UNIVERSITETI I EVROPËS JUGLINDORE
УНИВЕРЗИТЕТ НА ЈУГОИСТОЧНА ЕВРОПА
SOUTH EAST EUROPEAN UNIVERSITY

DOCTORAL PROGRAM: E-Technologies

FACULTY: Contemporary Sciences and Technologies

START OF THE STUDIES: September 2015

LANGUAGE FEATURE IDENTIFICATION FROM LARGE TEXT COLLECTIONS

PhD Student: Diellza Nagavci (ID: dn16574)

SUPERVISOR: Prof. Dr. Mentor Hamiti

Abstract

Natural Language Processing (also known as NLP) is a branch of study that has become more significant in the modern day. The natural language processing (NLP) resources are quite helpful when it comes to building a machine that is capable of and every language has its own dictionary, which as a result is spoken by a large number of people understanding natural language or translating between linguistic pairs – a solution that aims to resolve the issues caused by language barriers.

Understanding and creating natural language on several levels, including syntax, semantics, pragmatics, and discourse, is the core goal of language processing. These levels have included not just the syntax of a language but also its organization, semantics, and pragmatics.

Tokenization, normalization, stemming, labelling words as parts of speech, and other processes are some of the phases that have been involved in Natural Language Processing (NLP). The field of natural language processing (NLP) uses many different methods and grammar rules, such as inflection, derivation, tenses, semantic analysis, lexicons, morphemes, and corpora.

Although low-resource languages frequently lack sufficient annotated data to utilize supervised techniques successfully for Natural Language Processing challenges, so this thesis has been concentrated on unsupervised learning approaches based on raw text learning.

In this Ph.D. thesis, we have created the vocabulary from low resources, which has been completed with the increase of resources, and have intended to develop part of speech tagging by using the Chinese Whispers algorithm. In addition to this, it analyses the approaches to the Albanian language that can be accessed via unsupervised learning techniques that may be obtained from raw text.

The results showed that increasing the number of sources can improve the quality of the vocabulary in languages with low resources, compared with manual addition, which had limited success. Also, unsupervised learning methods have been applied to large text collections to identify the language features, such as part of speech tagging for the Albanian language.

Results yielded in our study showed that large text collections can be effectively used to improve language feature extraction for low-resource languages such as the Albanian language. This study contributes to the field of Natural Language Processing by the development of an appropriate digital vocabulary for different languages with limited linguistic resources.

Keywords: Natural Language Processing, Machine Learning, Part-of-Speech, Algorithms, Chinese Whispers, Clustering

Contents

List of Figures

List of Tables

List of Abbreviations

NLP	Natural Language Processing
SVM	Support Vector Machine
k-NN	k - Nearest Neighbor
TF	Term Frequency
TF-ID	Term Frequency Inverse Document Frequency

1 Chapter

1. INTRODUCTION

Natural Language Processing (NLP) is a part of artificial intelligence (AI) which aims to teach computers to comprehend text and spoken language in ways similar to how humans do [1].
Machine learning, statistics, and deep learning methods combine computational linguistics with rule-based models of human can. This technology has enabled computers to understand human language regardless of whether it is written or spoken and to understand its full meaning, including the speaker's or writer's intent.

In addition to translating text between languages, NLP allows computer programs to reply to spoken commands and summarize large amounts of text in real-time. In addition to translating text between languages, NLP allows computer programs to reply in real-time, it can understand spoken commands and summarize enormous quantities of material. Furthermore, natural language processing (NLP) is having an increasingly crucial role in enterprise solutions that assist expedite company operations, enhance staff productivity, and simplify mission-critical tasks.

The addition of natural language processes helps to develop a well-organized system for managing words, chunks, sentences, and texts. Language processing techniques such inflections, derivations, tenses, sentiment analysis, vocabulary, lexemes, corpora, and morphologies are covered by NLP [2]. Machine learning may have been a useful method for identifying text data.

NLP researchers have achieved significant results in a wide range of areas. Nonetheless, in the vast majority of situations, these outcomes are based on unsupervised learning using manually verified and annotated text corpora. A large amount of effort is required to create these corpora in order for these results to be implemented in a specific language. As a result, languages with a large number of speakers or those that are effectively supported by major expenditures have much more resources, and NLP techniques perform better overall.

The lack of annotated data for low-resource languages makes supervised methods for NLP inefficient. Unsupervised learning methods that learn from the raw text are one of the approaches that may be evaluated in these instances. Training a model without pre-tagging or annotation is known as unsupervised machine learning. Nevertheless, the unsupervised techniques are clustering Latent Semantic Indexing (LSI) and Matrix Factorization [3].

The Albanian Language morphological features are divided into two variables: the first is a variable part of speech and the second fixed part of speech. Words that can be inflected and conjugated are known as variable components. Nouns, adjectives, verbs, and pronouns are all used. Adverbs, prepositions, and exclamatory particles are examples of fixed elements of speech that cannot be inflected or conjugated. In order to accurately label the word forms of a text extracted from a document, it is possible to use word-class information, which is useful for downstream processes such as dependency parsers [4].

One of the unsupervised methods that we expect will provide good results is random graph-clustering algorithms such as 'Chinese Whispers', for which Biemann et al. [3] shows significant results in other languages with similar features.

Chinese Whispers belongs to the class of graph partitioning methods with the lowest computing complexity: at the very least, although attempting to partition a graph, the graph itself must be taken into consideration, and the list of edges is the densest form of its representation [5]. This allows for the clustering of very large networks, which is important for unsupervised POS-tagging with enormous vocabulary sizes.

1.1 Problem Description

It is important or even necessary to have linguistic data for numerous applications to make communication easier or at least to provide support data for building further linguistic datasets as resources for natural language processing [6]. Due to its complicated grammar and inflection paradigm, Albanian is one of the most intriguing and challenging languages to learn.

The low number of Albanian language resources is also a gap for this thesis. We define a problem by analysing the research gaps. The purpose of this thesis is to simplify the literature review process and speed up the process of defining the research problem.

How can we identify the language features in texts with low resources?

Our study aim is thus to investigate how Natural Language Processing can help to build Albanian vocabulary by using methods and machine learning algorithms. We consider that the Albanian language is a good representative of such low-resourced languages. Although we can find a significant amount of written text, there are almost no annotated resources available.

In light of this, this thesis will seek to use unsupervised learning methods that can be applied to large text collections in order to identify different language features that are used by NLP tasks.

We aim to first provide basic statistics about the language, such as word usage frequencies, character, and word n-grams. in this case the Albanian language. Based on these statistics, by analysing usage context and frequencies, we will provide a method to generate a spell-check dictionary that will continually be enhanced when adding new text.

Furthermore, we will provide means to inject language-specific information, as well as manually annotated data in this model, in order to enhance the results. In Natural Language Processing, part-of-speech tagging is often a fundamental step. Corpus linguists and lexicographers will benefit greatly from the expanded search possibilities available with tagged data. Next, we want to evaluate the results of applying unsupervised Part-of-Speech (POS) Tagging to raw text.

We aim to use random graph-clustering algorithms such as 'Chinese Whispers' to provide good results, as they show significant results in other languages with similar features.

For unsupervised POS-tagging, large lexicon sizes are a crucial criterion for clustering very large graphs. We will also focus on integrating variously supervised and unsupervised learning techniques into our model. We will evaluate the techniques and extract the most efficient ones.

1.2 Hypotheses

When discussing the state of the art, we have seen that studies have made significant accomplishments in several Natural Language Processing fields, the most of which have been focused on supervised learning using manually evaluated and annotated text corpora.

Since low-resource languages generally miss sufficiently annotated data to efficiently use the supervised methods for Natural Language Processing tasks, this thesis will focus on unsupervised learning methods based on raw text learning.

Evidently, with Albanian language belonging to the low resources' linguistic group, the state of art has shown that there is deficient research and scholarly work regarding NLP for this language, which shares little word etymologies with other higher resources languages. On that note, this thesis will focus specifically on this research gap: by providing insights as to how low-resource languages can be catch-up with the mainstream language group in terms of utility. The following hypotheses seek to establish a working framework that can be applicable to other low resources languages as well.

Null: Large text collection can be effectively used to improve language feature extraction for low resources languages

H1: Word usage differences can be used to detect misspelled words in automatically built dictionaries

H2: Structural characteristics of text can contribute to dictionary completion

H3:H1 and H2 results help improve automated POS tagging in low resources languages.

If we start with a research question, we can quickly dive into our research, but if we spend time defining our aims and objectives, we will know what our research should be about. In addition to reducing the likelihood of problems arising later down the road, this will lead to more thorough and coherent research. Below is a list of the research questions that will be trained in this thesis.

1.3 Research Questions

1. Is it possible to generate spell-check dictionaries from the raw text?

- There is a lot of raw text in low-resource languages, so we can use them to generate spell-check dictionaries [63]. To develop unsupervised algorithms, the following questions must be answered:
- Which software should be used to generate the spell-check dictionaries?
- Are we going to have results from raw text?
- Which method should be used to generate the spell-check dictionaries?

2. Is it possible to detect misspelled words based on usage differences?

A few questions need to be answered regarding the unsupervised areas to detect usage differences in misspelled words in low-resource languages:

- How can usage differences in a textual context be used to detect misspelled words in a text collection?
- Is it possible to obtain accurate results in languages with low resources?

3. What algorithms can be used to find rarely used words?

Algorithms can be used in NLP to improve the efficiency of the best way to find rarely used words from the raw text in Albanian language, thus this paper will also focus on queries such as:

- From the many algorithms that are represented, which is the desirable type for used words?

4. What results can be obtained by applying unsupervised POS tagging to a large text collection in Albanian?

Part-of-speech tagging will be described in detail and evaluated directly and indirectly in Albanian language, and a clarification will be needed as per:

- Which number of POS categories are part of the method?

5. How increasing the text collection affects the accuracy?

Text collection models perform better based on the words used in the corpus and the classification features used. Therefore, questions such as:

- Which techniques can improve the results form text collection in Albanian language?

1.4 Methodology

As a result of our research, answers to the hypotheses and questions above will be provided. We will create a sizable corpus of Albanian-named entities which would help both in a rule-based and statistical named entity recognition systems. Likewise, processing text written in the Albanian language can lead to non-text symbols and other inaccuracies.

The process of tokenization, splitting each phrase of the text collection into tokens. In addition to words, characters and sub words can also be used as tokens. Tokenization can therefore be divided into 3 classifications: word tokenization, character tokenization, and sub word tokenization (characters in n-grams). Recognition of mistakes when the textbook may contain many words that use letters like ‘Ç’, ‘ç’, ‘Ë’, ‘ë’, they can be presented differently in the file, as a result, we have read the file proper encoding. To obtain tokens, the corpus is tokenized. After tokenizing the corpus, the vocabulary is prepared.

Experiments and the generated empirical results will be the base for our quantitative research methodology. Consequently, we applied unsupervised Part-of-Speech (POS) Tagging to raw text to evaluate the results of unsupervised learning algorithms. Unsupervised learning is essential for use as training resources. It also entails using advanced machine learning methods to train models without pre-tagging or annotating them, as well as giving them access to virtually endless amounts of data.

The goal of this thesis is thus to evaluate how Natural Language Processing can help to build Albanian vocabulary by using methods and machine learning algorithms. Furthermore, we have provided a statistical spell-check dictionary for the Albanian language by comparing usage frequencies, differences in usage between different sources, as well as morphological properties of words.

1.5 Thesis Structure

In the first chapter, we define the research question, hypothesis, and challenges of low-resource languages. Furthermore, supervised versus unsupervised learning is defined and distinguishable. In this chapter, we represent to the readers the thesis outline and the main goal.

The second chapter provides information about the main concepts and techniques used in the dissertation. It begins with an introduction to Natural Language Processing and the importance of low resources, followed by the other tasks of NLP, including Text Pre-Processing, Tokenization, Semantic Analysis, Stop Word Removal, Stemming, and Lemmatization. Furthermore, it includes the fundamentals of unsupervised learning and different algorithms such as Chinese Whispers.

The third chapter of the thesis discusses the background and state of the art in the field. We begin this chapter with a list of references used for state of the art. In the following section, various machine-learning methods are discussed. Here, we present an overview of Natural Language Processing techniques, their types, and their characteristics.

The final part of Chapter 4 discusses the contribution of NLP in linguistics. Tokenization is one of the NLP processes discussed in Section 4.1. Section 4.2 represents another process of NLP such as Normalization of the text for the Albanian Language. The morphology tagging of the corpus is explained in section 4.3, and annotations using the Universal Dependencies Scheme are discussed in section 4.4.

Afterward, unsupervised learning techniques are described, along with their applications to text. Furthermore, various natural language processing data representation models have been reviewed in terms of their current state of the art represented in the Chapter 5. Furthermore, presents the final matrix that was obtained by the computation of the cosine similarity from the previous matrix. This calculation was performed on the previous matrix. Using the vocabulary created in chapter 4, continued experimenting with part-of-speech tagging in chapter 5 with success identifying adverbs, adjectives, and nouns from the vocabulary. Although we have increased resources, we haven't achieved anything, so there is still room for others to contribute.

Presented in chapter six are the experimental results for the Albanian language. Natural language processing tasks for building a vocabulary for the low-resource language. Using these results, we will be able to provide certain conclusions and suggestions to the whole process.

1.6 Discussion

In this chapter, we presented the research problem that emerged from the literature review presented in Chapter 2. We saw that, language identification involves identifying the language in which a text document has been written. In addition, almost all languages have a dictionary, but those with low resources often lack it. Utilizing natural language processing enables users and communities to share information and services.

In Section 1.2, we have defined the hypotheses which are crucial in pursuing the research and defining their claim. There are null and three hypotheses which we elaborated during this dissertation through empirical and analytical experiments, in order to prove that large text collection can be effectively used to improve language feature extraction for low resources languages.

Moreover, this chapter includes five research questions that target various issues on spell-check dictionaries. In this section 1.3, we deal with the importance and benefits of solving this problem. This will be a great contribution to linguistics. Furthermore, we have stated that the outcomes of this research would be used to enrich this low-resource language, particularly since the Albanian language lacks a sufficient digital vocabulary.

2

Chapter

2.FUNDAMENTALS

2.1 Natural Language Processing

To process natural language, NLP integrates principles from computer science and linguistics with computing. In this situation, spell checking would be used to offer one or more options; moreover, there are accurate spelling alternatives when a misspelled term is discovered. Albanian is a low-resource language without a properly defined dictionary. As a result, NLP involves defining a dictionary of computational representations and analyses that are used to understand and generate text. A spell check dictionary is also a well-known problem in Natural Language Processing. Despite all the research, most of the best research has been conducted for different languages, meaning that there is still a lack of research for every language, particularly for low-resource languages such as the Albanian language.

Although there are limited annotated resources available for Albanian, we can locate significant amounts of written material. The Albanian lexicon has been proposed in a few research papers for Natural Language Processing, and it contains 75,000 entities, but it is still in development. The development of Named Entity Recognition in Albanian using deep learning is proposed. The LSTM (long short-term memory) cells are used, along with the CRF layer. Manual annotation was performed using place, person, and organization names. The accuracy of the annotation would improve with a larger corpus. Additionally, the creation of a public annotated Albanian corpus would greatly improve the accuracy of named entity recognition.

Since there are no publicly available Albanian annotated corpora, we will create one based on linguistic features, frequency data, n-grams, and morphological tools from the lexicon.

A statistical spell-check dictionary for the Albanian language can be created using Natural Language Processing (NLP) by evaluating usage frequencies, differences in usage between various sources, and morphological properties of words. However, most of the resources used for training are not publicly available, making it impossible to compare results or build upon existing methods.

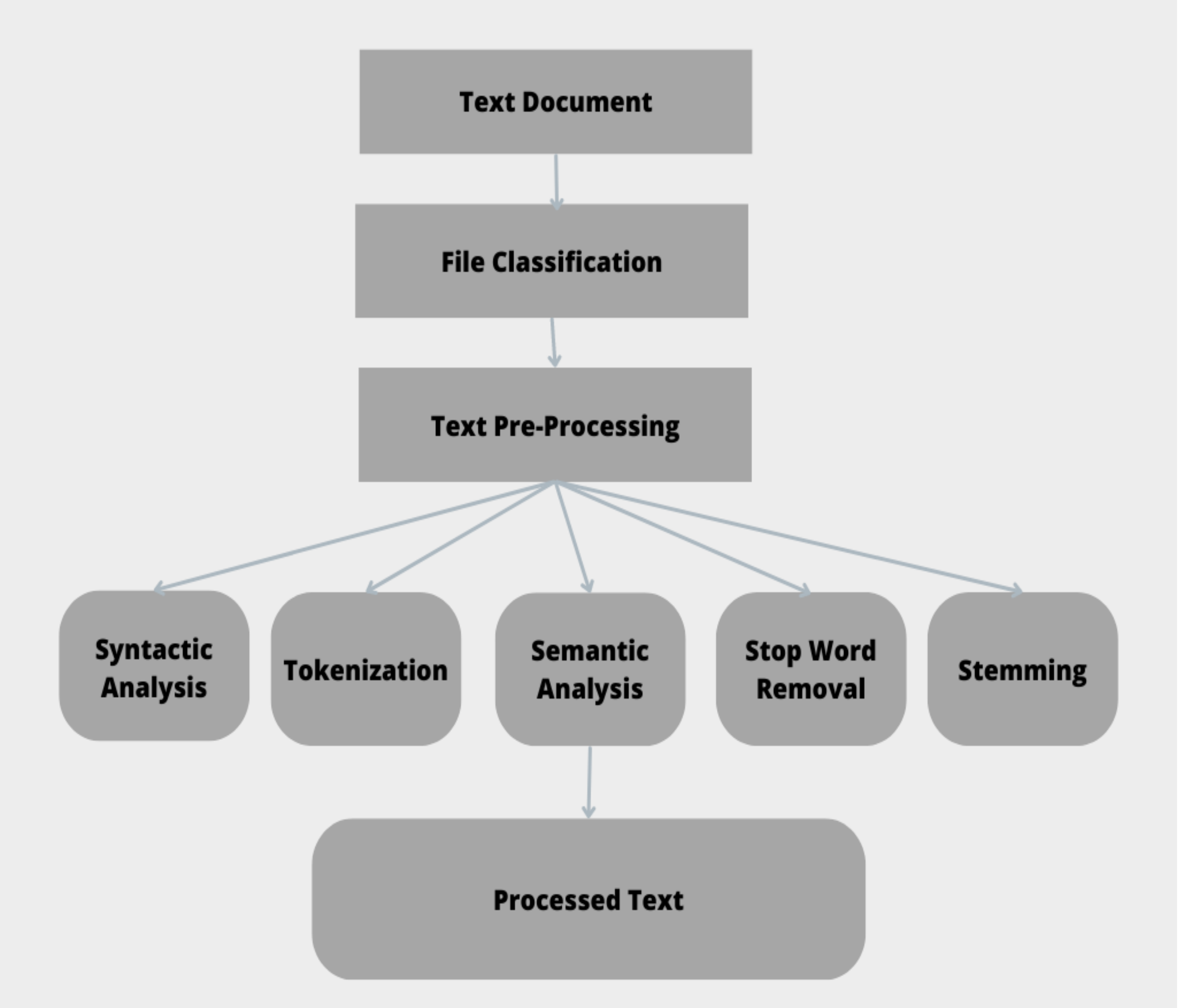


Figure 1. Text pre-processing model**.**

A growing area of computer science, natural language processing takes advantage of machine learning and computational linguistics. Interaction between humans and computers is primarily concerned with making it simple but efficient [1]. It learns the syntax and meaning of human language, processes it, and outputs it. Natural language processing involves making computer systems make sense of human-understood natural language.

As a result of natural language processing, we are going to be able to build models and processes that will take chunks of information as input, whether it is in the form of text, voice, or both, and manipulate them within the computer according to the algorithm.

An automatic text summarization algorithm based on templates is applied in two phases: the pre-processing and the information extraction phases. The text processing model is shown in figure 1 below.

Syntactic analysis: Syntactic analysis, often known as parsing or syntax analysis, is the third step of NLP. The purpose of this phase is to identify the exact meaning of the text, often referred to as dictionary meaning. Using the rules of formal grammar, syntax analysis determines whether the text is meaningful.

An input document's syntactic analysis module determines where each sentence begins and ends. Currently, the algorithm takes a full stop at the ending of the sentence. Any string of characters up to the full stop symbol is considered one complete sentence.

As a result of these findings, morphological analysis, often known as parser, is the process of evaluating natural language symbol characters in line with formal grammar rules. The term "parsing" is derived from the Latin "pars," which means "portion."

The task of parsing is carried out by a parser. It is a system software designed to take data and convert everything into a morphology provided as input data using formal grammar. A data structure is also constructed, which is usually in the shape of a parse tree, specific data tree, or even other hierarchical structure. Among the libraries available for use in completing the mission.

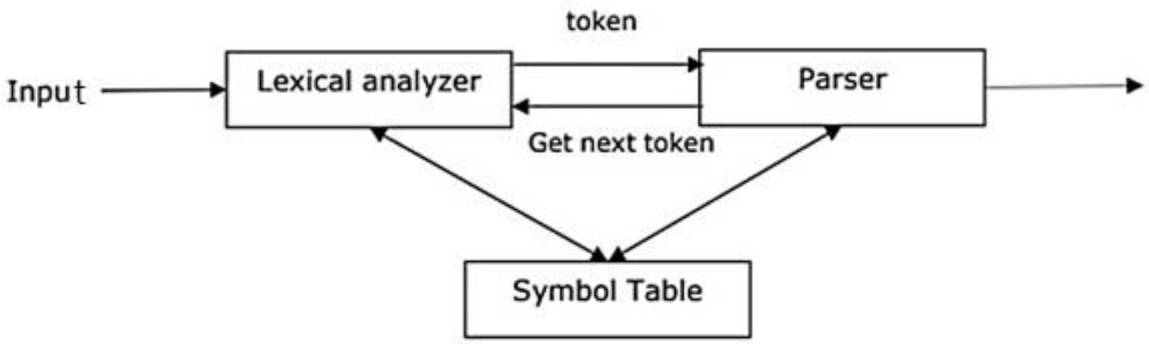


Figure 2. Concept of Parser[1]

Parsing has the following main roles:

- Reporting syntax errors.
- The purpose of this function is to recover from common errors and continue processing the remainder of the program.
- It creates the parse tree.
- It creates the symbol table.
- It produces intermediate representations (IR).

