjectives (19)	blue-blooded, high-tech, fair-haired, two-faced, easygoing, hardworking, big-hearted, well-off, narrow-minded, long-lasting, one-of-a-kind, old-fashioned, fast-paced, top-notch, high-spirited, short-tempered, well-known, easy-to-use, open-minded
Substantive (16)	the pot calling the kettle black, a safe bet, a skeleton in the closet, a square peg in a round hole, the straw that broke the camel's back, a swan song, the tail wagling the dog, the tip of the iceberg, a tough cookie, the milk of human kindness, a money pit, a loose cannon, a nest egg, the new kid on the block, a pain in the neck, a penny for your thoughts
Phraseological units related to animals (10)	Busy as a bee, Hold your horses, You can lead a horse to water, but you can't make it drink, A bird in the hand is worth two in the bush, Let the cat out of the bag, When pigs fly, Wild goose chase, It's raining cats and dogs, The early bird catches the worm, the tail wagling the dog, a swan song
Phraseological units related to the body (9)	All thumbs, Egg on your face, Fly off the handle, Get cold feet, Keep your chin up, All ears, Break a leg, Pull someone's leg, Up in arms
Not included in the highlighted classification features (14)	Catch-22, Diamond in the rough, In the same boat, Jump the gun, Leave no stone unturned, Money talks, On the ball, Pull yourself together, Saved by the bell, The whole nine yards, Vanishing point, Walking on eggshells, X marks the spot, Zero hour

Figure 6: Results of applying existing and proposed software to solve the research problem.

Thus, the author's software not only reduced the processing time for PhUs but also established additional regularities for the features by which these objects are classified.

During the research, a comparative evaluation of the application of the existing and proposed software was conducted on other datasets of PhUs. The classification results were found to be similar to those analyzed above. Furthermore, in each of the studied cases, regularities regarding the results and time of automatic classification, as presented in the analyzed variant, were observed.

6. Conclusion

Based on the results of the study, the following conclusions can be drawn:

- currently, one of the key problems in natural language processing is the recognition and classification of phraseological units;

- the task of automating the classification of phraseological units in English texts is relevant;
- existing systems for automating the classification of phraseological units contain a number of shortcomings that do not allow to effectively solve the problem of their qualitative classification;
- an urgent task is to develop an effective system for automatic classification of phraseological units in English texts that would be reliable, applicable to the processing of various sentence structures and different types of phraseological units including fixed expressions, idioms and phrases and would contain a minimum number of shortcomings;
- solving the problem of developing a system for automatic classification of phraseological units in English texts which is formulated in the article can increase the efficiency of classification of phraseological units in English texts;
- the theoretical basis for solving the problem formulated in the article can be the hybrid method proposed by the authors, the main idea of which is to use a rule-based method to identify and distinguish specific types of PhUs and further apply machine learning methods to classify PhUs based on their semantic and syntactic properties;
- an effective means of implementing the hybrid method can be a system whose structure includes the following modules: Hybrid Soft; tokenization; tagging; base determination; division; corpus module; classification module;
- it is advisable to use Python to develop a system for automatic classification of phraseological units in English texts that implements the hybrid method.

The direction of further research is to fully evaluate the effectiveness of the proposed system.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] I. Basaraba, "English-language phraseological units: the problem of classification." *Scientific notes of the V. I. Vernadsky Tavrichesky National University. Series: Philology. Social communications* 31 (70), no. 4 (2020): 1–8.

[2] I. Basaraba, O. Lemeshko, "Correlation of cognitive abilities and translation skills of phraseological units." *SKASE. Journal of Theoretical Linguistics*, Košice, Slovak Republic, 18, no. 2 (2021): 34-50.

[3] I. Basaraba, L. Borovyk, "Application of Linguistic and Statistical (Quantitative) Methods to the Research of the Idiomatic Space of Military Fiction." *SKASE* 21, no. 2 (2024): 141-160.

[4] I. Basaraba, "Challenges encountered in automatically classifying phraseological units." *Current issues of the humanities: interuniversity collection of scientific works of young scientists of the Ivan Franko Drohobych State Pedagogical University*, Drohobych, no. 75 (1) (2024): 145-152.