2.2 Text Pre-Processing

The pre-processing part mainly involves information retrieval and feature extraction. Different methods have been suggested: terms, a bag of words, term frequency, term frequency-inverse document frequency, enhanced TF-IDF models, sequences of words, and others.

The pre-processing of text consists of a series of steps that are applied to each source of Albanian Language.

These steps include:

1. Stopping word removal using the sci-kit-learn same as in English dictionary,
2. Removing all special characters (numbers, hyphenated words, apostrophes words)
3. Remove all single characters,

[1] Taken from https://tinyurl.com/yckxe82h [accessed 22.04.2020]

4. Substituting multiple spaces with a single space,

5. Lowercase,

6. Uppercase,

7. Tokenization

2.2.1 Tokenization

Tokenizers are responsible for breaking the sentence into tokens given as outputs from the syntactic analysis module. Broken sentences can be words, numbers, or punctuation marks [2]. Tokenization may be achieved using a variety of techniques and tools. Most of the other libraries which can be used to finish the work include NLTK, Gensim, and Keras.

Tokenization can be performed either on words or sentences. Word tokenization is the process of separating words from text using a separation technique, while sentence tokenization is the process of separating sentences in the same way. Figure 3 shows a tokenization example.

Tokenize() is a module in NLTK that is further divided into two categories:

- Tokenize the words in a sentence by using the word_tokenize() method
- Another method to separate a document or paragraph into tokens, is the sent_tokenize() method

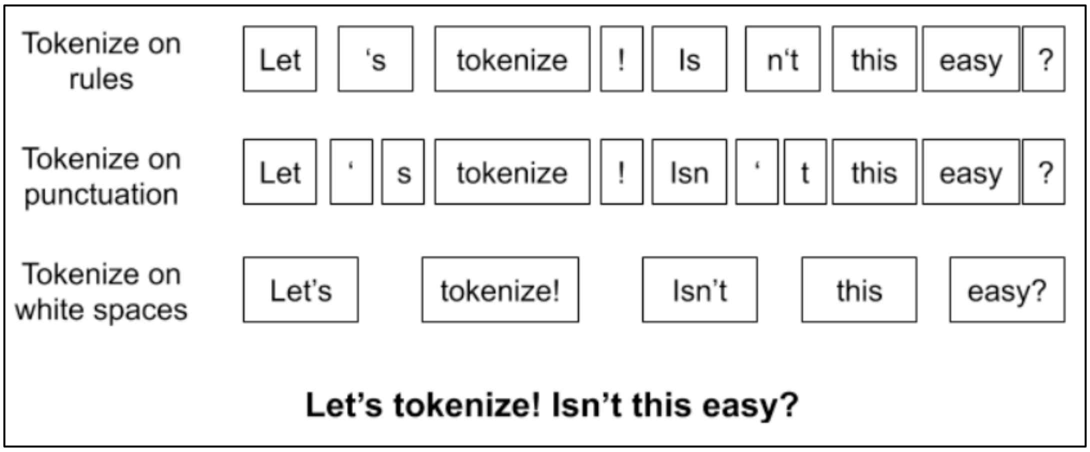

Tokenize on rules
Let | 's | tokenize | ! | Is | n't | this | easy | ?
Tokenize on punctuation
Let | ' | s | tokenize | ! | Isn | ' | t | this | easy | ?
Tokenize on white spaces
Let's | tokenize! | Isn't | this | easy?
Let's tokenize! Isn't this easy?

Figure 3. Tokenization in action[2]

[2] Taken from https://blog.floydhub.com/tokenization-nlp/ [accessed 16.06.2020]

2.2.2 Semantic Analysis

The process of finding meaning from text is called semantic analysis. Computers study the sentence's grammatical syntax and identify specific words and their relationships in order to better understand and interpret phrases, paragraphs, or complete texts.

Semantic analysis is used to identify the exact meaning of a text or the dictionary definition of that text. A conceptual analyzer determines the meaning of a document. Machine translation, chatbots, search engines, and sentiment analysis are all examples of technologies that apply semantic analysis.

Lexical analysis is part of semantic analysis. Therefore, the meaning of the word is studied through lexical semantics. It identifies the relationships between lexical items. Hyphens, synonyms, antonyms, and homophones are some of the relations among words.

Furthermore, presented are details about the relations:

- Hyponymy: So, it is a connection between occurrences of a generic word. A hypernym is a generic term, whereas a hyponym is an occurrence.
- Homonymy: A collection of words with the same spelling but distinct meanings.
- Polysemy: A polysemy is a term or phrase with a different meaning that is comparable. Despite their similar spellings, polysemy has a different meaning.
- Synonymy: The association between two elements that, despite their differences, convey the same or similar meaning.
- Antonymy: A semantic connection between two lexical elements with symmetric components along an axis.
- Meronomy: A rational combination of words and letters denoting a single part or component of something.

The program can discern the context of any phrase or paragraph by recognizing these linkages and compensating for symbols and punctuation marks.

2.2.3 Stop Word Removal

Stop - words include phrases that do not add much meaning to the statement. As a result, they may be safely ignored by the corpus without jeopardizing the sentence's meaning. When considering the overall meaning of the sentence, certain words result in a significant increase than others in natural language text, but overall usefulness to extract meaning is minimal. We mark these words as stop words and remove them. Stop-words are words that have no significance in the text and, if eliminated, have no effect on how the text is processed for the purpose specified. They are deleted from the lexicon to minimize noise and the size of the feature set.

2.2.4 Steaming

Using a text processing model, stemming involves evaluating the basic form of words in an input document. The same words are written in different tenses, yet all have the same meaning. To avoid this, stemming is done, so that words with the same meaning, but in different tenses, are converted into basic simple tenses.

Stemming is a component of morphological and artificial intelligence (AI) information retrieval and extraction in linguistics. We can extract useful information from enormous sources such as big data or the Internet using stemming and AI expertise, because there may be more variants of a term linked with a concept which need to be explored. The stemming technique is also employed in inquiries and Search engines.

2.2.5 Lemmatization

Lemmatization is the process of removing suffixes and applying rules to get the valid root/lemma word. However, in order to design the lemmatization, certain rules for the removal and insertion of suffixes were devised, as well as a knowledge base of unusual terms. If an input word matches one of the exceptional lists, the lexicon - based result is much like the input text; alternatively, the word must be treated according to the intended rules. The knowledgebase, on the other hand, requires a lot of memory space, but it is possible to extract an exact root word and it delivers a quick result.

2.3 Unsupervised Learning

Unsupervised Learning promises to learn efficiently from unlabeled data (no labeled data is required for training). This is a substantial benefit over Supervised Learning because unlabeled data in electronic information is plentiful, but labeled datasets are often expensive to develop or acquire, especially for common NLP tasks like PoS tagging or Syntactic Parsing.

Unsupervised Learning models are pre-programmed with all of the intelligence and automation needed to function autonomously and automatically in order to discover information, structure, and patterns in data. Unsupervised NLP may now shine as a result of this.

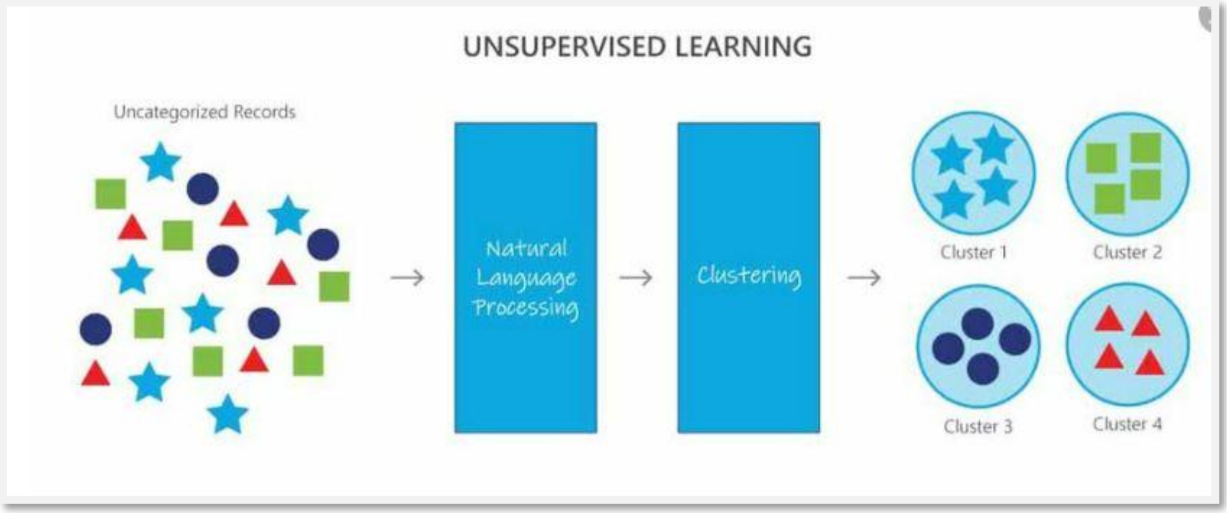


Figure 4. Unsupervised Learning[3]

Clustering (including such K-means, Mean-Shift, Density-based, Spectral clustering, and so on) and association rules approaches are the most prominent uses of Unsupervised Learning in sophisticated AI Chatbots / AI Virtual Assistants. Clustering is a technique that is widely used to automatically group semantically related user utterances together in order to accelerate the derivation and verification of underlying shared user intent. Unsupervised Learning is frequently used in association-rule mining, which would be the process of directly discovering correlations between attributes in data.

2.3.1 Chinese Whispers

Graph-clustering algorithm Chinese Whispers offers randomized graph clustering with a time-linear number of edges. Chinese Whispers is assessed on Natural Language Processing (NLP) tasks including such language separation, learning of syntactic word classes, and word meaning identification [33] after a thorough description of the system and an assessment of its strengths and weaknesses. In NLP, the small-world property is used for many graphs.

[3] Taken from https://medium.com/@rohithramesh1991/unsupervised-text-clustering-using-natural-language-processing-nlp-1a8bc18b048d/ [accessed 09.02.2021]

Below is presented the algorithm’s outline:

```
initialize:
for all vi in V: class(vi)=i;
 while changes:
for all v in V, randomized order:
 class(v)=highest ranked class
in neighborhood of v;
```

Figure 5. The Chinese Whispers algorithm

The following are some NLP experiments with graphs derived from natural language data. The first challenge, which used language separation and word meaning, would have been to split a multinational corpus by language, presuming tokenization in phrases. Using Chinese Whispers-partitioning, the graph was split into monolingual parts. Throughout the identification of languages based on the words, these units function as word lists. They obtained a near-perfect performance evaluation on differentiating 7-lingual corpora with equitized portions and strongly skew combination of two languages

Chinese Whispers: Label Weighting

Typical strategies to weigh the labels in the neighborhood Gu of u in G:

- Sum of the edge weights corresponding to the label i (top):

$$\text{weight } (G_u, i) = \Sigma_{\{u,v\} \in Eu:\text{label}(v)=i}\, w(u, v)$$

- Use the node degree deg(v) to amortize highly weighted edges (nolog):

$$\text{weight}(G_u,i) = \Sigma_{\{u,v\}\in Eu:\text{label}(v)=i}\ \frac{\text{w(u,v)}}{\deg(v)}$$

- Use log-degree for amortization (log):

$$\text{weight}(G_u,i) = \Sigma_{\{u,v\}\in Eu:\text{label}(v)=i}\ \frac{w(u,v)}{\log(1+\deg(v))}$$

The technique that can split the combined graphs into their prior portions may be measured unsupervised to evaluate word meaning. Bordage et al. [10] defined four measures: retrieval precision (rP), retrieval recall (rR), precision (P), and recall (R). The test was designed to compare the findings to those of Bordag, who used a triplet-based hierarchy graph clustering approach.

However, the approach was chosen solely for its applicability for unlabeled data: lacking linguistic preparation, which includes tagging or parsing, only the disambiguation mechanism, rather than the efficacy of the preprocessing techniques, is investigated. They have presented him with test 1 (word classes individually) and test 3 results (words of different frequency bands). Data was obtained from the raw text of BNC, and 45 test words were evaluated.

Table 1. The percentage of disambiguation depends on the word class (nouns, verbs, adjectives)

%	Bordag				Chinese Whispers			
POS	P	R	rP	rR	P	R	rP	rR
N	86.0	**85.7**	90.8	64.1	**89.9**	79.4	**94.7**	**71.2**
V	**77.3**	64	80	55.1	77.5	**67.0**	**87.2**	**57.8**
A	88.5	**72.0**	87.0	64.4	**92.1**	61.8	**89.2**	**71.8**

As can be seen in tables 1 and 2, both algorithms achieve about average performance (P) prediction and (R) recall. As a result of the identical data, the Chinese Whisper method clustering yielded the same information as the specialized algorithm for word sense induction. The much better performance on (rR) retrieval precision and (rP) retrieval recall suggests that CW clusters have fewer words, which might be advantageous when employing the clusters as indications in word meaning disambiguation.

Table 2. Disambiguation results in % dependent on frequency

%	Bordag				Chinese Whispers			
Freq	P	R	rP	rR	P	R	rP	rR
Hight	93.7	**78.1**	90.3	**80.7**	93.7	72.9	**95.0**	73.8
Med	**84.6**	**85.2**	89.9	54.6	80.7	83.8	**91.0**	**55.7**
Low	**74.8**	49.5	71.0	41.7	74.1	**51.4**	**72.9**	**56.2**

As both a consequence of either the NLP data conversation, Chinese Whispers significantly outperformed both these classification algorithms that choose the number of courses on their own and can handle clusters of varying sizes, making it suitable for NLP difficulties in which class distributions are frequently excessively skewed and the classification is known ahead of time, such as WSI (Word Sense Induction).

Different graphs are offered for such acquiring of word classes: the 2nd graph on surrounding co-occurrences. By calculating co-occurrences, the graph is constructed, progressing crucial word pairings based on their closeness as immediate neighbors. An example of a bipartite graph is shown in figure 4. If two identical words appear in both portions, they will produce two different nodes. By evaluating the variety of similar rights and left neighbors for two words, this graph depicts the transformations into a second-order graph.

The sum of common neighbors is the similarity between two words that includes the edge weight. Figure 5b shows the second-order graph produced from Figure 6a, as well as its division by Chinese

Whispers. The word "drink" (to drink the drink) is an instance of a word-class ambiguous word that is accountable for all intra-cluster interactions [4]. The implication is that words with a large number of neighbors should be detected with much the same POS and receive high weight in the second chart. Figure 15 shows three clusters that correlate to distinct portions of speech (POS).

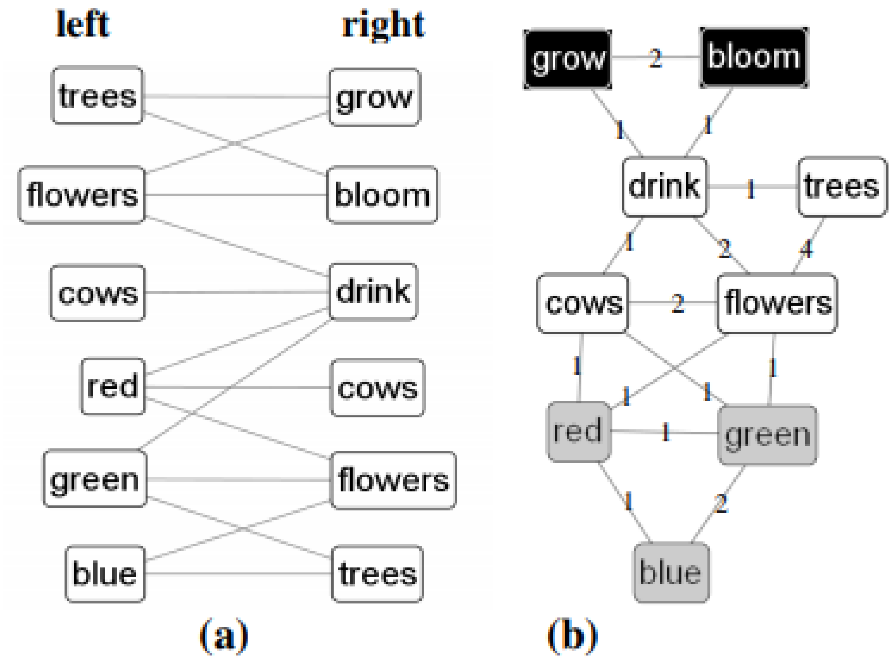


Figure 5. Chinese Whispers are grouped in a bipartite adjacent co-occurrence network (a) and a second-order graph on neighbouring co-occurrences (b).

Another large-scale test is performed in the British National Corpus (BNC), that either excludes the highest 2000 words, computes the second-order similarity graph, and draws connections between words if they have at least four left and right neighbors. Each cluster is checked against a lexicon that includes the most frequent tag for each word in the BNC. The most significant clusters are shown in table 3 below.

Table 3. The biggest clusters obtained after dividing the 2nd order graph with Chinese Whispers

size	tags:count	sample words
18432	NN:17120 AJ: 631	authorities, transportation, unemployment, farm, municipality, woods, procedure, grounds, …

4916	AJ: 4208 V: 343	busy, drab, little, thin, adequate, appealing, vital, …
4192	V: 3784 AJ: 286	filled, disclosed, experienced, learnt, pushed, happened, …
3515	NP: 3198 NN: 255	Black, Yellow, Jones, Hill, Brown, Lea, Lewis, Old, ...
2211	NP: 1980 NN: 174	‘Ian’, ‘Alan’, ‘Martin’, ‘Tony’, ‘Prince’, ‘Chriss’, ‘Emma’, ‘Hanrey’,’Cara’, …
1855	NP: 1670 NN: 148	Central, Leeds, Manchester, Australia, Yorkshire, Belfast, Glasgow, Middlesbrough

Among all, Chinese Whispers formed two clusters, with 26 of them exceeding the size of 100. Furthermore, cluster clarity has a weighted average rate of 88.8 percent, and that is the amount of dominating tags divided by a number of clusters. The 88.8 percent precision much outperformed the 53 percent precision on word type observed by Biemman et al. [9] on a comparable test.

2.3.2. TF-IDF

For information retrieval, the TF-IDF model is extensively employed. The vector models may be generated to use these word occurrences without depending on any specific ordering. During the preprocessing step, we evaluated two approaches:

1) TF (Term Frequency) and
2) TF-IDF (“Term Frequency, Inverse Document Frequency).

The importance of a word (phrase) in a text is determined with how often it appears in other papers. The IDF (inverse document frequency) determines how much information is included in a single word. Divide the total number of papers N by its number of papers containing the phrase *i* to get the answer.

The following equations (1) is used to compute TF-IDF:

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{dfi}\right)$$

(1)

For a word I in a text j, use the following TF-IDF:

- tf i,j = number of times I appears in j
- df I = the number of documents that contain I, N = the overall number of documents.

It is feasible to tokenize texts, understand the vocabulary and inversion articles weighting factors, and encrypt additional documents using TfidfVectorizer. The table below shows how to use TfidfVectorizer to effectively acquire lexicon and inverse text frequency across three small Albanian papers and then encode a few of those texts. In Section 3.2.4, examples are provided.

2.3.3 K-Means Algorithm

Clustering is a type of unsupervised learning method, and it also plays a significant role in natural language processing. K-means clustering is a partitioning clustering method commonly used in data mining based on a specified value of K, this algorithm partitions N documents into K clusters.

The true K value for some models can be found by using heuristic approaches discussed below, or even by users of the division [3]. Hence the true K will use to partition our N documents into K different classes in which each document belongs by using some similarity, the same cluster must be similar to each other and dissimilar from the limitations of the K-means reduces the summation of the square distance between data points and those cluster centres. The calculation steps for the K-means clustering method are given below.
Assign K cluster centers to the initial cluster

$$a1(1), a2(1), a3(1) \dots ak$$

The data [X] should be distributed in K clusters after k iterations, the following relation can be used

$$X \epsilon Cj\ (K)\ if\ \|x - aj\ (k)\| < \|x - ai\ (k)\|$$

(2)

On each 1,2,3,4, … ,K; $i \neq j$; where Cj (k) is the collection of data elements where cluster centers are aj (k). Calculate the new center aj (k + 1),j = 1,2,3, … ,K by minimizing the sum of squared distances to an initial cluster center from all locations in Cj (k).

The part that works to minimize distance is simply the mean of Cj (k). Therefore, the new cluster center is calculated as follows:

$$aj\ (k + 1) = 1\ N\ \sum x \in C\ x, j = 1,2,3, \ldots ,K\ j\ (k) \tag{3}$$

While the N j stand for the No. of samples in Cj (k).

If aj (k + 1) = aj (k) for j = 1,2,3, … ,K

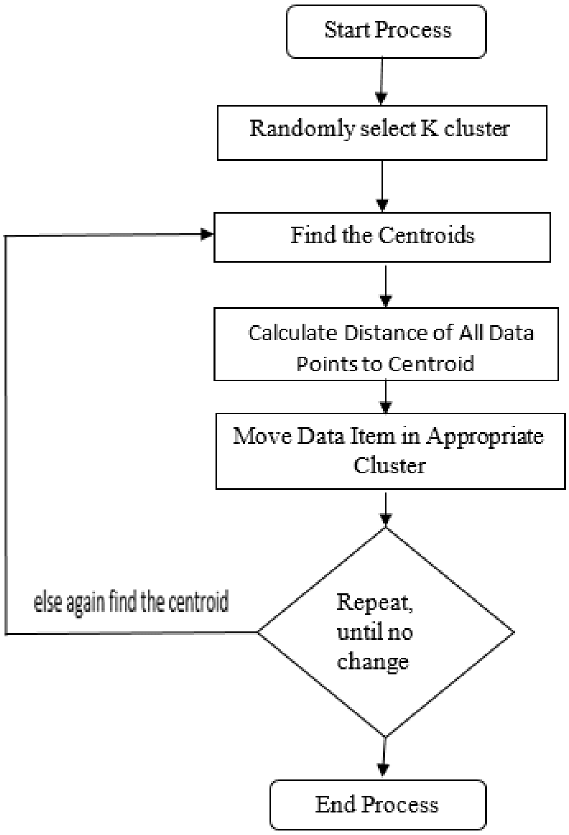

Start Process
Randomly select K cluster
Find the Centroids
Calculate Distance of All Data Points to Centroid
Move Data Item in Appropriate Cluster
Repeat, until no change
else again find the centroid
End Process

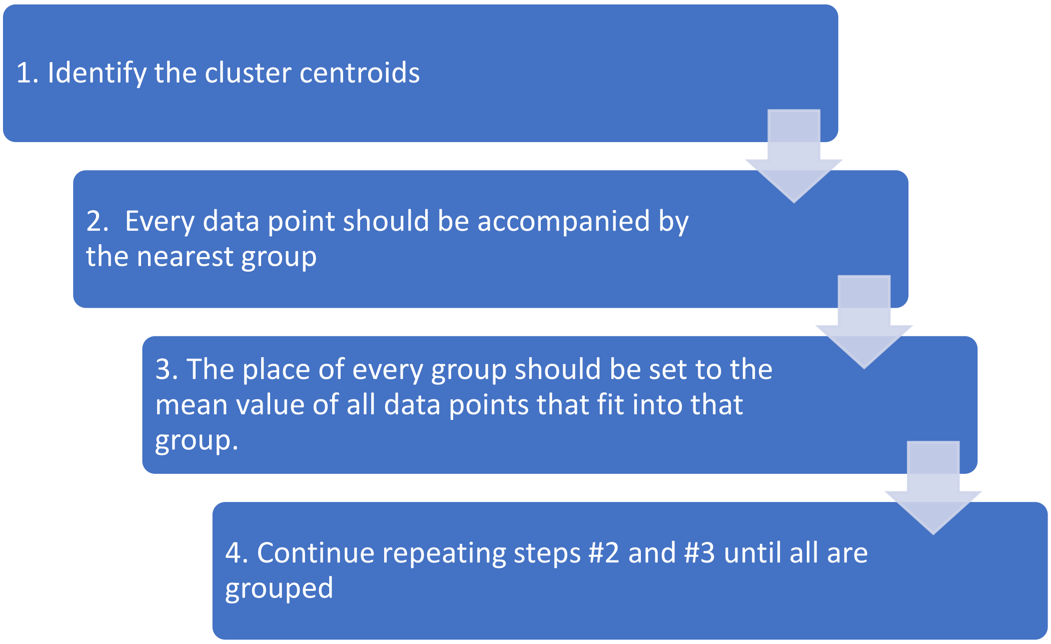
Figure 6. K-means Algorithm Process

Then the algorithm halts due to converged action, otherwise repeat step (b). As a result of this process, it is obvious that the final clustering results are always influenced by the initial seed and the true value of K, but initial seeds and the true value of K present in the data set require previous knowledge that is not always available or practicable. K-means algorithm process is shown in Figure 6.

2.3.4 K-means Clustering

The K-means algorithm is a center-based clustering method in which the most representative point is chosen. A representative point is chosen for one cluster, and the distance between it and all other points is calculated for all other points. A K-means algorithm is called such because one can choose the number of clusters to use. In other words, K representative objects (i.e. centroid or medoid) are present. The object is then assigned to the closest centroid and, therefore, the related cluster. The center point is updated based on the objects that have been added next in the cluster. New objects added to the cluster are added to the centroid. The process repeats itself until the centroids don't change, and then the process is finished. At worst, the algorithm will converge after k·n iterations [4].

Steps of the K-Means algorithm are represented in the Figure 7 below:



1. Identify the cluster centroids

2. Every data point should be accompanied by the nearest group

3. The place of every group should be set to the mean value of all data points that fit into that group.

4. Continue repeating steps #2 and #3 until all are grouped

Figure 7. Continue repeating steps #2 and #3 until all are grouped

Despite its widespread use in practice, the K-Means algorithm has some disadvantages:

- It is highly sensitive to initialization,
- Outliers are a concern,
- It can only deal with clusters with symmetrical point distributions, and
- K must be defined at the beginning of the process.

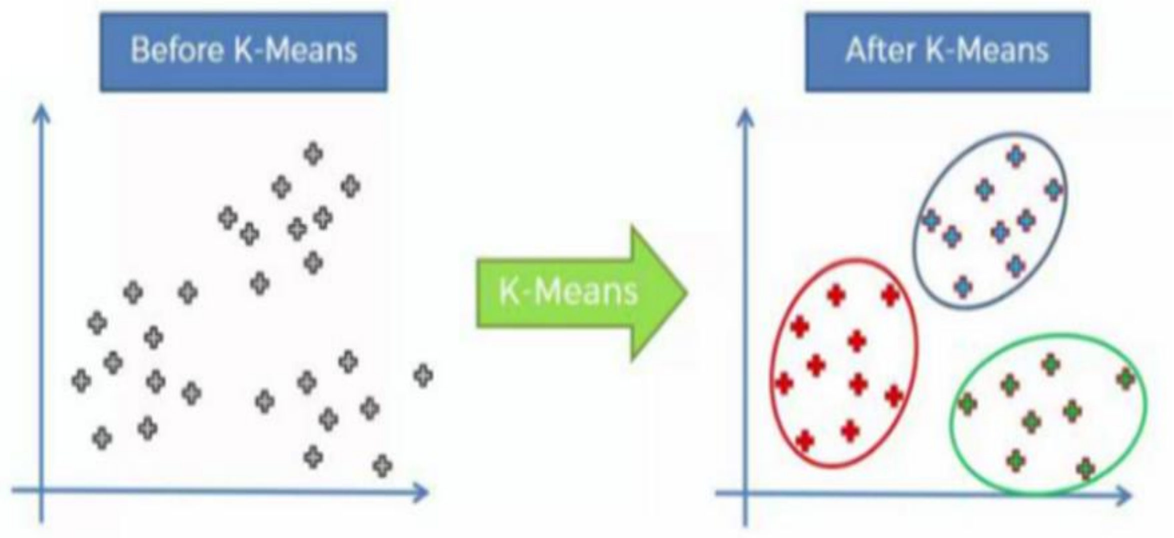


Figure 8. Example of 3 centroids, K=3

A crucial part of this algorithm is finding the optimal number of clusters. The elbow method is commonly used to find optimal K values. We vary the number of clusters (K) in the Elbow method from 1 to 10. WCSS (Within-Cluster Sum of Squares) is calculated for each value of K.

The summing of the squared differences between every position and the centroid is WCSS.

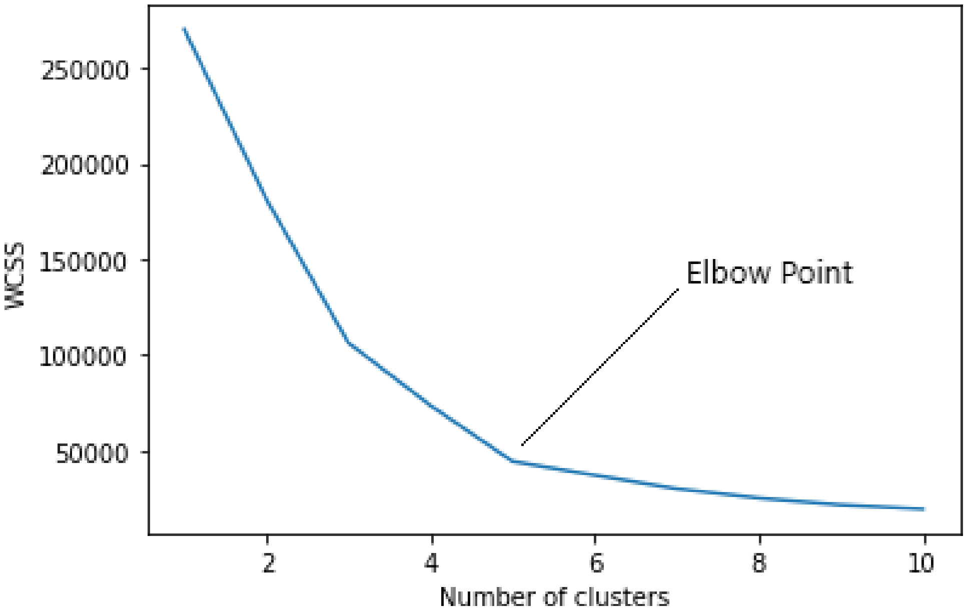


Figure 9. Elbrow method

The WCSS plot with a K value resembles an elbow. The WCSS value decreases as the number of clusters increases. When K = 1, the greatest WCSS value is obtained. We can observe from the graph there is a quick shift at a point, resulting in an elbow shape. At this point, the graph begins to travel practically parallel to the X-axis. The ideal Value of k, or the optimum number of clusters, corresponds to this point.

3

Chapter

3. LITERATURE REVIEW

After analyzing papers in the related discipline, a research gap was evidenced that the Albanian language belongs to the low resources linguistic group. In light of such findings, this thesis will use an unsupervised POS-tagging system that was described briefly above and would be further expanded in the paper; and evaluated directly and indirectly on Albanian language and tasks.

Additionally, based on assessing and cross-referencing linked publications, develop and address the research questions.

Following a thorough search, the most appropriate and relevant articles are chosen, and the categorization scheme is created. On that basis, the research questions are answered based on the results of the mapping as well as the overall conclusion of the systematic review procedure. This strategy is ideal since it frequently gives a visual summary, such as a map, of its findings [7]. Initially, this thesis seeks to collect all relevant articles pertaining to the subject of interest. Simultaneously, an overview of this study field has to be presented, identifying the number, type of investigation, and accessible outcomes.

Table 4. Search Strings

No.	**Search String**	**No. of papers**
SS1	((("Abstract": Natural Language Processing OR " Abstract": Machine Learning) AND "Abstract": Language Identification)	205
SS2	((("Natural Language Processing ") OR "Unsupervised") AND Machine Learning) AND Language Identification	132
SS3	((((("Natural Language Processing ") OR" Unsupervised") AND Machine Learning) AND Language Identification) AND prediction	23

The goal of this thesis's state of the art is to assess publications that have addressed Natural Language Processing predicated on a few research issues such as:

- What is the main area of focus addressed in the articles?
- What type of methods were used regarding Natural Processing Learning?
- How have publications evolved over time?
- What are the current trends in research and publication?
- Which algorithms can aid in the identification of seldom-used words?

Most of the research articles utilized during cross-referencing as well as analysis in this study were obtained through library resources such as IEEE-Xplore and ACM, with some articles being obtained via Springer Link. The search terms indicated in Table 1 were used to run the searches inside the digital libraries described above.

Different search strings have been proposed in most articles. Based on that, we selected only the ones that were considered more relevant to the topic of this paper (shown in Table 4). The majority of the articles chosen have been published in recent years. Tab. 5 presents the number of recent publications (during 2015 to 2019) - again, focused on those relevant to our study. Further analysis is performed on the selected publications, selecting only those that are relevant to NLP and Machine Learning, Unsupervised and Language Identification. As a consequence, after deleting duplicates but also irrelevant papers, there were only 125 relevant publications left.

The classification scheme is provided in three columns, each of which represents one of the research's key fields of interest (Fig. 10). Machine learning will be an important topic of study in the future, which would also ultimately lead to the production of vocabulary for low-resource languages using unsupervised POS-tagging other methodological approaches. The first column in the scheme defines the field of interest, which includes Machine learning, NLP, and Ontology as areas of interest.

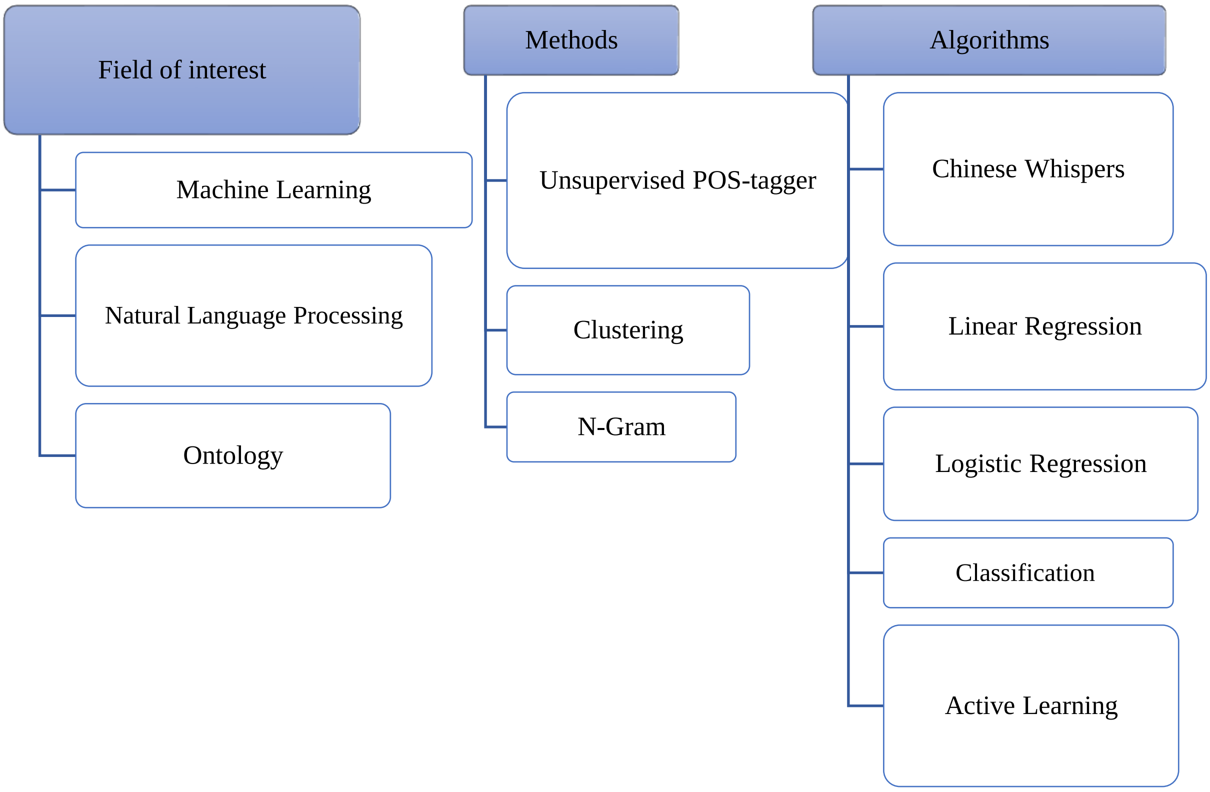


Figure 10. Classification scheme

In order to offer responses to such five research topics, the selected research articles were divided into separate groups. The next paragraphs provide the systematic review study's research questions and subsequent responses.

The unsupervised learning may then be utilized to aid enhance automatic POS-tagging in low-resource languages, as illustrated in the second column. Finally, throughout the third column, several Machine Learning Algorithms – such as that of the Chinese Whispers method – that will be employed in low-resource languages are mentioned. Based on the data analysis of the gathered publications, the gap was identified in relation to 'low-resource languages,' to which such a study may make a significant contribution

RQ1: What is the main area of focus addressed in the articles?
This question focuses on the major area of interest investigated for each of the publications. We're interested in Natural Language Processing, however, because we applied many search terms, we obtained a variety of results. We devised the 'Field of Interest' categorization for the papers to solve this question.

Table 5 shows that machine learning is the major emphasis of around 54 percent of the studies [9], along with other approaches that have been used.

Natural Language Processing is the next most stated topic of interest, accounting for about 37% of all mentions.

Semantic role labeling, spatial expression recognition, feedback, topic linkage, and visualization plug-ins are examples of publications in this field. It has a wide range of additional important applications, including OCR, parser, natural language comprehension, named entity identification, and machine learning [8].

Table 5. Number of Papers by Main Field of Interest

Field of interest	Number of papers	Percentage
Machine Learning	**68**	**54%**
Natural Process Learning	**46**	**37%**
Ontology	**11**	**9%**

RQ2. How have publications evolved over time? What are the current trends in research and publication?

We analyze the year of dissemination for each publication, paying special attention to the time periods around 2015 and 2019. The bulk of the chosen papers (44.80%) was published in 2018. In actuality, when we look at the figures in Fig. 11, we can see that the publishing rate is increasing year to year, showing rising interest in the topic.

Table 6. The number of articles published each year till the first quarter of 2019.

Year	Numbers of Papers	%
2019	24	19.20%
2018	56	44.80%
2017	18	14.40%

2016	17	13.60%
2015	10	8.00%

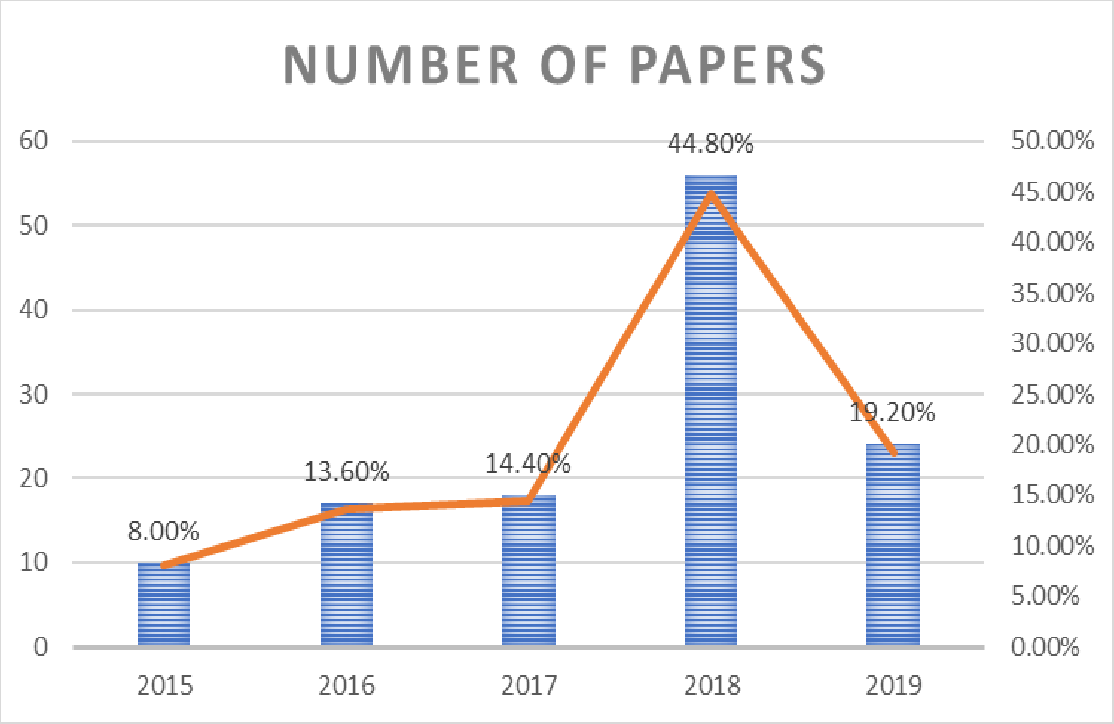


Figure 11. The number of papers per year

RQ3. What type of methods were used regarding Natural Processing Learning?

Over the past few years, NLP (Natural Language Processing) has been used to overcome many language difficulties. Dipanjan's research on NLP highlighted the utility of graph-based labeling in mapping part-of-speech information across multiple languages [11].

These findings show that reliable POS taggers can be trained using languages that lacked annotated information but have translations more into a resource-rich language.

The research also points to a preference for unsupervised POS tagging, while also related strategies that employ direct projections to fill the gap among both strictly supervised and unsupervised part of speech tagging models [9]. According to the findings of this research, (please see Tab. 7) the majority of the selected studies (61 percent) were focused on approaches connected to Unsupervised Part-Of-Speech tagging, while the remaining (39 percent) dealt with Clustering.

Table 7. A number of papers by framework type

Methods Type	Number of papers	Percentage
Unsupervised POS- tagging	76	61%
Clustering	49	39%

RQ4: Which algorithms can assist in the discovery of seldom-used words?

The search for the rarest words in low-resource languages is a vital part of our research. In terms of Natural Language Processing, Biemann [3] use the Chinese Whispers technique, which is a very simple approach for partitioning the nodes of coupled, undirected networks. The Chinese Whispers approach is based on Natural Language Processing (NLP) difficulties such as language separation, syntactic word-class acquisition, and word meaning disambiguation.

Rahamn et.al [11] present document clustering in the Urdu language using an unsupervised clustering algorithm. This algorithm works with training datasets and is domain-independent. They have used a dataset of 1000 documents in the Urdu language and each document contains a different number of sentences and tokens using the K-means algorithm. For the data pre-processing they have presented each document as a bag-of-word model, this makes good use of Term Frequency Inverse Document Frequency for the documents clustering and it describes the frequency.

They preserved the comparability of each text after pre-processing in the form of a numeric value ranging from 0 to 1. Consequently, these data are inserted into the K-means algorithm as input clustering, and the output is compared to the manually categorized clusters. The outcome of clusters may be found in five distinct approaches to similarity metrics. The estimate of Cosine Similarity, TFIDF, Levenshtein distance, and Jaccard Co-efficient is 0.78, 0.44, 0.61, and 0.59 in tab 8.

Table 8. Result of Similarity Measures after Pre-Processing

Number of Clusters	**Similarity Measures**			
	Cosine Similarity	TF-IDF	Levenshtein distance	Jaccard Coefficient
Cluster 1	0.55	1	0.9	0.95
Cluster 2	0.45	0.2	1	1
Cluster 3	0.95	0.65	0.15	1
Cluster 4	0.95	0.25	1	0
Cluster 5	1	0.1	0	0
Average	0.78	0.44	0.61	0.59

Apart from the TF-IDF measure, the results of this experiment show that each similarity metric has a significant influence on Urdu document clustering.

Marenglen et al. [10] employed several algorithms in their research articles, such as Naive Bayes, SGD, Logistic, HyperPipes, and RBFNetwork, to evaluate the effectiveness of classification algorithms and opinion mining in a multi-domain corpus in the Albanian language.

They developed 11 text corpora of Albanian written thoughts culled from several well-known Albanian publications. For each corpus, the amount of text documents classed as positive responses and the amount of textual information categorized as bad reviews is the same.

Additional entity recognition systems were developed and tested using known machine learning methods including decision trees, but also neural networks given by Georgios et al. [14]. These systems were assessed inside the Greek text corpus, and they contribute to the identification of the drawbacks and constraints imposed by the examined algorithms whenever used towards natural language data.

This category includes a new approach called inductive grammar learning. The capacity to handle textual input, as well as the possibility of employing learned phrasal verbs in actual systems, substituting manually produced grammars, are the primary advantages of this technique above other machine learning techniques [11]. An innovative method has been developed for using induction grammar learning, which exclusively learns grammar using positive cases.