[5] L. Thompson, "Advances in Natural Language Processing Techniques." (Part of the publication: Conference Materials) In: *The Annual Conference of the Association for Computational Linguistics*, Vancouver, (2023), 22-35.

[6] Bishop, Christopher M., and Hugh Bishop. *Deep learning: Foundations and concepts*. Springer Nature, (2023).

[7] Sun, Wei, and Eunjeong Park. "EFL learners' collocation acquisition and learning in corpus-based instruction: A systematic review." *Sustainability* 15.17 (2023): 13242.

[8] 8. Eichstaedt, Johannes C., et al. "Closed-and open-vocabulary approaches to text analysis: A review, quantitative comparison, and recommendations." *Psychological Methods* 26.4 (2021): 398.

[9] Natural Language Toolkit. url: https://www.nltk.org/ (2023).

[10] Wang, Meng, and Fanghui Hu. "The application of nltk library for python natural language processing in corpus research." *Theory and Practice in Language Studies* 11.9 (2021): 1041-1049.

[11] Zong, Chengqing, Rui Xia, and Jiajun Zhang. "Information extraction." *Text data mining*. Singapore: Springer Singapore, (2021). 227-283.

[12] Chollet, Francois, and François Chollet. *Deep learning with Python*. Simon and Schuster, (2021).

[13] North, Kai, Marcos Zampieri, and Matthew Shardlow. "Lexical complexity prediction: An overview." *ACM Computing Surveys* 55.9 (2023): 1-42.

[14] Liu, Chen, et al. "FigMemes: A dataset for figurative language identification in politically-opinionated memes." *Proceedings of the 2022 conference on empirical methods in natural language processing*. (2022).

[15] Wankhade, Mayur, Annavarapu Chandra Sekhara Rao, and Chaitanya Kulkarni. "A survey on sentiment analysis methods, applications, and challenges." *Artificial Intelligence Review* 55.7 (2022): 5731-5780.

[16] Li, Qian, et al. "A survey on text classification: From traditional to deep learning." *ACM Transactions on Intelligent Systems and Technology (TIST)* 13.2 (2022): 1-41..

[17] Mardanova, Aziza. "Role of Phraseology in Developing Linguistic and Intercultural Communication Competences." *American Journal of Philological Sciences* 3.05 (2023): 68-72.

[18] Spinde, Timo, et al. "Automated identification of bias inducing words in news articles using linguistic and context-oriented features." *Information Processing & Management* 58.3 (2021): 102505.

[19] Miletic, Filip. "Bridging Across Datasets and Disciplines: The Contribution of Corpus Phonology to the Study of Lexical Semantic Variation." *Spoken English Varieties: Redefining and Representing Realities, Communities and Norms*. (2021)..

[20] Sung, Min-Chang, and Hyunwoo Kim. "Effects of verb–construction association on second language constructional generalizations in production and comprehension." *Second Language Research* 38.2 (2022): 233-257.

[21] Bozşahin, Cem. "Referentiality and Configurationality in the Idiom and the Phrasal Verb." *Journal of Logic, Language and Information* 32.2 (2023): 175-207.

[22] Dhar, Ankita, et al. "Text categorization: past and present." *Artificial Intelligence Review* 54.4 (2021): 3007-3054.

[23] Bardab, Saeed Ngmaldin, Tarig Mohamed Ahmed, and Tarig Abdalkarim Abdalfadil Mohammed. "Data mining classification algorithms: An overview." *Int. J. Adv. Appl. Sci* 8.2 (2021): 1-5.

[24] Lyu, Jinghui, et al. "A character-level convolutional neural network for predicting exploitability of vulnerability." *2021 International Symposium on Theoretical Aspects of Software Engineering (TASE).* IEEE, (2021).

[25] Maulud, Dastan Hussen, et al. "State of art for semantic analysis of natural language processing." *Qubahan academic journal* 1.2 (2021): 21-28..

[26] Venkateswaran, N., et al. "Study on Sentence and Question Formation Using Deep Learning Techniques." *Digital Natives as a Disruptive Force in Asian Businesses and Societies.* IGI Global Scientific Publishing, (2023). 252-273.