This new algorithm can infer context-free grammars and is based upon the existing algorithm-GRIDS [12], enhancing both the user-friendliness and the algorithm is proposed in the space of potential grammars, hence boosting the new method's applicability to larger collections of data.

Karl et al. [13] first proposed anchor-NMF. This learning system is used to handle unsupervised POS tagging. The purpose of such a challenge is to activate the proper sequence of POS tags (hidden states) based on a word sequence (observation states). The anchor condition inside the system of each POS tag relates to the assumption that at least one phrase that occurs is discovered under that tag.

Piton et al. [14] must have benefited greatly from the application of machine learning algorithms for numerous natural language processing activities, particularly data that can be collected from materials published in several languages. The researchers noted also that NooJ s graphs-morphological graphs as well as syntactic grammars-are especially successful for computerized Albanian language processing.

Researchers explored the connection among elements in sense and units for form, as well as the words inside one linear section of text. The word combinations were derived from several forms throughout the Albanian language. As a result, they discovered that morphemes are commonly concatenated, juxtaposed, or contracted. This same conjugated grammatical of NooJ enables the creation of flexible form vocabularies. Tools are required to identify words created by simple conjunction or to handle with contractions because of the variety of word structures. We may develop grammar methods to describe and address these events using NooJ's morphological tools.

Rashiti and Damoni was using the Levenshtein method in the Albanian language. The above approach, known colloquially as the Levenshtein distance, seems to be essentially an algorithm for adjusting the distance between the two places that can also be used for text conversion [15]. Distance proximity between words is calculated by dividing the difference between the letters in those words. The Levenshtein approach appears to work well enough for letters and numbers only with one letter, such as the English alphabet; but, it fails to compute the necessary distances for alphabet letters with two fundamental letters, including the Albanian alphabet. The Albanian language features nine double characters, including "dh," "gj," "sh," "th," "ll," "rr," "nj," "xh," and "zh." As a result, a new strategy has been suggested in certain situations.

In another research study, Raufi discussed a smartphone development approach and the integration of mobile applications using machine learning algorithms in real-time mobile settings. The goal was to detect and prevent hate speech and objectionable language [16]. The results showed that the machine learning techniques are applicable in mobile situations and have a high degree of classifier accuracy. Using machine learning under this type of text, analytics is also highly recommended because the necessity to answer requests/responses fast is significant in the mobile 'world.'

Despite being evaluated in a basic feed-forward network library; the neural network algorithms are also used to provide quite good results. Artificial Neural Network (ANN) approaches are utilized to aid in the identification of inflammatory or hateful language. Whatever expression, communication, action, or text that endorses, threatens or provokes violent acts is commonly referred to as hate speech. Many organizations use prominent social media networks, blogs, and communities to promote online radicalization and violence, offering a broad technique that relies on a text documents classification system.

For the Albanian language, Skënduli presented the Named Entity Recognition (NER) approach. NER is focused on recognizing individual, regional, organizational, and other entity categories in the raw text. Their model was built using the greatest entropy method [4]. A few manually annotate corpora, frequently on social and historical issues, before training models to develop classifiers capable of recognizing essential items in the text. Affected by a lack of Albanian corpora as well as the fact that it's the first NER study for the Albanian language in almost a decade, the results demonstrate how powerful the models could be with a larger training corpus.

Supplementary information POS tagging, which allocates each word to the appropriate part of speech. Because it's difficult to assign specific parts of speech to input text based on context, POS tagging is indeed a fundamental difficulty in NLP. Part of Speech tagging is one of the most powerful features of the NLTK module. This activity entails labeling district elements of electronics text input, and it has a significant influence on Natural Language Processing.

Moreover, the statistical POS tagger is used to accomplish automated tagging in the Albanian language. With morphologically rich languages, POS tagging confronts several challenges. Either rule-based or neither statistical strategy appears to be appropriate effective POS tagging for morphologically rich language. The rule-based method is based on good language knowledge, while the analytic approach focuses on a large number of corpora.

The introduction of a learning algorithm to a range of NLP tasks, notably the arduous issue of extracting knowledge in multilingual texts, has made linguistics a lot better [17].

Part-of-speech tagging for the Albanian language is presented by Hasanaj [18] and consists of a small tagset of 16 tags and a large tagset of 326 tags. Three tags for delimiters, two for unusual situations, one for research papers, and ten for common elements of speech make up the primary tagset. The large tagset encodes word classes, such as JJ [=Adj.], NN [=Noun], VB [=Verb], and PR [=Pronoun], as well as extra properties, such as Number (Sg. and Pl.). PRDFSE, which means PR. dem[onstrative] fem[inine] pl[ural] nom[inative], might be a tag from the huge tagset. Cross-validation and a sample text were used to test the tagger model.

Baxhaku [19] describes the research he carried out using open-source toolkits. The results indicate that such original Cavnar and Trenkle (2016) language recognition system fails to distinguish Albanian in txt files (its accuracy was only 16.5%). The primary goal of this research work is to determine which strategy and configuration would produce more effectiveness in a dataset that matches Albanian text texts published on the internet. Then it would provide for a more targeted crawl of the "Albanian Web."

According to studies, short texts (news item titles in their instance) have a detection rate of approximately 95% and lengthy texts have a detection rate of more than 99 percent. For written news stories, such a study has concentrated on both the Standard and Gheg dialects of Albanian. As well as naive Bayes, in-grams have also been used as classification features in both tools.

In the Albanian languages alphabet letters "Ë" and "Ç" as "E" and "C" have a significant impact on the accuracy of the tested tools, so they have demonstrated that misreading the Albanian language. This has been especially true for shorter works (accuracy dropped by 20-30 percent). Another experimentation with a custom-built training corpus that includes random (with a chance of 0.5) misspelling variants of these letters was also conducted. Langid.py was trained on this corpus and obtained an efficiency of more than 99 percent both for Standard and Gheg Albanian.

In conclusion, there have been several attempts to create annotated corpora and to apply POS tagging methodologies to the Albanian language. However, training resources are often not openly available, making it impossible to compare outcomes and improve on current approaches. Another issue with POS-tagging is the lack of a tag-set standard to utilize, despite the fact that the tag-sets supplied mostly by the Universal Dependencies project [20] may be deemed acceptable, with the added benefit of being equivalent to other languages.

Considering the great challenges of creating a vocabulary for any language, this thesis attempts to enrich this low-resource language with its results, especially since Albanian is one of the most linguistically diverse languages without a digital vocabulary, therefore, in such a case, it would be useful. There are, however, some limitations to supervised learning that can be overcome by using unsupervised methods to identify language features in text collections. Additionally, the corpus will be open for researchers in this field to continue experimenting and contributing to NLP.

4

Chapter

4. PROCESSING RAW TEXT COLLECTIONS

Open-source programming languages such as Python, Java, or C++ are used to build NLP applications. Natural language processing supports can be developed using a variety of tools in various combinations. Many of the best tools available for natural language processing use Python, and there are many alternatives in every category, from string pre-processing to sequence tagging to machine learning libraries. There are fewer tools available for working with Java as a programming language, but the tools are still as effective as those for Python.
Python has recently become the most used programing language in NLP.

We have chosen Python using Natural Language Toolkit (NLTK) libraries to build our vocabulary in the Albanian Languages. The Natural Language Toolkit was developed by the University of Pennsylvania in 2001 as open-source software as a collection of Python libraries for natural language processing. After the text is mined from file input, NLTK splits it into words and finds semantic meaning based on the split text. String processing, part-of-speech tagging, classification, chunking, and parsing are some of the main features [5]. Since then, it has been developed and expanded by dozens of contributors. As a result, it has been incorporated into dozens of university courses, and it has served as the basis for many research projects. Figure 5 compares the ratings of NLTK, CoreNLP, SpaCy, and others.

Using nltk.FreqDist as a module of Natural Language Toolkit (NLTK) is very helpful to find words used more often with frequencies. So, these statistics should be derived from the list of tokens produced by the same tokenizer. When we count how often words are used in a corpus, we compile a frequency list that we will build as a spell-check dictionary. When one derives a frequency list from an electronic text in the Albanian language, the tokens are counted and each distinct type with its frequency count is added to the list.

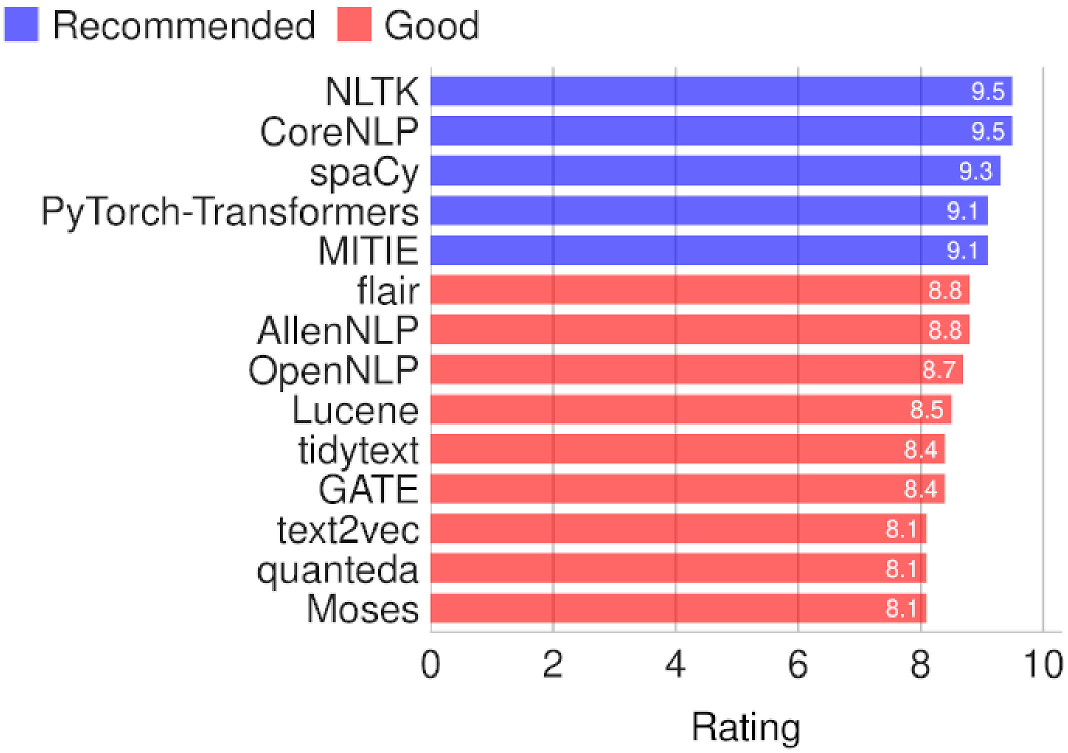


Figure 12. The best free NLP Tools rating[4]

The word types that make up the list determine the vocabulary as observed in the corpus. The list then contains all the word forms checked. The word forms retain their inflections in a frequency list; they are not typically reduced to their lemma, the word form under which the various inflected forms would fall in a dictionary. The vocabulary contained in a corpus should not be taken to constitute the language's vocabulary: there may be many words in the language that happen not to be in the corpus at all, however large this corpus is.

Tokenization: The major goal of tokenization is a process that can split the text into words and sentences. A linear succession of signals, letters, phrases, or sentences constitutes electronic text.

[4] Taken from [10.2021] https://www.linuxlinks.com/naturallanguageprocessing/

Tokenization is a type of pre-processing in the sense that it identifies the fundamental chunks that will be processed.

Naturally, the text must be separated into language items or pieces including letters, punctuation, numerals, and roman numbers before any serious text processing can occur. Additionally, one of the difficulties in the tokenization of text, especially for Albanian, is the file encoding of ASCII versus UTF-8. Letters such as 'të', 'ç', 'Ç', 'Ë', will be presented differently if the file is not read with the right encoding.

Various types of apostrophes exist, such as ‘‘, ', which might affect how we handle words like 't'i, ç'është' etc. Furthermore, we have different types of hyphens such as -, –, −, which may affect how we handle words like 'tekniko-juridike', 'indo-evropiane', 'shoqërore-ekonomiko', etc.

Table 9. Example of character definitions

Category	Characters
Alpha	a b c ç d e ë f g h i j k l m n o p q r s t u v x y z
Alpha capital	A B C Ç D E Ë F G H I J K L M N O P Q R S T U V X Y Z
Numeric Roman Numerals	0123456789 XX XXI
Sentence end	.?!
Punctuation	,: “‘() [] <>
Hyphen	-

Words in the Albanian Language are speared with hyphens at the end of lines. Another challenge is that we will handle words such as 'ç'është', 'ç'rast' as one word or two words, but it makes sense to consider them as two words 'ç' and 'është'. Further, in the Albanian language is to omit the space, which is between the characters forming a single word, e.g., 'teknike-juridike', which we can handle as 'teknike-' and 'juridike', because especially that a lot of people will write this word incorrectly as 'tekniko-juridike', where 'tekniko' is not a real word. In addition, there are titles written in uppercase in the text, which we will see how to deal with to some extent in the later phases.

Ta: no capital letters (Wa1), the initial letter capitalized (Wa2), all capital letters (Wa3), combined cases (Wa4), and so on. • numeric Tn: Wn1 (single digits), Wn2 (numbers with periods or colons), and so on. • commas (Wp3), brackets and quote marks (Wp2), only one sentence-internal marking (Tp), etc. • Tm mixtures: beginning with a hyphen (Wm1), ending with a hyphen (Wm2), beginning with a hyphen (Wm3), including slashes/hyphens (Wm4), containing numbers (Wm5), comprising roman numbers (Wm6), etc. These fundamental forms are straightforwardly allocated to tokens, using Table 9's categorization of characters into separate groups.

Moreover, it is critical to focus just on fundamental analysis or creation while ignoring fundamental pieces. However, without these fundamental chunks obviously segregated it is impossible to carry out any analysis or generation.

The identification of chunks that do not need to be further decomposed for subsequent processing is an extremely important one. Mistakes made at this step are very likely to induce more errors at later stages of text processing and are therefore very dangerous. The tokenization process is illustrated in Figure 13.

A set or a group of strings is represented by Regular Expressions for NLP. When there is a pattern in a string, regular expressions can be used to obtain, substitute, and execute a range of other string manipulation actions. Regular expressions may be used with practically every major programming language because they come with their very own compilers [24].

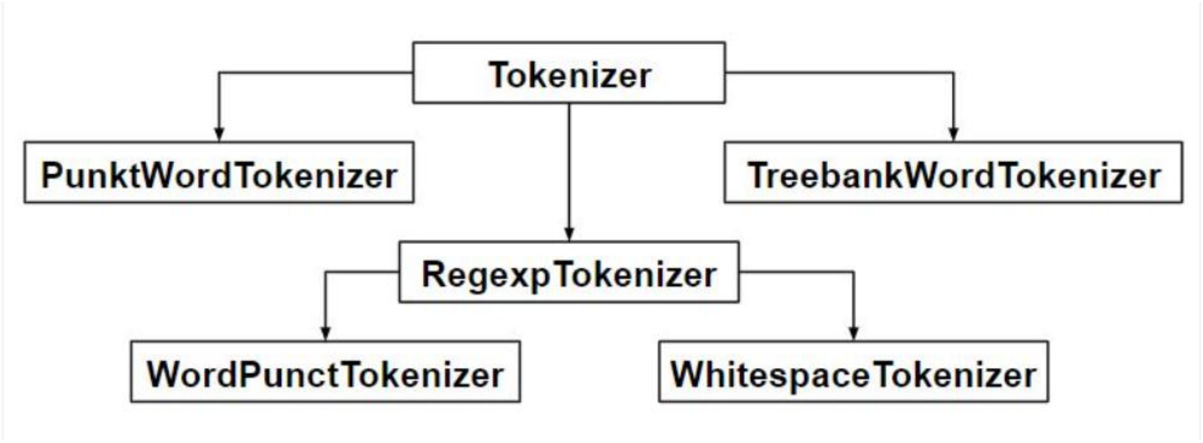


Figure 13. Tokenization[5]

We utilized the Regular Expression 'RegEx', which allows you to see if a text contains the search pattern you specify. Because the Albanian language contains many special characters, we used several Regular Expression algorithms developed in Python to find all the words that did not contain any special characters and punctuation.

Common Regex Functions: There are several special characters that are used as quantifiers, e.g., ‘?’, “*”, “+’”, “(‘, ‘)”, “{“, etc. can also appear in the input text for the Albanian Languages. In these cases, we have used escape sequences to extract these specific characters. Breakout letters, represented more by slash '/', are being used to bypass particular character meanings. We used '?' in the regular expression to match a question mark (this is defined as trying to escape the character) and '+' in the regular expression to match a plus sign. To avoid the punctuation mark from the '/' character, which is really a special character, we utilized the pattern '/'.

4.1 Building Dictionaries

The usage of labelled datasets is one of the key distinctions between supervised and unsupervised machine learning. Using examples of what the computer should look for and how it should assess these characteristics in supervised machine learning, a collection of textual data is annotated. These papers are being used to build a statistical model, which would be subsequently presented with untagged text to examine. If the model learns more about the documents it analyzes later, it can be

[5] Taken from: https://bit.ly/3rHcGQu [accessed 15.08.2019]

retrained with larger or better datasets. Unsupervised learning models, in contrast to supervised learning, work independently to find the structure of unsupervised learning. Validation of the output still requires human participation. The contrasts among supervised machine learning, as well as the techniques utilized in each, are depicted in Figure 14.

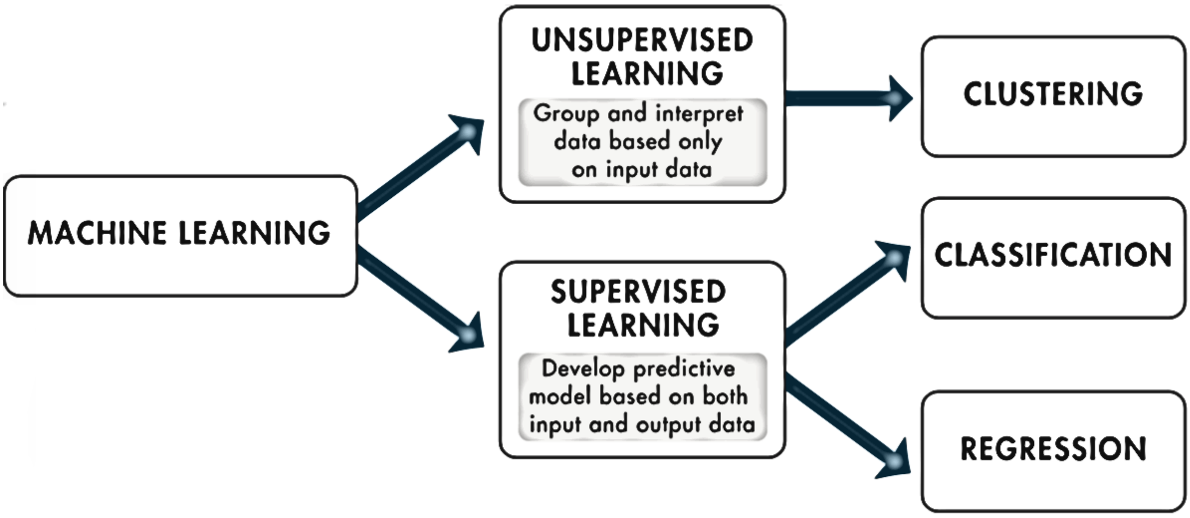

MACHINE LEARNING
UNSUPERVISED LEARNING
Group and interpret data based only on input data
SUPERVISED LEARNING
Develop predictive model based on both input and output data
CLUSTERING
CLASSIFICATION
REGRESSION

Figure 14. Unsupervised and supervised machine learning: differences and similarities

Unsupervised learning is sometimes preferred over supervised learning for various reasons. The following are some of the advantages:

- Data labelling by hand is time-consuming and costly. Unsupervised machine learning entails learning data and categorizing it without the use of labels.
- Labels can be added after the data has been classified, making the process much easier.
- It can be used to find patterns in data that are impossible to find through other methods.
- Unsupervised learning can be used to reduce dimensions.
- Unsupervised machine learning is ideal for data analysts since it may assist them in comprehending raw data.
- As the model learns slowly and then calculates the result, it is similar to human intelligence in some respects.

The approach is to implement our thesis using an unsupervised method. First, we have used initial resources; A portion of the text is taken from 20 books, but there are 13 authors including one author from two books. With the author's consent, these books are being taken for educational purposes [6]analyzed. The resources will be added as sources in the following text. In Table 10, we list the authors and years of publications from different fields published by South East European University from five different faculties (Economics, Business, Law, Computer Science, etc.), so that all studies are represented fairly.

Table 10. List of analyzed texts

NO	AUTHOR	TEXTBOOK	PLACE AND YEAR PUBLISHED
1.	Prof. dr. Asllan Bilalli dhe Prof. dr. Hajredin Kuçi	'Zanafilla, zhvillimi historik dhe burimet e së drejtës ndërkombëtare private'	Tetovë, 2006
2.	Dr. Asllan Bilalli dhe dr. Hajredin Kuçi	'Kolizioni i ligjeve (Konflikti ndërkombëtar dhe interlokal i ligjeve)'	Gostivar, 2006
3.	Prof. dr. Abdylmenaf Bexheti	'Një dekadë e mendimit të ri ekonomik dhe politik'	Tetovë, 200
4.	Prof. dr. Abdylmenaf Bexheti	'Teksti Universitar-Financa Publike'	Tetovë, 2006
5.	Dr. Etem Aziri	'Sociologjia e partive politike'	Tetovë, 2006
6.	Dr. Nexhbi Vejseli	'Ekonomia e ndërmarjes Economics of the Firm, Microeconomics'	Tetovë, 2006

7.	Dr. Sc. Blerim Reka- Mr. Sc. Arta Ibrahimi	'STUDIME EVROPIANE'	Tetovë, 2004
8.	Dr. Mustafa Ibrahimi	I'nterpretime dhe studime gjuhësore'	Shkup, 2003
9.	Dr. Mustafa Ibrahimi	'Folklori shqiptar në regjionin e Pellagonisë'	Shkup, 2002
10.	Grup autorësh: S. Xhaferi, M. Ibrahimi dhe B. Ymeri	'Emigracioni në Rumani'	Shkup, 2004
11.	Dr. Hamit Xhaferi	'Kahe Letrare '	Shkup, 2005
12.	Dr. Hasan Jashari	'Sociologjia e arsimit'	Shkup, 2005
13.	Mustafa Spahiu	'Me buzëqeshje u dal përballë'	Shkup, 2005
14.	Riza Lahi	'Riza Lahi-poezi'	Shkup, 2005
15.	Grup autorësh	'Emigracioni'	Shkup, 2003
16.	Shazie Hoxha	'Vashat e malësisë'	Shkup, 2005
17.	Shazie Hoxha	'Kur sytë flasin'	Shkup, 2005

18.	Shazie Hoxha ‘Dashuria e paharuar’ Shkup, 2005

Tokenization - the technique of collapsing a textual piece into smaller parts such as words and sentences - is the initial stage in text analysis. Tokens are single entities that form sentences and paragraphs. We have combined the statistics for all sources into one file, which shows the words that are recognized as words without any special characters, such as ‘.’, ‘-’, ‘\’, ‘&’, ‘%’, ‘(‘, ‘)’, ‘”‘, ‘+’, ‘_’ etc.
NLP uses the frequency process to determine how often words or phrases appear in a document. Depending on each document's length, a term may appear more frequently in longer documents than in shorter ones.

The statistics in table 11 show some commonly used words out of 631,008 that were used more frequently in all textbooks in the Albanian language for this research. The correct word with the highest number of appearances in all sources is ‘të’ – appearing in 49,929 out of 631,008 words, with a frequency of 7.92%; the word ‘e’ has 37,165 appearances and a frequency of 5.89%; the word ‘në’ has 19,577 appearances and a frequency of 3.10%, the word ‘drejtës’ which has appeared 626 times and has frequency 0.10%, the word ‘ndryshme’ has appeared in all sources 628 and has a frequency of 0.10 %.

The word ‘juridike’ has 620 appearances in sources and has 0.10 % of the frequency, the word ‘Evropian’ has 575 number of appearance and has 0.09% of the frequency and the word ‘ndërkombëtare’ has 482 and the frequency of 0.08 %; the words like ‘rëndësishme’ has appeared 280 times in source and has the frequency of 0.04%; the words ‘ekziston’ and ‘jetën’ have 281 number of appearance and 0.04% frequency, the word, ‘Kështu’, appeared 279 times with 0.04% frequency from 631,008 the total number of words.

Table 11. Frequencies and appearances of 200 words used more frequently

TOTAL WORD COUNT			**631.008**
WORD	**Appearance**	**Frequency**	
të	49992	7.92%	
e	37165	5.89%	
në	19577	3.10%	
drejtës	626	0.10%	
	……..		
ndryshme	628	0.10%	
juridike	620	0.10%	
evropian	575	0.09%	
ndërkombëtare	482	0.08%	
	……..		
rëndësishme	280	0.04%	
ekziston	281	0.04%	
jetën	281	0.04%	
Kështu	279	0.04%	

In all sources we have other words that have less appearance such as ‘mënyra’, ‘partisë’, ‘shtëpi, ‘sistem’, which have appeared in all sources 183 times and have a frequency of 0.029 % each. Words like ‘këta’, ‘lehtë’, and ‘Mirëpo’ have appeared in all sources 177 times and have a frequency of 0.028 % each. Words like ‘bashku’, ‘Do’ and ‘jote’ have appeared in all sources that we have tested 155 times and have a frequency of 0.025% each. During the testing of all sources we have found that words such as ‘fjala’, ‘konkrete’, ‘krijuar’, ‘mundur’, ‘nevojshme’, ‘tretë’, and ‘vete’ have appeared 141 times, and have a frequency of 0.022% each.

Words like ‘elementet’, ‘rastin’, ‘shpenzime’ and ‘thellë’ have appeared 133 times and have a frequency of 0.021% each. Words like ‘fëmijët’, ‘jashtme’, ‘mori’, ‘sistemin’, and ‘zhvillimit’ have appeared in all sources 127 times and have a frequency of 0.02% each. The words ‘drejtata’, ‘materiale’, ‘pari’, ‘person’, ‘personale’, ‘plot’, ‘pushtete’, ‘pyetje’, and ‘vendin’ have appeared 119 time and have a frequency of 0.020% each. The words ‘anëtar’, ‘dëshiron’, ‘megjithatë’, and ‘unik’ have appeared 112 times and have a frequency of 0.018% each. The words ‘anëtarëve’, ‘bashkëkohore’, ‘larg’, ‘ndërmarrjes’, ‘Ne’, ‘shpejt’, ‘sistemeve’ have appeared 109 times out of 631,008 words in total from all sources and have a frequency of 0.017%.

The words ‘dhënave’, ‘duhur’, ‘fëmijë’, ‘planin’, ‘popullit’, ‘qytetit’ and ‘vendimtar’ have appeared 97 times and have a frequency of 0.015% each. The words like ‘bësh’, ‘dikush’, ‘filluar’, ‘Gjithavehtu’, ‘grup’, ‘numri’, ‘qysh’, and ‘yt’ have appeared 87 times in all sources and have a frequency of 0.014 % each. The words ‘angleze’, ‘Botërore’, ‘dallimi’, ‘Drejtën’, ‘dukuri’, ‘fuqinë’, ‘grupet’, ‘grupeve’, ‘krijohet’, ‘shoqërore’, ‘sisteme’, ‘sy’, and ‘zgjedhjeve’ have appeared in all sources 74 times and have a frequency of 0.012% each. The words such as ‘anëtarët’, ‘groupe’, ‘Jo’, ‘kolizionin’, ‘kombëtar’, ‘miratuar’, ‘moderne’, ‘nivelin’, ‘qetë’, ‘shpeshherë’, and ‘vendosën’ have appeared 69 times and have a frequency of 0.011% each.

Words like ‘dobët’, ‘dukshëm’, ‘familjen’, ‘format’, ‘fushë’, ‘gjetur’, ‘kafshët’, ‘karakteri’, ‘ligji’, ‘marrëveshje’, ‘ndikimin’, ‘personave’, ‘poezisë’, ‘pushtetin’, ‘qytetarët’, ‘rregullat’, ‘shpirti’, and ‘teje’ have appeared in all sources 60 times and have a frequency of 0.010% each. The words that have appeared 49 times are the words ‘anëtarësim’, ‘barabartë’, ‘dispozitave’, ‘dorën’, and ‘vuri’ which have a frequency of 0.01% each. Some of the words that have appeared 40 times are the words ‘aktivitetet’, ‘Amentit’, ‘aplikuar’, ‘butë’, ‘çmimin’, and ‘zgjedhje’ have a frequency of 0.008% out of 631,008 total counts from all sources.

Some of the words that are appeared 32 times in all sources are the words administrative’ ‘ardhshme’, ‘asocimit’ and ‘zbatojë’ have a frequency 0.005% each. The words such as ‘Andaj’, ‘arsyeja’, ‘ulëta’, ‘ushqimeve’, ‘ushqimit’, ‘vajtën’, ‘vëlla’, ‘vjeçare’, ‘vjershat’, ‘votuesit’, ‘zezë’, and ‘zgjerimin’ have appeared in all sources 26 times and have a frequency of 0.004% each. All of words that have appeared 20 times in all sources are; ‘administratës’, ‘Akti’, ‘Amerikës’, ‘arsimtare’, ‘arsimtarët’, ‘atdheut’, ‘baj’, ‘ballkanike’, ‘bashkëpunimin’, ‘besojnë’, ‘ciklit’, and ‘zgjedhja’ and ‘Zi’ have a frequency of 0.003% each.

The words that have appeared in all sources 17 times are; ‘aeroplan’, ‘afërsisht ‘afrohet’, ‘zhvilluara’, and ‘zmeraldi’ have a frequency 0.003% each. The words that have appeared 12 times are; absolutisht', 'Adrianën', 'Afrikës', 'afrohej', 'afruar', 'aksionarët', 'ambicie', 'analizës', 'Zëri', 'zgjidh', 'zvogëlohet', 'zyra', and 'zyrtar' have a frequency of 0.002% each. Words that have appeared 4 time in all sources are 'abstrakte', 'absurde', 'acaruar', 'adresat', 'afarizmit', 'afatshkurta', 'Kuvendin', 'laburiste', 'lagjet', 'lajmëtarët', 'lakminë', 'Lakorja', 'largësi', 'largëta', 'largimit', 'Largohu', 'largue', and 'larmishme' have a frequency of 0.001% each.

As illustrated in Figure 15, there are 631K words that appear in the figure.

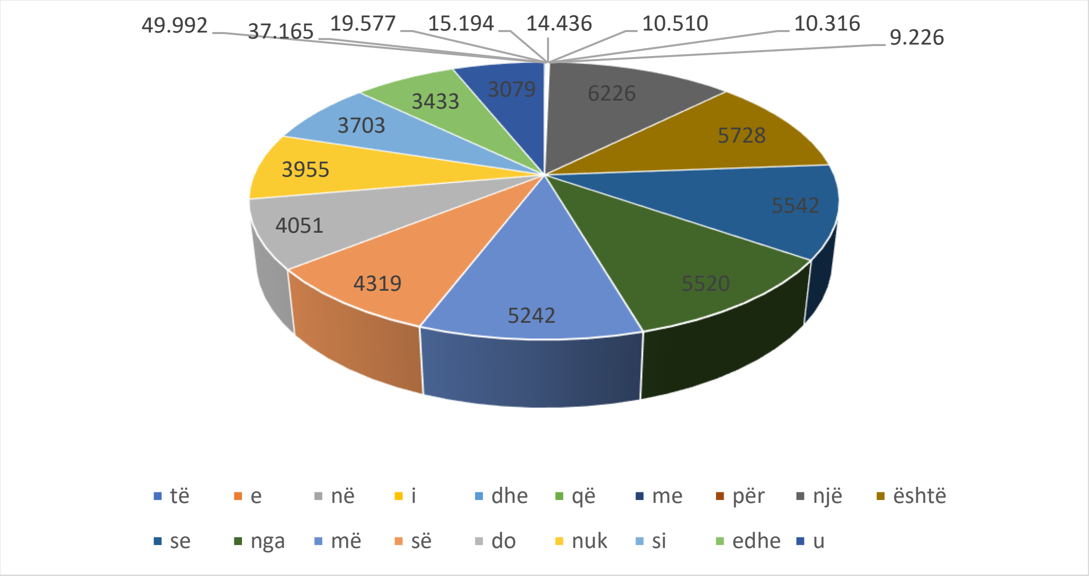
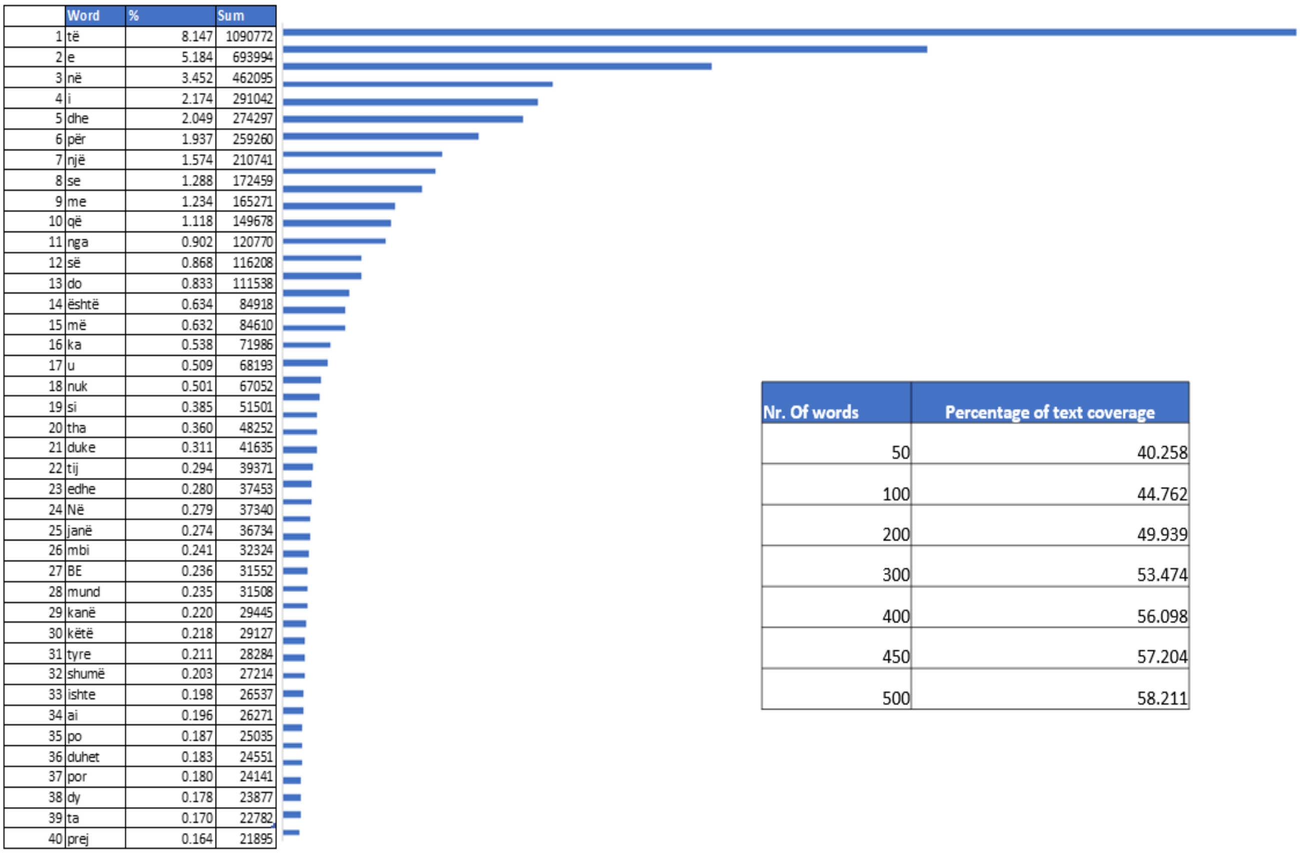


Figure 15. The word appearances in All Sources

The percentage and total have been determined by doing calculations using the terms shown in figure 16 below. For instance, the coverage percentage of 50 words in the Albanian language is 40.25 percent, the coverage percentage of 100 words is 44.76 percent, the coverage percentage of 200 words is 49.93 percent, the coverage percentage of 300 words is 53.47 percent, the coverage percentage of 400 words is 56 percent, the coverage percentage of 450 words is 57.20 percent, and the coverage percentage of 500 words is 58.211 percent.



	Word	%	Sum
1	të	8.147	1090772
2	e	5.184	693994
3	në	3.452	462095
4	i	2.174	291042
5	dhe	2.049	274297
6	për	1.937	259260
7	një	1.574	210741
8	se	1.288	172459
9	me	1.234	165271
10	që	1.118	149678
11	nga	0.902	120770
12	së	0.868	116208
13	do	0.833	111538
14	është	0.634	84918
15	më	0.632	84610
16	ka	0.538	71986
17	u	0.509	68193
18	nuk	0.501	67052
19	si	0.385	51501
20	tha	0.360	48252
21	duke	0.311	41635
22	tij	0.294	39371
23	edhe	0.280	37453
24	Në	0.279	37340
25	janë	0.274	36734
26	mbi	0.241	32324
27	BE	0.236	31552
28	mund	0.235	31508
29	kanë	0.220	29445
30	këtë	0.218	29127
31	tyre	0.211	28284
32	shumë	0.203	27214
33	ishte	0.198	26537
34	ai	0.196	26271
35	po	0.187	25035
36	duhet	0.183	24551
37	por	0.180	24141
38	dy	0.178	23877
39	ta	0.170	22782
40	prej	0.164	21895

Nr. Of words	Percentage of text coverage
50	40.258
100	44.762
200	49.939
300	53.474
400	56.098
450	57.204
500	58.211

Figure 16. Word Frequency

During the parsing of the text, we have noticed words that use special characters such as the hyphenated words, presented below is Table 11. The problem with these words is that they can be treated in three ways:

Example: ‘juridiko-civile’:

- juridiko, civile
- juridiko- ,civile
- juridiko-civile

The third version that will be adopted within the context of the thesis are calculated according to this. The hyphenated word that has a relatively higher frequency is the word ‘juridiko-civile’ which appears 113 out of a total of 631.008 tokens, with a frequency of 0.0179%.

The word ‘juridiko-private’ has appeared 38 times in all sources and has a frequency of 0.006%, the word ‘juridiko-civil’ has appeared 24 times and has a frequency of 0.0038%, the word ‘holandezo-flamane’ has appeared in all sources 20 times and has a frequency 0.0031%, the word ‘evro-atlantike’ has appeared 11 times and has a frequency 0.0017%, the word ‘pluralo-partiake’ has appeared 10 times and has a frequency of 0.0015%.

The word ‘anglo-amerikane’ has appeared 7 times and has a frequency of 0.0011%, the word ‘ekonomiko-shoqërore’ has appeared 6 times and has a frequency of 0.0009%, the word ‘shoqëror-politik’ has appeared in all sources 4 times and has a frequency of 0.0006%, the words that are appeared 3 times are; ‘ushtarako-politike’, ‘pluralo-politike’, ‘Maqedono-Kosovare’, and ‘kulturo-artistike’ and have a frequency of 0.0004%.

The words ‘çeko-sllovak’, ‘formalo-juridikisht’, ‘socio-ekonomike’, ‘edukativo-arsimore’, and ‘aritmetiko-logjike’ have appeared in all sources twice and have a frequency of 0.0003%. The hyphenated words with the lowest frequency are the words ‘latino-krishtere’, ‘partizano-çetnikët’, ‘liberalo-demokratike’, ‘filozofiko-politike’, ‘francezo-gjermane’, ‘evro-skeptikët’, ‘fantastiko-shkencore’, that appears once out of 631.008 with a frequency of 0.0001%.

Table 12. Frequency of hyphenated words from the sources out of 631k

Word	No. of Appearances	Frequency %
juridiko-civile	113	0.0179
juridiko-private	38	0.006
juridiko-civil	24	0.0038
holandezo-flamane	20	0.0031
evro-atlantike	11	0.0017
pluralo-partiake	10	0.0015
anglo-amerikane	7	0.0011
ekonomiko-shoqërore	6	0.0009
shoqëror-politik	4	0.0006

ushtarako-politike	3	0.0004
pluralo-politike	3	0.0004
Maqedono-Kosovare	3	0.0004
kulturo-artistike	3	0.0004
çeko-sllovak	2	0.0003
formalo-juridikisht	2	0.0003
socio-ekonomike	2	0.0003
edukativo-arsimore	2	0.0003
aritmetiko-logjike	2	0.0003
latino-krishtere	1	0.0001
partizano-çetnikët	1	0.0001
liberalo-demokratike	1	0.0001
filozofiko-politike	1	0.0001
francezo-gjermane	1	0.0001
evro-skeptikët	1	0.0001
fantastiko-shkencore	1	0.0001

Below you can find the same words that appear in different sources, as shown in Table 13, through which it can be observed that the word 'juridiko-civile' has a higher frequency in Source 2 (0.239%) than in Source 1 (0.021%). The word 'holandezo-flamane' has a higher frequency in Source 1 (0.064%) than in Source 2 (0.011%). The lowest frequency has is the hyphened word 'socio-ekonomike' – which in Source 5 has a frequency of 0.006% whereas in Source 4 it has a frequency of 0.0018%.

Table 13. Comparison of frequency for the same words appearing in different sources

Sources	Total words	Word	Appearance	Frequency %
Source 1	23342	juridiko-civile	5	0.021%
		holandezo-flamane	15	0.0006
Source 2	45240	juridiko-civile	108	0.239%
		holandezo-flamane	5	0.239%
Source 3	92471	juridiko-civile	1	0.011%
Source 4	55865	socio-ekonomike	1	0.0018%
Source 5	159375	socio-ekonomike	1	0.0018%

Furthermore, the words in the text that appears with ‘(apostrophe) such as ‘t’i’, ‘ç’rast’ etc. are handled as two tokens, Table 27 represented all the words divided with ‘ (apostrophe). The most apparent is t,’ with 772 and the lowest is ‘Dhiat’, ‘dit’, and ‘‘vdekura” appeared 1 time and have a frequency of 0.003%.

Table 14. Words with apostrophe

Words	No. Appearances	Frequency %
t’	772	0.214%
s’	343	0.095%

ç’	87	0.024%
m’	29	0.008%
S’	27	0.007%
Ç’	21	0.006%
d’	12	0.003%
n’	10	0.003%
T’	6	0.002%
N’	4	0.001%
D’	3	0.001%
gjith’	2	0.001%
l’	2	0.001%
Lal’	2	0.001%
Rek’	2	0.001%
shtetasit’	2	0.001%

Our results begin with a simple calculation of frequency, after which all sources' frequencies are calculated, and the total frequency is compared with it. Additionally, after reviewing all the results, we have taken the words from the different sources, and the total, which is the expected frequency, in order to calculate the average difference. For implementation in Python, text files in pdf format have been converted into .txt files. This is shown in table 15, which is called the matrix schema.

The frequency is calculated as follows:

d1=Total (Expected Frequency) - Source1

Average Difference is calculated according to the formula

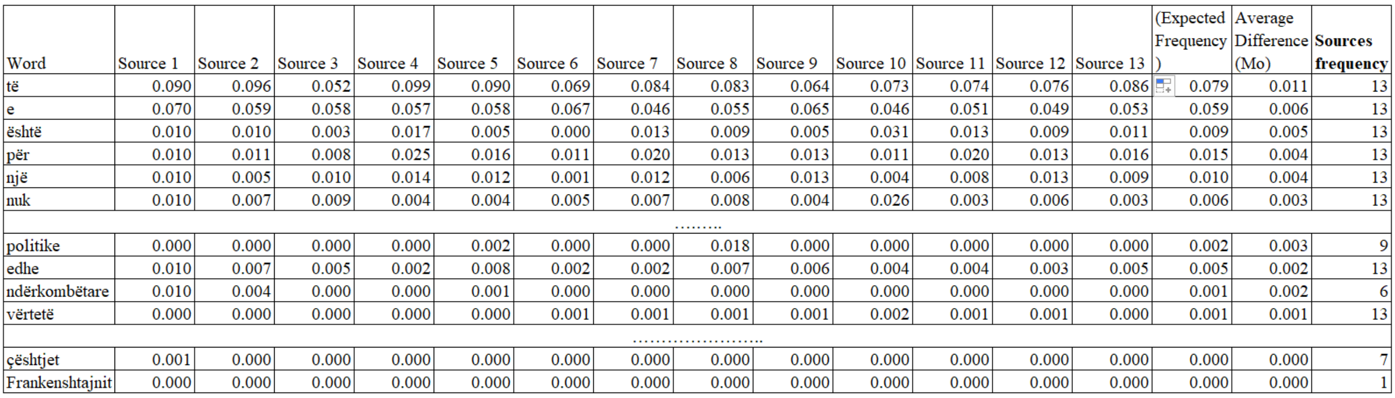
$$\triangle M=((M-M1)+(M-M2)+..(M-Mn))/n$$

Table 15. Matrix schema

Word	Source 1	Source 2	Source 3	Source 4	Source 5	…..	Source 13	Total (Expected Frequency)	Average Difference $\triangle M$
	d1	d2	d3	d4	d5	…..	d13		

After applying this schema in our corpus, we got the results that are represented in table 15; the words 'të' has the largest average difference with 0.011%, the word 'e' has an average difference with 0.006%, the words like 'dhe', 'është' and 'në' have an average difference is 0.005%; the words 'për', 'me' and 'një' have the average difference 0.004%; the words that haven average difference with 0.003 are the words 'nuk', 'politike', 'që', 'se', 'më' etc. With a value of 0.002 of average differences is the words 'ka', 'ndërkombëtare', etc. A lot of words that have an average difference of about 0.001 are 'vërtetë' etc.

Table 16. The average frequency for all sources

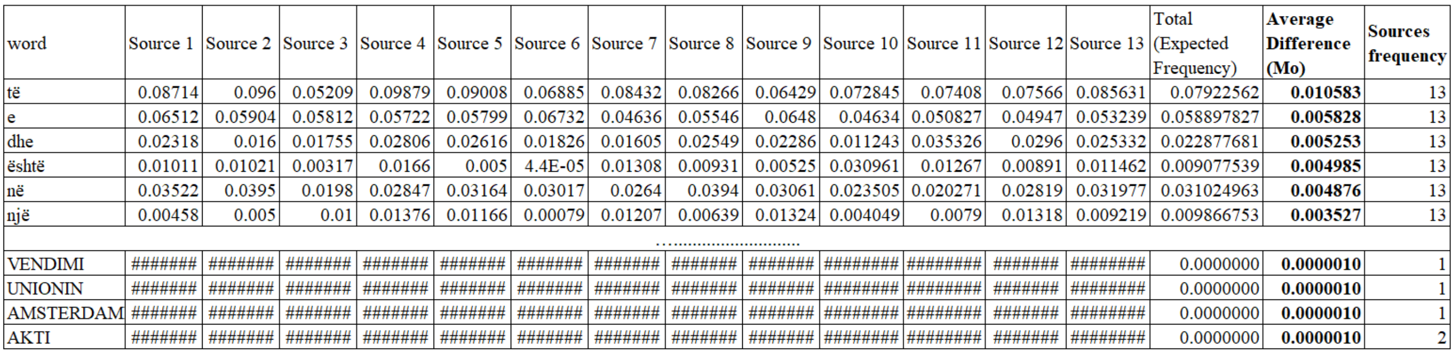


Word	Source 1	Source 2	Source 3	Source 4	Source 5	Source 6	Source 7	Source 8	Source 9	Source 10	Source 11	Source 12	Source 13	(Expected Frequency)	Average Difference (Mo)	**Sources frequency**
të	0.090	0.096	0.052	0.099	0.090	0.069	0.084	0.083	0.064	0.073	0.074	0.076	0.086	0.079	0.011	13
e	0.070	0.059	0.058	0.057	0.058	0.067	0.046	0.055	0.065	0.046	0.051	0.049	0.053	0.059	0.006	13
është	0.010	0.010	0.003	0.017	0.005	0.000	0.013	0.009	0.005	0.031	0.013	0.009	0.011	0.009	0.005	13
për	0.010	0.011	0.008	0.025	0.016	0.011	0.020	0.013	0.013	0.011	0.020	0.013	0.016	0.015	0.004	13
një	0.010	0.005	0.010	0.014	0.012	0.001	0.012	0.006	0.013	0.004	0.008	0.013	0.009	0.010	0.004	13
nuk	0.010	0.007	0.009	0.004	0.004	0.005	0.007	0.008	0.004	0.026	0.003	0.006	0.003	0.006	0.003	13
								………								
politike	0.000	0.000	0.000	0.000	0.002	0.000	0.000	0.018	0.000	0.000	0.000	0.000	0.000	0.002	0.003	9
edhe	0.010	0.007	0.005	0.002	0.008	0.002	0.002	0.007	0.006	0.004	0.004	0.003	0.005	0.005	0.002	13
ndërkombëtare	0.010	0.004	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.002	6
vërtetë	0.000	0.000	0.000	0.000	0.000	0.001	0.001	0.001	0.001	0.002	0.001	0.001	0.000	0.001	0.001	13
								………………….								
çështjet	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	7
Frankenshtajnit	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1

This version takes the numbers of words from different sources with three digits, showing that starting with 1303 words rows becomes 0.000 over 49515words rows in total, which concludes that frequency and average difference should not be 0. Because it was limited to 1303 words rows for this reason ongoing, we take the numbers without digit restrictions from all the sources.

Furthermore, the results in table 16 below, shows that higher average differences have the words 'të' with 0.10583%; the word 'e' with 0.005828%; the word 'dhe' has a 0.005253 average difference; word 'është' has a 0.004985 average difference; with average difference results of 0.004876 is the word 'në'; word 'një' has the average difference 0.003527 and so one, till at the row 49534 of the word above 49585 words in total from all the sources, starting from the word 'VENDIMI' has the total expected frequency 0.0000000000 and the average difference of 0.000001, the rest of the words such as 'UNIONIN', 'AMSTERDAMIT', 'AKTI' etc. have the same average difference.

Table 17. Average Difference for 13 sources



word	Source 1	Source 2	Source 3	Source 4	Source 5	Source 6	Source 7	Source 8	Source 9	Source 10	Source 11	Source 12	Source 13	Total (Expected Frequency)	**Average Difference (Mo)**	**Sources frequency**
të	0.08714	0.096	0.05209	0.09879	0.09008	0.06885	0.08432	0.08266	0.06429	0.072845	0.07408	0.07566	0.085631	0.07922562	**0.010583**	13
e	0.06512	0.05904	0.05812	0.05722	0.05799	0.06732	0.04636	0.05546	0.0648	0.04634	0.050827	0.04947	0.053239	0.058897827	**0.005828**	13
dhe	0.02318	0.016	0.01755	0.02806	0.02616	0.01826	0.01605	0.02549	0.02286	0.011243	0.035326	0.0296	0.025332	0.022877681	**0.005253**	13
është	0.01011	0.01021	0.00317	0.0166	0.005	4.4E-05	0.01308	0.00931	0.00525	0.030961	0.01267	0.00891	0.011462	0.009077539	**0.004985**	13
në	0.03522	0.0395	0.0198	0.02847	0.03164	0.03017	0.0264	0.0394	0.03061	0.023505	0.020271	0.02819	0.031977	0.031024963	**0.004876**	13
një	0.00458	0.005	0.01	0.01376	0.01166	0.00079	0.01207	0.00639	0.01324	0.004049	0.0079	0.01318	0.009219	0.009866753	**0.003527**	13
								……………………….								
VENDIMI	#######	#######	#######	#######	#######	#######	#######	#######	#######	########	########	#######	########	0.0000000	**0.0000010**	1
UNIONIN	#######	#######	#######	#######	#######	#######	#######	#######	#######	########	########	#######	########	0.0000000	**0.0000010**	1
AMSTERDAM	#######	#######	#######	#######	#######	#######	#######	#######	#######	########	########	#######	########	0.0000000	**0.0000010**	1
AKTI	#######	#######	#######	#######	#######	#######	#######	#######	#######	########	########	#######	########	0.0000000	**0.0000010**	2

In this we can conclude that the lower the expected frequency is of the word the larger the average e difference will be, this way will help us to build the Albanian dictionary. Furthermore, is considered the calculation of sources frequency for each word appearance. This will provide more information about the rare words which have a low frequency but appear in all sources.

The text tokenization has generated 250,152 words – for which the individual frequency
analysis and difference average (△M) frequency were calculated, which deduced four different word categorizations: (1) candidate words for the dictionary, (2) extreme cases, (3) nonextreme cases, and (4) rare words shown in table 18 below.

Table 18. Four different word categorizations

NO	Case	Value	Comment
1	$\frac{\boldsymbol{M}}{\Delta\boldsymbol{M}}$	$\boldsymbol{M}$ ↑	Word candidate for dictionary
		$\Delta\boldsymbol{M}$ ↓	
2	$\frac{\boldsymbol{M}}{\Delta\boldsymbol{M}}$	$\boldsymbol{M}$ ↑	Extreme cases, should be discussed, those words maybe should be not in vocabulary
		$\Delta\boldsymbol{M}$ ↑	
3	$\frac{\boldsymbol{M}}{\Delta\boldsymbol{M}}$	$\boldsymbol{M}$ ↓	Non-extreme cases but should be discussed, those words should be on the waiting list
		$\Delta\boldsymbol{M}$ ↑	
4	$\frac{\boldsymbol{M}}{\Delta\boldsymbol{M}}$	$\boldsymbol{M}$ ↓	Rare words
		$\Delta\boldsymbol{M}$ ↓	

1. Case: $\frac{\boldsymbol{M}}{\Delta\boldsymbol{M}}$

In the first case; if M is a large value and ΔM is a small value, we can conclude that the frequency of use of these words is very large, but also the difference is small use in different sources, which the distribution of the word is almost the same in all sources, this means that these words are correct and should be treated as words that are candidates to be in the vocabulary. So, they have acceptable distributions in all resources.

2. Case: $\frac{\boldsymbol{M}}{\Delta\boldsymbol{M}}$

In the second case, if both M and M are large values, we can draw the conclusion that not only is the frequency of use of these words very high, but also the difference between them is high.

This indicates that these words may be incorrect versions of words that are used frequently in one or more sources, but which have been used incorrectly. Therefore, this is an extreme example that has to be examined, and it's possible that we don't use these particular terms very often.

3. Case: $\frac{\boldsymbol{M}}{\Delta\boldsymbol{M}}$

In the third case; if M and ΔM are both small values, we can conclude that the frequency of use of these words is small, but the difference is also large. So, these words are used once or a few times in some sources, which may be a more specific word or part of a particular field. They may not be wrong, but they may be treated as words on the waiting list for verification when the vocabulary is enriched to see what will occur.

4. Case: $\frac{\boldsymbol{M}}{\Delta\boldsymbol{M}}$

In the fourth case; if M and ΔM are both small values, we can conclude that the frequency of use of these words is small, but also the difference is small of use in different sources, so these words can be considered as rare words. Furthermore, some existing algorithm that contributes to the processing of rare words should be implemented.

As calculated during tokenization, the corpus used generates 49,514 unique words – which represent the base for further experimentation. Noting that word frequency is relative to each individual source when calculating the average frequency, this paper has taken into account the concept of the regression toward [29] the mean as well as the law of large numbers; [30] noting that as additional sources would be added to the corpus, the more the frequency would be rounded up to the expected average.

To this end, the generated unique words through tokenization have been (i) calculated individually in relation to their absolute frequency in each source and then (ii) found the average frequency from the combined absolute frequency from all sources. Below we have the formula (1) used:

$$\mathrm{M_s}=\frac{\mathrm{Ws}}{\mathrm{Wst}}$$

and consequently:

$$\mathrm{M_T}=\sum_{Ms}^{s=13} st$$

or summarily:

$$\mathrm{MT}\frac{\sum s=13\frac{Ws}{Wst}}{St}$$

(1)

Where:

- Ms denotes an individual word’s absolute frequency per source
- MT denotes total expected frequency
- Ws denominates individual words within each source.
- Wst denominates the total number of words within each source
- St denominates the total amount of sources

Formula (1) has been used for all the unique words generated from the experiment, to find the total expected frequency for all the words. This is denoted by ‘M’.

The next step in the experiment was to calculate the average difference of expected frequency so that we can rank-order all words in terms of their occurrence across multiple sources. This is done via the following formula (2):

$$\triangle MT=((MT-MS1)+(MT-MS2)+(MT-MS3)\ldots(MT-MSn))/St \tag{2}$$

Where:

- △MT denotes the Average Difference of Expected Frequency
- MT denotes total expected frequency
- Ms denotes individual word's absolute frequency per source

Having calculated MT and △MT, the corresponding correlation is used to sort the words in categories:

$$Ma=\frac{\triangle MT}{Wn}$$

$$Mw=\frac{MT-\triangle MT}{Ma} \tag{3}$$

Where:

- △MT denotes the Average Difference of Expected Frequency
- MT denotes total expected frequency
- Ms denotes individual word's absolute frequency per source
- Ma denotes the overall Average Difference of Expected Frequency
- Mw denotes the deviation from the average of the Expected Frequency

There are three categories that are observed because of these experimentations:

(1) definite candidate words for the dictionary,

(2) potential candidate words for the dictionary,

(3) rare words.

This categorization will be used to decide which words will be included in the dictionary, and which words will be omitted and discarded. Each category corresponds to a specific correlation between MT and △MT – as described in the following cases.

Case 1: definite candidate words for the dictionary

In the first case, when sorting for the variables from the Mw, the group of words that result in an index larger than 1 is selected as words that are strong contenders to be added to the dictionary. The experimentation yielded an extremely positive result of 55.49, represented by word 'të' – which is the most used word in the text sources used in this paper.

Because of the linguistic properties of languages, in this case, the Albanian language, this case will typically include prepositions, conjunctions, adverbs that are used more frequently throughout a simple sentence composition. Examples of such words are presented in Table 19. Approximately 48% from all sources.

Table 19. Words from the first case

WORDS	**M_W [>1.0]**
të	55.4942
gjitha	9.0986
këtë	8.7162
……..	
disa	5.6283
tjetër	3.9182

janë	3.3404
……..	
vend	2.1712
brenda	1.9539
veçantë	1.7344
……..	
kështu	1.5950
fundit	1.5563
vërtetë	1.4946

Case 2: potential candidate words for the dictionary

In the second case, when sorting for the variables from the Mw, the group of words that result in an index larger than 0.1 but lower than 1 are considered as potential candidate words to be included in the dictionary. These words are typically nouns, verbs, and adjectives, that in relative terms are less often used as opposed to the grammatical categories that are included in Table 20. Approximately 37% from all sources.

Table 20. Words from the second case

WORDS	**Mw [0.1-1]**
lartë	1.2973
qenë	1.1180
pastaj	1.0625
……..	

punën	0.8838
jetën	0.8385
kombëtare	0.7443
……..	
shtëpi	0.6945
veçanta	0.6564
unike	0.6152
……..	
qytetarëve	0.3762
Evropën	0.2940
zgjedhjet	0.1899

Case 3: rare words

In the third case, when sorting for the variables from the Mw, the group of words that result in an index smaller than 0.1 are considered rare words, and as such, they are not included in the dictionary. These words need further attention by linguists; or their index value may increase because of corpus expansion – as new sources are added, the word could have more incidences of occurrence. For this case, the experimentation yielded a marginally positive result of 0.0993, represented by word 'progress'. This word is typical for Case 3 because it is not an official word, but an unofficial adaptation from other languages (progress (Eng.: progress) – in Albanian: përparim). Table 18 presents more examples of such words. Approximately 15% from all sources.

Table 21. Words from the third case

WORDS	**M_W [<0.1]**
progres	0.0993
vazhdueshëm	0.0981

kompjuterëve	0.0893
……..	
edukimi	0.0865
inteligjencave	0.0851
Shpirtërore	0.0743
……..	
zotësitë	0.0621
dyanshëm	0.0544
ushtrimeve	0.0427
……..	
Kuptosh	0.0227
balansuara	0.0167
supozimet	0.0187

Furthermore, some existing algorithm that contributes to the processing of rare words should be implemented. In the case of rare words, we suggest providing the list of such words to linguists so that they can determine which of these words are foreign and which are authentic Albanian language words. All three categories are represented in Table 19 based on 13 sources.

Table 22. The frequency of all sources

Case	***13 Sources***
(1)	48%
(2)	37%
(3)	15%

Nevertheless, from 631.008 words, through tokenization only 49,514 words were selected that are correct, whereas 581,494 were words that have other characters or are misspelled. The linguists, in this case, would examine those words and decide which will be added to the Albanian language dictionary.

4.1.1 Pearson Correlation testing

In addition, in order to adequately reflect all of those findings, we have made use of the formula for the Pearson Correlation Coefficient for 250,152 words, which demonstrates that rxy =0.87 is significant.

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

x=M

y=$\triangle M$

rxy= 0.87

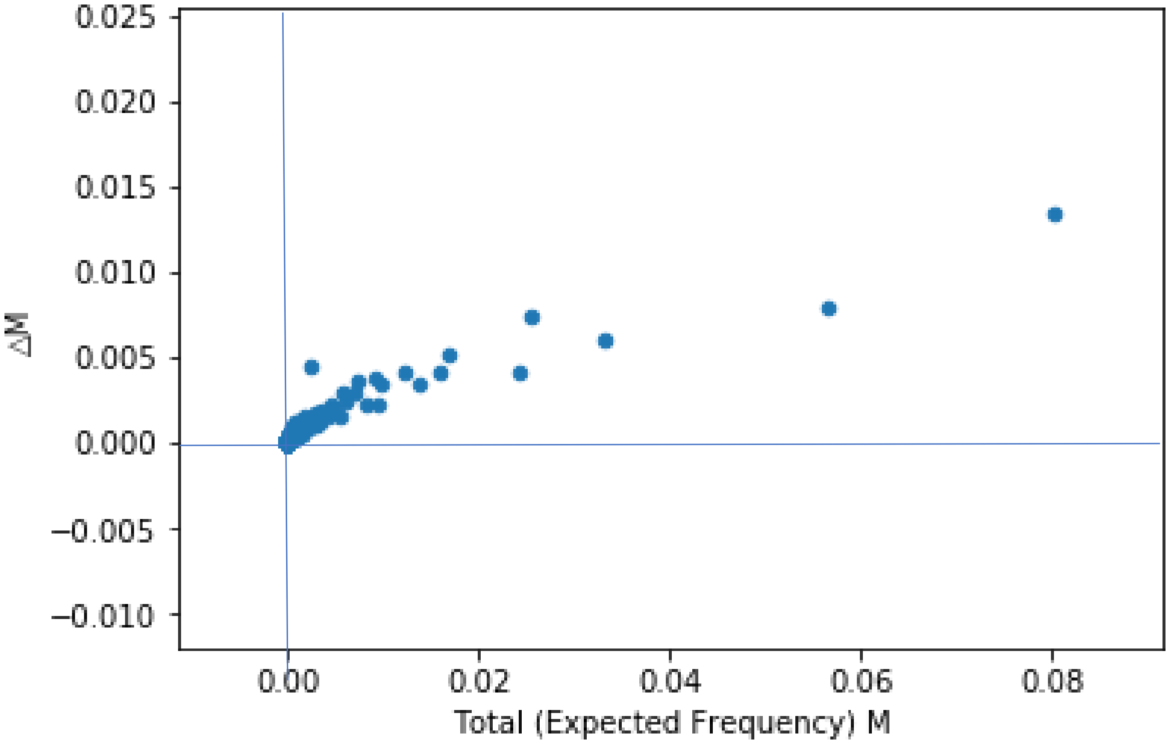


Figure 17. Pearson Correlation ation Coefficient for all sources

Figure 17 represents 3000 words and has a significance of rxy =0.928

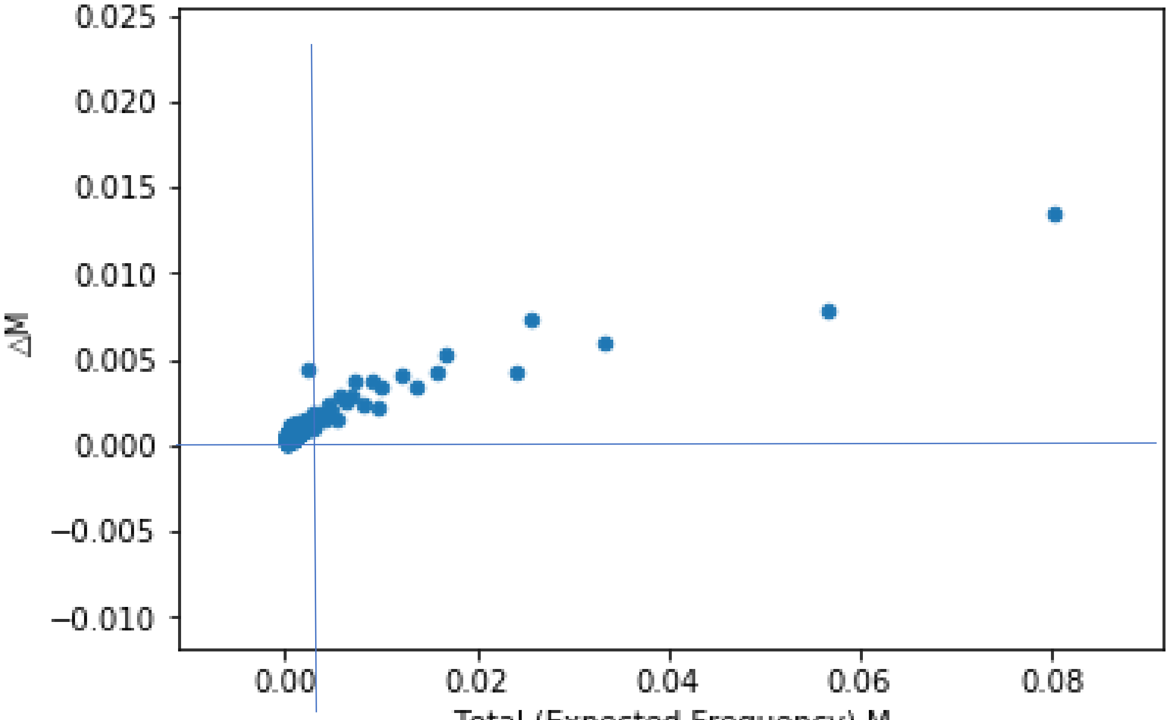


Figure 18. Pearson Correlation Coefficient for 3000 words

As can be seen in Figure 18, 300 words were randomly selected; significance = 0.210

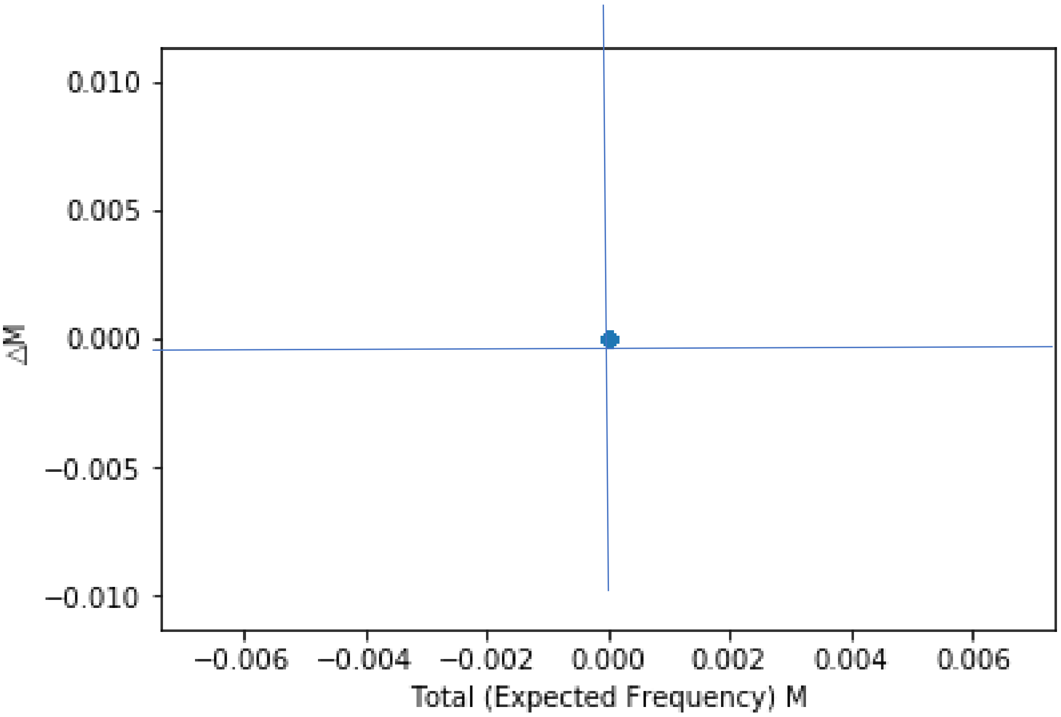


Figure 19. Pearson Corrolation Coefficient of 300 words

The corpus has 250,152 words; figure 19 represents the correlation of 300 last words from the dictionary which has the expected frequency 0.00000001. The significance is rxy= 0

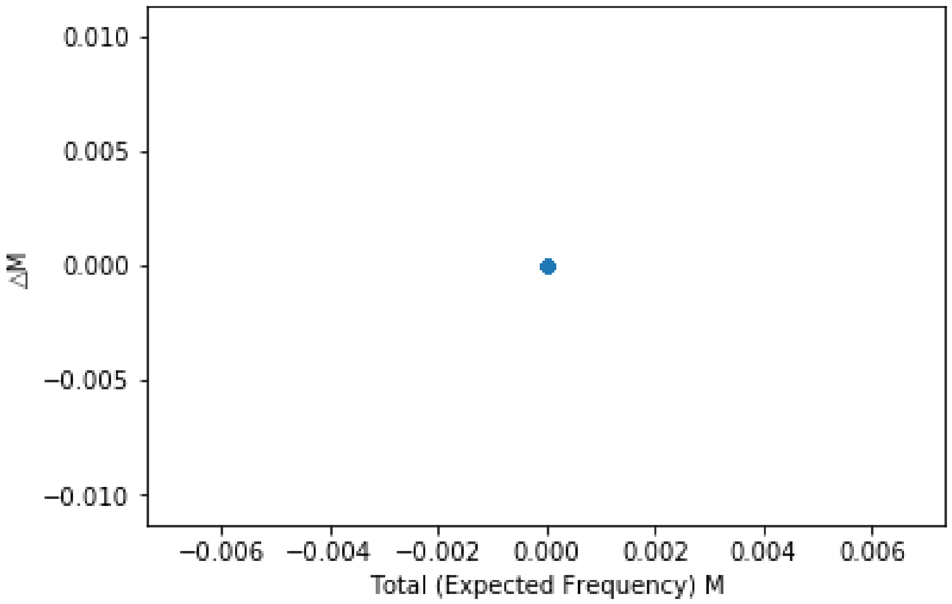


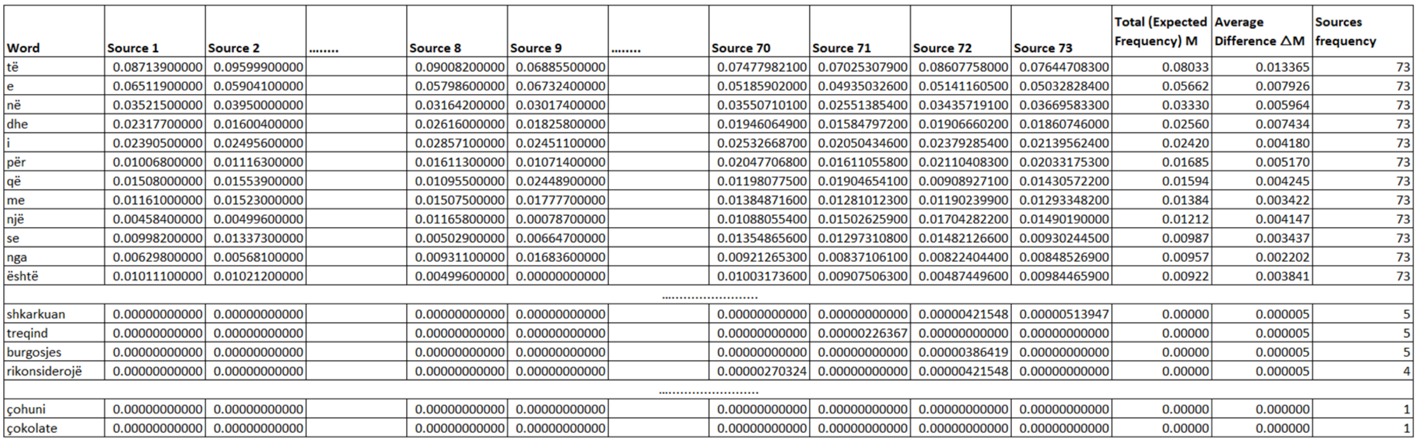
Figure 20. Pearson Correlation Coefficient the last 300 words from dictionary

While we tested as a version, no progress was made in our topic because the Pearson Correlation Coefficient formula for regression uses two different variables, whereas in this case there is one independent variable. Therefore, the results are not based on data.

4.1.2 Adding new sources

Afterwards, 60 additional sources were added to the data source with a total capacity of 77MB, amounting to a total of 631,008 words. So, in total so far, the experiment included 73 sources and 250,152 words in total. Table 23 presents a selection of terms from several sources, together with their anticipated frequency and average differences. The word 't' has the largest predicted frequency and average differences, followed by the word’s 'n', 'q', 'e', 'dhe', and so on. In addition, for each word that appeared in the studies, the frequency of the source was determined. This will provide you more details on the terms that are uncommon yet present in all sources.

Table 23. Average Difference for all sources



Word	Source 1	Source 2	……..	Source 8	Source 9	……..	Source 70	Source 71	Source 72	Source 73	Total (Expected Frequency) M	Average Difference ΔM	Sources frequency
të	0.08713900000	0.09599900000		0.09008200000	0.06885500000		0.07477982100	0.07025307900	0.08607758000	0.07644708300	0.08033	0.013365	73
e	0.06511900000	0.05904100000		0.05798600000	0.06732400000		0.05185902000	0.04935032600	0.05141160500	0.05032828400	0.05662	0.007926	73
në	0.03521500000	0.03950000000		0.03164200000	0.03017400000		0.03550710100	0.02551385400	0.03435719100	0.03669583300	0.03330	0.005964	73
dhe	0.02317700000	0.01600400000		0.02616000000	0.01825800000		0.01946064900	0.01584797200	0.01906660200	0.01860746000	0.02560	0.007434	73
i	0.02390500000	0.02495600000		0.02857100000	0.02451100000		0.02532668700	0.02050434600	0.02379285400	0.02139562400	0.02420	0.004180	73
për	0.01006800000	0.01116300000		0.01611300000	0.01071400000		0.02047706800	0.01611055800	0.02110408300	0.02033175300	0.01685	0.005170	73
që	0.01508000000	0.01553900000		0.01095500000	0.02448900000		0.01198077500	0.01904654100	0.00908927100	0.01430572200	0.01594	0.004245	73
me	0.01161000000	0.01523000000		0.01507500000	0.01777700000		0.01384871600	0.01281012300	0.01190239900	0.01293348200	0.01384	0.003422	73
një	0.00458400000	0.00499600000		0.01165800000	0.00078700000		0.01088055400	0.01502625900	0.01704282200	0.01490190000	0.01212	0.004147	73
se	0.00998200000	0.01337300000		0.00502900000	0.00664700000		0.01354865600	0.01297310800	0.01482126600	0.00930244500	0.00987	0.003437	73
nga	0.00629800000	0.00568100000		0.00931100000	0.01683600000		0.00921265300	0.00837106100	0.00822404400	0.00848526900	0.00957	0.002202	73
është	0.01011100000	0.01021200000		0.00499600000	0.00000000000		0.01003173600	0.00907506300	0.00487449600	0.00984465900	0.00922	0.003841	73
…………………….													
shkarkuan	0.00000000000	0.00000000000		0.00000000000	0.00000000000		0.00000000000	0.00000000000	0.00000421548	0.00000513947	0.00000	0.000005	5
treqind	0.00000000000	0.00000000000		0.00000000000	0.00000000000		0.00000000000	0.00000226367	0.00000000000	0.00000000000	0.00000	0.000005	5
burgosjes	0.00000000000	0.00000000000		0.00000000000	0.00000000000		0.00000000000	0.00000000000	0.00000386419	0.00000000000	0.00000	0.000005	5
rikonsiderojë	0.00000000000	0.00000000000		0.00000000000	0.00000000000		0.00000270324	0.00000000000	0.00000421548	0.00000000000	0.00000	0.000005	4
…………………….													
çohuni	0.00000000000	0.00000000000		0.00000000000	0.00000000000		0.00000000000	0.00000000000	0.00000000000	0.00000000000	0.00000	0.000000	1
çokolate	0.00000000000	0.00000000000		0.00000000000	0.00000000000		0.00000000000	0.00000000000	0.00000000000	0.00000000000	0.00000	0.000000	1

Increasing the number of sources to 73, we have conducted different experiments for all word categorizations.

Case 1: Definite candidate words for the dictionary

First, when sorting the variables from the MW, words that result in an index larger than 1 are considered strong candidates for inclusion in the dictionary. A total of approximately 63%.

Table 24. The first case's words

WORDS	M_W [>1.0]
të	14891.29
gjitha	186.86
janë	567.67
……..	
shumë	407.28
disa	161.89
tjetër	145.47

vend	58.50
brenda	16.68
veçantë	8.35
ndihma	3.98
gjenetikës	1.64
klimë	1.15

Case 2: Potential candidate words for the dictionary

Similarly, when sorting for the variables from the Mw, the group of words with an index greater than 0.1 but lower than 1 is considered a potential candidate for inclusion in the dictionary. The overall percentage is about 29% calculated from 73 sources.

Table 25. Examples of words from the second case

WORDS	MW [0.1-1]
sociale	1.00
lutjen	0.99
…..	
dijetarë	0.74
firme	0.54
shkarkohet	0.42
……..	
arriturave	0.27

falëm	0.18
diskutoje	0.12
……..	
verzion	0.11
shqiptaro	0.11
biologë	0.10

Case 3: rare words

In the third case, the group of words whose index is smaller than 0.1 when sorted for the variables from the MW is considered rare words, and as such, they are not included in the dictionary. Amount based on 73 sources: 8%.

Table 26. Examples of third case

WORDS	**M_W [<0.1]**
fëminore	0.099
multietnicitetit	0.086
sezonet	0.077
……..	
etnikësh	0.063
mjerimi	0.056
prushi	0.052
……..	
pararoje	0.049

titiani	0.043
esnafëve	0.039
……..	
përfaqësoja	0.027
tyrqet	0.015
investigohen	0.003

Based on a comparison of 13 sources and 73, Figure 27 shows the results.

Table 27. Comparison percentage from different sources

Case	***13 Sources***	***73 Sources***
(1)	48% (119.894)	63% (157.362)
(2)	37% (92.418)	29% (72.436)
(3)	15% (37.469)	8% (19.983)

Furthermore, some examples are provided for some categories. So far, the contribution to the dictionary is promising, as 48% (119.894) of the definite candidate words for the dictionary, 37% (92.418) of potential candidate words, and 15% (37.469) of rare words were identified in the research with 13 sources. In contrast, when we increased the number of courses to 73, the results were 63% (157.362) for the first category, 29% (72.436) for potential candidate words, and 8% (19.983) for rare words.

4.1.3 Vocabulary correction effects with words that appear only once in one source

The distribution of words that appear only once as they are spread across sources. Moreover, for our experiments, we have analyzed the words appearing only once in one source. In the third category, 20.001 words appear only once, and they are not in any of the other categories. There is 8% of the words appear only once in the third category, representing rare words.

Moreover, we examined words that appear three times and twice in different sources. Table 28 shows that in Case 2, the words appear approximately 3.638 times, while in Case 3, the words appear 15.128 times. The number of words that appear twice in Case 2 amounts to 117.371.

Table 28. Distribution of words appearing in sources

WORD APPEARANCE	CASE 1	CASE 2	CASE 3
3	0	3.638	15.128
2	0	0	117.371
1	0	0	20.001

Additionally, the results are represented by a chart to illustrate the distribution of words in Case (1) Definite candidate words for the dictionary, Case (2) Potential candidate words for the dictionary, and Case (3) rare words. The results are shown in Fig.21.

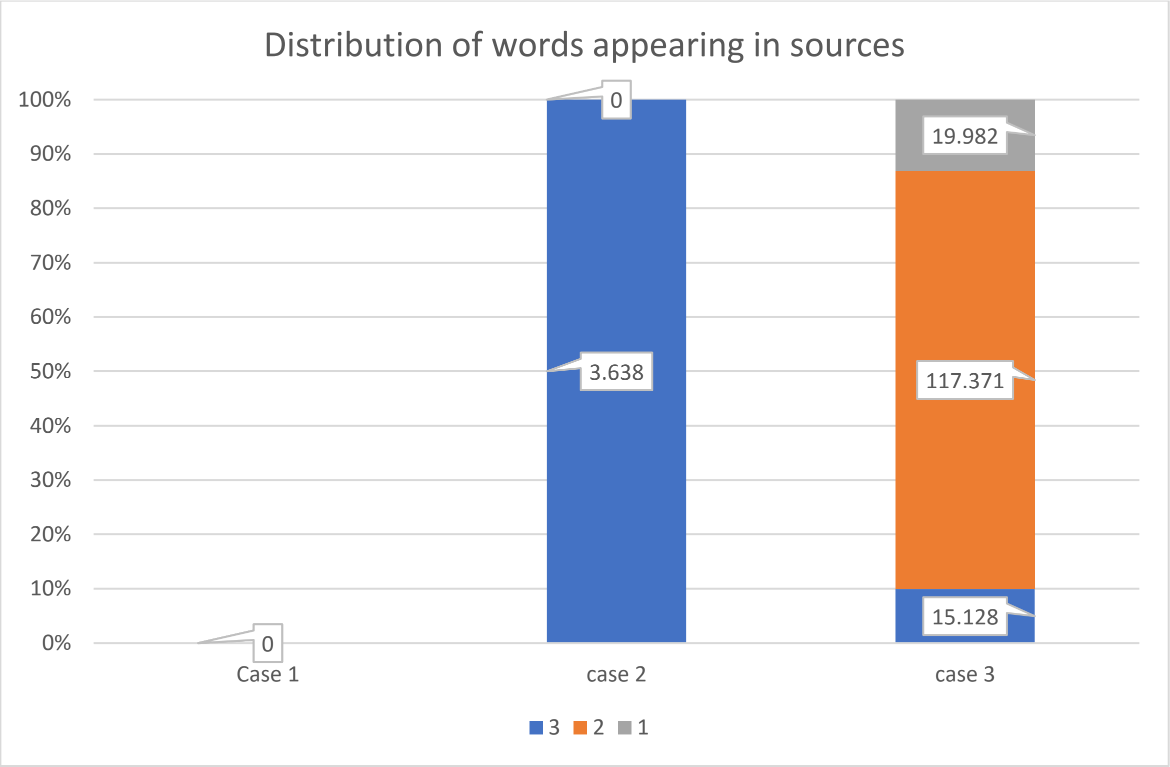


Figure 21. Distribution of words in different cases

Furthermore, from the chart, we can see that in case 1 which includes 20.001 words and has only appeared in one source, we have obtained a great result. On another hand, Case (3) includes 117.371 words that have appeared twisted in the sources, case (2) includes 3.638 words that have appeared

three times in the sources, and Case (3) includes 15.128 words and have appeared in the three sources.

4.1.4 Correction of words spelled in lowercase or uppercase

While building a dictionary, we considered also uppercase words, such as titles, names of places, names of institutions, etc. By using a Python script, we have converted all uppercase letters to lowercase in this part.

We have tokens in the corpus, for example, 'Shkupi', which is also spelled 'shkupi' because both words have the same meaning as listed in table 29.

Table 29. Converting all words to lowercase

WORD (CASE 3)	Corrected Words
PUNA	puna
France	france
VENDIN	vendin
……	
Pushimore	pushimore
Luftoi	luftoi
VETES	vetes
……	
INFORMATIKA	informatika
Shtegu	shtegu
Politikë	politikë

We have analyzed that a sentence, such as Tetova, begins with a capital letter after it appears at the beginning of the sentence, so to convert this word, we compared if it often appears in the lower case and then converted it. There have also been cases in which the word 'TETOVA' was only written in capital letters and if its frequency was high then it was not converted.

Based on the frequency of each word, the comparison formula was used to analyze the data. The process was repeated for all the categories. It is estimated that the number of words in the dictionary has decreased by 11% and the total is now 220,604 (89%)

Table 30. Results of categorizing words by converting characters

Case	*Word*	*Percentage %*
(1)	142.072	65%
(2)	64.151	29%
(3)	14.438	6%

In the category of rare words are also included words that are misspelled and words that are merged. Such as 'franqezogjermane', 'SocialDemokratike', 'VeriAtlantikë', 'RaportPrgoresi' etc. There are also misspelled words such as 'taj', 'tecce', 'remolo', 'renvoi' etc.

4.2 Reinforcement learning with manually annotated data

We have compiled this list, and throughout the procedure, we will be sure to include the terms that are on it. must conform to the manner in which these phrases are presented in the dictionary or to the manner in which they are appropriately identified in the dictionary. The fact that the dictionary contains many inaccurate entries, which we discovered, is, in our opinion, incorrect.

We have analysed the third category to determine which words have had their spellings rectified and which terms still need attention. The following are some instances that fall under the third category. It will be simpler in this manner to monitor whether or not the dictionary has been updated accurately.

Table 31. Verifying words manually

Correct Words	Misspelled words
Kosovë	rfederalistë

regjistova	zum
Njeriut	zyhdiut
ndryshme	zimesh
Evropiane	vjzave
rreziqet	vjeë
luftës	villacska
votave	verijub
Ushtrisë	urdhni
Sigurimit	unicefi
takimi	dnevik

Random selection was used to choose, from within each group, a total of 150 properly spelt words. There was not a single misspelling in any of the first category's fifty random words.

In the second scenario, there are just 12 words with an incorrect spelling, which accounts for around 24 percent, thus we selected those words to be rewritten with the proper spelling. In the third instance, 76 percent of the text, or 38 words, included misspelt terms, and the right words had to be picked by hand. Afterward, an automated method may be derived by utilising the ready list as a resource.

Table 32. List of vocabulary words

Words from 2nd category	Words from 2nd category	Words from 3rd category
lirisë	çertifikatës	fesat
martën	copëzimi	enegjinë
Zoti	diamantit	magjistraturën
……..		
Maqedonisë	interesojë	diskutushëm
zihesh	dhuratash	zërimet
marrë	kanadezi	dizinfektoi
……..		
Ministri	ftohte	zhvillohesh
politike	dendësoi	zjarrë
kolizionit	çështja	Ekuilibrojmë
……..		
vendit	festoni	emaila
Njeriut	apliko	fakturimi
juridike	zgjedhje	fëmijëve

4.3 Evaluation

In this chapter, we presented all of the experiments that we conducted and the empirical data that we obtained while working on the Albanian language dictionary. Unsupervised learning allowed us to construct this lexicon, which we can now use.

In Section 4.1, we presented all the empirical results for the text Natural Language Processing. The evaluated method was unsupervised machine learning. Initially, the results of the tokenization process for the text of the Albanian Language. In the following experiments, we examined the frequency and appearance of tokens. A variety of special characters experimented, including hyphenated and apostrophe words.

We had a categorization of words for the dictionary where Mw [>1.0] represents approximately 48% from all 13 sources, potential candidate words for the dictionary where Mw [0.1-1] have produced 37% from all sources, and rare words are represented by 15% from all sources where Mw [<0.1].

In Section 4.1.1, we conduct an analysis using the Pearson Correlation Coefficient to evaluate our data. In spite of testing as a version, we did not make any headway on our issue. This is due to the fact that our topic depends on two variables, while this example just depends on one variable. As a consequence, the findings are not supported by the data.

In Section 4.1.2, the results on the accuracy of 73 sources were presented. We saw three-word categorization in which Mw[>1.0] had the best results approximately 63% from all sources, following case 2 where Mw[0.1-1] has 29% overall and the rare words category amount based on 73 sources is 8%.

We focused on comparing word categorization across 13 and 73 sources, and so far, the results are promising.

Section 4.1.3 represents analyzing the word appearing only one in one source. The third category of words with 8% appears only once, approximately 20.001 words. Additionally, are presented the words appear approximately 3.638 times in Case 2, they appear 15.128 times in Case 3. In Case 2, 117.371 words appear twice.

The frequency of each term was reported in section 4.1.4, and the comparison formula was used in order to conduct an analysis of the data. This method was carried out with regard to every category. The number of terms in the dictionary has decreased by 11%, and there are now 220,604 of them (89 percent).

Section 4.2 represents reinforcement learning with manually annotated data. A total of 150 correctly spelled words were randomly chosen from each category. In the first case, all words are correct, in the second case approximately 24%, and in the third case, 76% of the words are misspelled.

5

Chapter

5. UNSUPERVISED POS TAGGING

Unsupervised POS tagging does not utilize labeled text or predetermined categories like regular (supervised) POS tagging. As a point-of-sale tagger, an application in and of itself, but also functions as a preprocessing stage for systems that will be built on top of it, the names because the number of syntax categories is often insufficient important.

Large data collection efforts in any domain which require extensive data annotation can be supported and time-consuming [38]. To allow a machine learning system to understand and learn from the information presented, and placed in such a configuration a manner that the computer can easily execute patterns and conclusions. Including all relevant information in a dataset is a common method for achieving this goal.

In the field of natural language, datasets are referred to as corpora, and an annotated corpus is a single collection of data that has been annotated with points that are mostly related to one another.

In light of this, the process of linguistic annotation could prove to be an essential component in the creation of intelligent human language systems. In a language with little available resources, such as Albanian, there is no publicly accessible morphologically annotated corpus and no tools for lemmatization, morphologically, or part-of-speech tagging.

A contribution that will be made to the Albanian language is the creation of a publically accessible corpus containing manually annotated part-of-speech tags, morphological characteristics, and lemmas. We have compiled the corpus of the Albanian language, which is comprised of 631.008 tokens taken from a variety of text sources.

In Albanian, the unsupervised POS-tagger is created from scratch. There is a large amount of unlabeled, tokenized data entering the system that is monolingual without POS information. After applying the Chinese Whispers algorithm to analyzing distributional similarity of a corpus, a subset of the 250,000 most frequent words in the database are clustered together in the range of a few hundred different groups to create a tag cloud.

Additionally, neighboring co-occurrence profiles are compared to obtain similarity scores

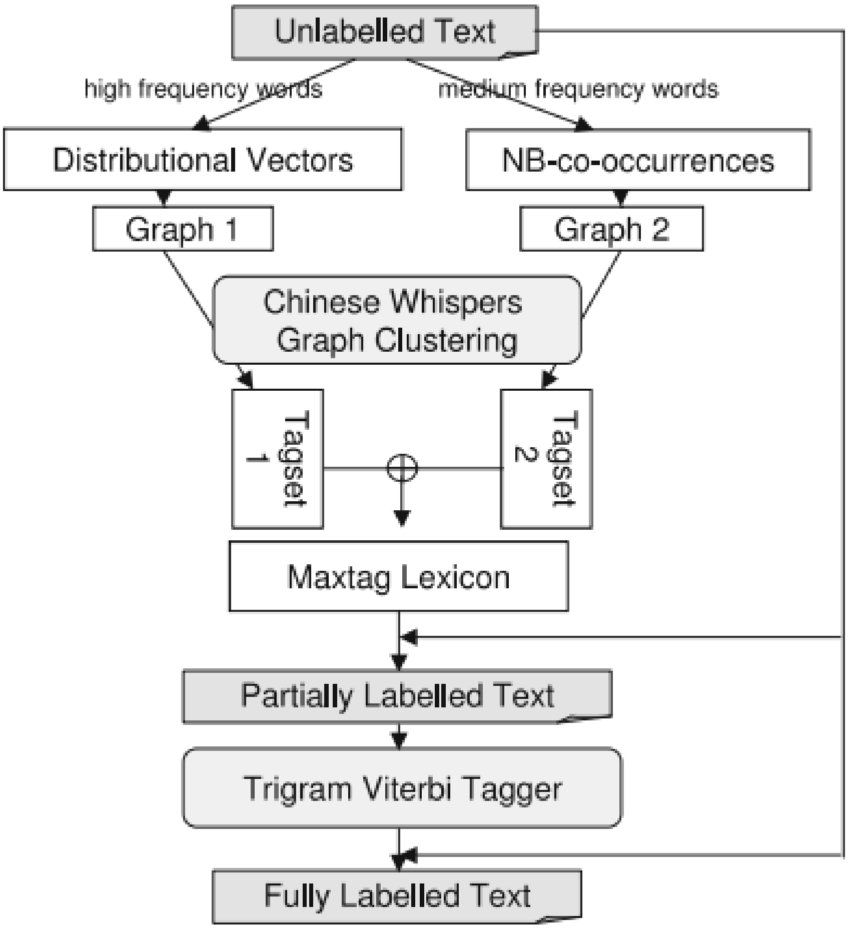


Figure 22. An illustration of the unsupervised POS-tagging process, from unlabelled to partially to fully labeled text[6] .

The integration of the two subsets results in sets of word forms that are members of similar induced syntactic categories as a consequence of the combination. This is because the combination produces the word forms by combining the two partitions. The vocabulary has been expanded by

[6] Taken from: https://link.springer.com/article/10.1007/s11168-010-9067-9 [accessed 05.10.2020]

the addition of ambiguous high-frequency terms that were taken out of tagset 1 due to the fact that they were used so infrequently.

In conclusion, this lexicon is used in the training of a Viterbi trigram tagger that is supplemented by an affix classifier for terms that are not known. The process of unsupervised POS-tagging, which may be shown below in figure 22, begins with text that is not labelled and ends with text that is tagged and labelled.

Tag set 1: Words With a High, Medium, and Low Frequency

In order to generate tagset 1 for both high and medium frequencies, focusing on a text corpus, there are four steps that need to be followed in the correct order.

1. Counting the frequency of 10,000 targets and 200 feature words
2. Constructing a graph based on context statistics

In step 2, an edge is added between two words during the graph construction proces a and b with weight[7] w = 1/(1 − cos($\vec{a}$, $\vec{b}$)), computed using the feature vectors $\vec{a}$, and $\vec{b}$ of letters a and b. Only if w is greater than a certain similarity threshold (s), will the edge be drawn. It is responsible for determining the total number of words that cluster together in the graph that represents the search results.

Clustering words that have a higher similarity threshold should be done whenever it's possible. confidence rather than running the risk of joining two clusters that are not related due to the presence of an excessive number of ambiguous words that connect them. As a consequence of the third step, there is already a portion of the target words that can be interpreted in the manner of a tagset.

A selection of clusters for the Albanian Language Corpus is presented in Table 50. There are a few distinct groups of nouns that can be seen here. When lexical clusters are compared to a gold standard, the results may not be conclusive because the clusters and the gold standard typically have different levels of granularity. For instance, in the singular form of the Albanian language, the

[7] POS induction is commonly measured by cosine similarity, but there are also other options

nouns are grouped together, but the first and last names are kept separate. The tagset that was utilised for the gold standard has a significant impact on the evaluation scores.

Using an information-theoretic measure, we can interpret evaluation scores intuitively: Entropy precision (EP) is a measure of how closely the gold standard classification reflects reality. It is similar to precision in information retrieval. A recall as the number of retrieved instances versus the total number of instances represents how well the clustering algorithm covered the target words. This same gold standard identifies the instance of a word that occurs most frequently. It does not take into account POS ambiguities. Nevertheless, despite all of these drawbacks, EP provides a solution in the form of a comparison of the quality of partitions for different thresholds.

Clusters were chosen based on the clustering of the Albanian Language corpus, with 10,000 words in each partition as shown in Table 50 below. The total number of clusters that are obtained is 928. This partition has an EP value of 1.6552. An Albanian language corpus has been used to collect gold standard tags; example phrases are presented in decreasing frequency order.

Table 33. Clusters were selected from the Albania Language corpus clustering with 10,000 words in each partition. In total, 928 clusters are obtained.

RANK	SIZE	GOLD STANDARD TAGS (COUNT)	DESCRIPTION	SAMPLE WORDS

1	3650	NN1(2816)
		NN2(960)
			Singular nouns	Jetë, nata, banesa, bota, shtëpia, dielli familja, kuvend, zhvillim, ...

2	2650	NN1(1960)
		NN2(898)
			Plural nouns	shqiptarëve, vlera, punëtorë, traifa, copë, fitimtarë, shitëse, mundësi, ...

3	453	NP0(452), NN1(2)	First names	Adem, Afrim, Agoll, Aleksandra, Artë, Blerim, Hajri, Ibrahim, Ilir, Isak, Kadare,

				Myzeqe, ...
4	865	AJ0(865)	Adjectives	Mbikëqyrës, Kaluar, gjerë, juglindor, punëmbarë, shmangshëm, truplidhur, zemërmadh, bërthamor, shtrëngueshëm, ...
		....		

5	1378	NN1(789)
		NN2(589)
			Singular and plural nouns	futbolli , binar, shqyrtimi, shteti, shkruar, letrave, zgjedhjeve, karrierë, zemër, gazetat, veri, ...

6	454	VV(460)	Verbs	Emërtoj, Abstenoj, bjeshkoj, blej, largoj, nxij, shkëput, trokas, sheh, vështrojë, vërsulem, ….
7	65	AJ0(65)	Adjectives (country)	Britanezë, individual, Evropianë, re, Nacional, ...
8	47	AJ(47)	Adjectives (size/quality)	vogë, madhe, mire, fit, disa, ca, speciale, keq, re, vjetër, mjaftë, pak, mesatare, …
9	35	NP0(35), NN1(1)	Countries	Amerika, Franca, Itali, Gjermania, Kosovë, Japoni, Brazil, Kanada, India, Poloni, Kina, Britani, Bahamat,

				Greqi, ...
		....		
10	25	NP0(25)	Cities	Tetovë, Durrës, Gjakovë, Kavajë, Tiranë, Krujë, Lushnjë, Gjilan, Ferizaj, Mitrovicë , Ohër, ...
11	14	NN2(14)	Plural professions	Arsimtarët, profesorët, shkenctarët, bujqit, guvernatorët, punëtorët,....
12	12	CRD(12)	Numbers	Një, dy, trembëdhjetë, katër, shtatëmbëdhjetë, tetëdhjetë, nëntë, dhjatë, njëmbëdhjetë, ...
13	10	NP0(10)	Titles	Zonjë, Znj, Zt, Zotëri, Teze, Dajë, Mixhë, Dr., MSc, PhD,....

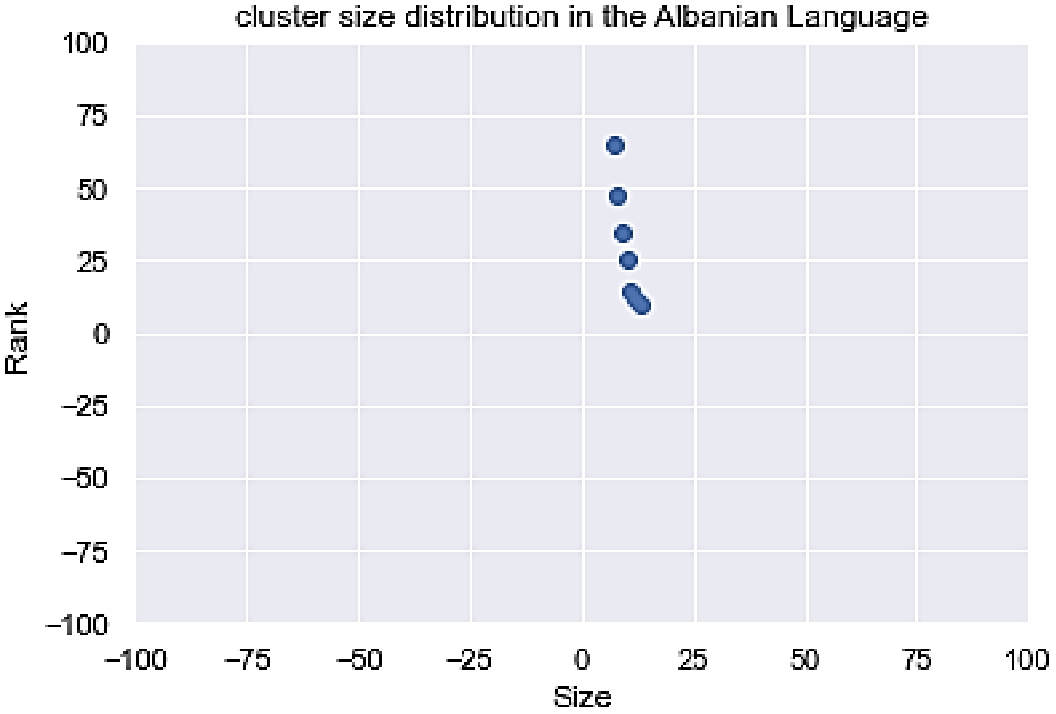


Figure 23. Cluster size distribution for tagset in the Albanian Language, ordered by decreasingly by cluster size (rank)

Half of the phrases in the sample lexicon were selected at random, and the other half were chosen by hand in order to provide space for a wider range of linguistic phenomena. This was done in order to accommodate. Nouns, verbs, pronouns, adjectives, adverbs, prepositions, conjunctions, particles, numerals, and interjections are the ten different parts of speech that are recognized in the Albanian language. The only way to differentiate between the two is by considering the context.

Noun: There are four morphological categories for Albanian nouns: Every noun has a gender, but the majority are feminine in nature most are masculine, while a few are neuter. Many nouns are heterogeneous, which means they have different genders in the plural and singular forms. In Table 34, are represented the Albanian noun tags. Consider the following example: 'Të rinjtë janë nga Maqedonia e Veriut.', Eng. 'The young people are from North Macedonia.', which is analyzed as Të\Art rinjtë\NArt janë\V nga\Prep Maqedonia e Veriut\Nm.\Punct.

Table 34. Descriptions of Albanian noun tags

#	Tag	Name	Example in Albanian language
1.	N	Noun	premte
2.	NA	Noun preceded by article	e premte
3.	NHg	Het. noun	kuq(sg.m.) vs.kuqe(pl.f.)

Number: There are nouns that can take either the singular or the plural form. Occasionally, the singular and plural have the same meaning, e.g. "një shtëpi" (singular, engl. house) and "disa shtëpi" (plural, engl. houses).

Case: There are five cases: nominative, genitive, dative, accusative, and ablative. The genitive and nouns tend to have the same form, differing only by the preceding article.

Definiteness: On the other hand, as it pertains to definiteness (indefinite and definite): Indefinite forms like (një) vajzë - Eng.(one/a) girl, can be distinguished by the definite form, vajza - Eng. (the) girl.

Nouns in the Albanian language are categorised as per their gender, which can be feminine, masculine, or neutral, and their number, which can be singular or plural; however, in the vast majority of instances, nouns change gender when they are expressed in the plural form.

They are referred to as heterogeneous nouns because, for example, the word mësim/mësimi is masculine in the singular but feminine in the plural (mësime/mësimet). Because of this phenomenon, testing for congruence that does not include a morphological component can be challenging. The tags for Albanian nouns are presented in table 35.

Table 35. Albanian language proposed noun tags

Number	Name	Tag	Example in Albanian Language
1	Noun	No	mërkure
2	Noun word prep. art	NArt	mërkure; (të) rinjtë
3	Heterogeneous Noun	HgsNo .	mësim, -i sg. m. vs. -e, -et pl. f.
4	Name	Nm	Shkupi; Ana; Ola;

The following example illustrates this: 'Të rinjtë janë nga Gjakova.', Eng. 'The young people are from Gjakova.', which is analyzed as Të\Art rinjtë\NArt janë\V nga\Prep Gjakova\Nm .\Punct

Adjective: In terms of morphology, adjectives are gendered, numbered, and cased depending on the related noun. Additionally, adjectives have a degree of gradation. A positive grade, a comparative grade, and a superlative grade are assigned in Albanian morphology. Albanian adjectives describe a person or thing, which should also correspond to the gender and number of the noun.

Considering an example with a masculine and feminine gender: "Ky është Batoni, vëllau im." - Eng. "This is Baton, my brother."; whereas for feminine, it would be: "Kjo është Gea, vajza ime." - Eng. "This is Gea, my daughter. The male singular ky is represented in English by the word 'this' in both genders, while the female singular kjo is represented by the word 'kjo'.

A further difference between the two pronouns is the personal pronoun 'im', in masculine singular, and 'ime' in feminine singular – which in English are represented by gender-neutral 'my'. Albanian adjectives are divided into five categories, including adjectives before nouns, adjectives with proposed articles, and non-inflectional adjectives. As shown in the table below.

Table 36. Albanian language proposed adjective tags

Number	Name	Tag	Example in Albanian Language
1	Adjective	Adje	[vajza] mençur
2	Proposed adjective	PPAdje	mençuri [vajzë]
3	Adjective word article	AdjePpArt	[vajzë] e mirë.
4	PPAdj. word article	PPAdjePPArt	e mira [vajzë]
5	Noninflected adjective	AdjNIfe	blu/neto

Positive, comparative, and superlative adjectives are classified as categories in the Albanian language [7]. Adding the comparative article to the base word creates escalation ‘më’, e.g. (1) positive: shpejtë - Eng. ‘fast, (2) comparative: ‘më shpejt - Eng. ‘faster and (3) superlative: ‘më i shpejti Eng. ‘the fastest.

Numerals: In the Albanian language, the numbers are categorized as ordinal numbers and cardinal numbers. A nominal number has the same characteristics as an adjective, except for escalation and a definite article.

Table 37. The Albanian language suggested numeral tags

Number	Name	Tag	Example in Albanian Language
1	Cardinal number	NumCa	tre [fitore]
2	Ordinal number	NumOr	[fitorja] e pestë

Consider the following example: ‘Kjo ishte fitorja e saj e pestë brenda një muaji.’ - Eng. ‘This was her fifth victory within one month.’, which is tagged as Sot\Adv ishte\V fitorja\N e\Art pestë\NumO e\Art saj\PossPr brenda\Prep një\NumC muaji\No.\Punc.

Pronoun: In Albanian, pronouns are broken down into subtypes according to the level of specificity they possess. Additionally, a relative pronoun cannot be interchanged with an interrogative pronoun or a personal pronoun. Each specialised category of pronoun is represented by its own tag in the table16 that follows. It is possible for a proposed article to come before certain pronouns, just as it is possible for it to come before nouns or adjectives.

When used in conjunction with the relative pronoun i cili," which means "which," the interrogative pronoun "cili," which in English means "who," can be transformed into "which." This raises the question of whether or not the article and the pronoun should be considered together.

Table 38. Proposed pronoun tags for Albanian

Number	**Name**	**Tag**	**Example in Albanian Language**
1	Personal pronoun	PersPr	‘t’i
2	Demonstrative pronoun	DemP	‘ky’/’kjo’/’këta’ /’ai’/ ‘ajo’
3	DemonstrativePron w. art	DemPPPArt	‘i tillë’
4	Possesive pronun	PossPr	‘im’, ‘e’, ‘tu’, ‘l’, ‘saj’
5	PossP w. prep. Art.	PossPPPArt	i tij/të vetën
6	Interrogative pron.	IntPr	‘kush’, ‘cili’, ‘cila’ etc.
7	IntP w. art.	IntPPPrArt	‘i kujt’/’i cilit’
8	Relative pronoun	RelaPro	‘që’
9	RelP w. art.	RelPPPrArt	‘i cili’ / ‘e cila’/ ‘të cilat’

10	Indefinite pron.	IndefPr	‘dikush’/ ‘askush’/ ‘ndonjë’
11	Reflexive pron.	ReflPr	‘(më’)’vete’/ ‘vetveten’

When used in conjunction with a noun, the subjective pronouns "ai" and "ajo" can take on the role of demonstrative pronouns. A few pronouns, like some nouns as well as some adjectives, can have an article come before them. In addition to this, there is a condensed version of the subjective pronouns.

Verb: The verb tense in Albanian can be either variable or constant, depending on the context. There are nine different tenses associated with Albanian verbs (imperfect, past, past perfect, past perfect (past), future, future past, and future front); two voices (active and passive); and six different moods associated with Albanian verbs (indicative, subjunctive, conditional, admirative, optative, and imperative).

Table 39. Proposed verb tags for the Albanian language

#	Tag	Name	Example in Albanian language
1.	V	Verb (finite forms)	Tha
2.	VPart	Participle(non-finiteforms)	Thënë
3.	VCl	V.w.clitic	i tha
4.	VImpv	Imperativeform	Prit / Fol
5.	VImpvCl	Imperativew.clitic	Shpjegomëni
6.	VPass	V.w.pass.part.u	U bë
7.	VPassCl	V.w.pass.part.andclitic	Ua tha
8.	VSubj	V.w.subj.particle	Të thotë
9.	VSubjCl	V.w.pass.part.andcl.	Ta tha
10.	VSubjPass	V.w.subj.a.pass.part	T’u thotë

11.	VAux	Auxiliar verb	kam
12.	VMod	Modal verb	mund
13.	VRecp	Reciprocal verb	njihen
14.	VRef	Reflexive verb	ushqehem

The past participle, the infinitive, the present participle, and the negative are the four fundamental forms of a verb [13]. In light of all of these characteristics, it is possible for us to draw the conclusion that a single verb can take on 47 distinct inflectional forms.

The verb tags for the Albanian language are shown in Table 39.

Table 40. POS examples from the Albanian corpus

Word	**Part of Speech**
Angli	Noun
fillore	Noun
filozofi	Noun
……...	
kërkues	Adjective
zemërmadh	Adjective
buzëqeshur	Adjective
………	
lexoj	Verb
ngatërroj	Verb
prezentoj	Verb
afërmi	Adverb

ngrohtë	Adverb
kolektivisht	Adverb
………	
obobo	Interjection
ore	Interjection
ose	Interjection
………	
përballë	Preposition with ablative
përgjatë	Preposition with ablative
rreth	Preposition with ablative

Inflections of verbs are grouped into three categories: first inflection verbs (verbs ending with -j in their singular 1st person indicative present tense active voice form, e.g. 'punoj'), second inflection verbs (those ending in consonants, e.g. 'shoh'), and third inflection verbs (those ending in vowels, e.g., 'ha'). Example of Parts of speech are shown in table 40 from the Albanian corpus.

Table 41. Identifying morphological tags, text segmentations, and lemmatization in the Albanian language

Target	**Accuracy %**
Token	97.85%
Sentences	97.92%
Part of Speech	93.65%
Lemmas	88.95%
Features	85.31%

A number of factors are evaluated and ranked according to the degree of accuracy they provide, including token and sentence segmentation, part-of-speech tagging, and morphological analysis, characteristics and lemmatization

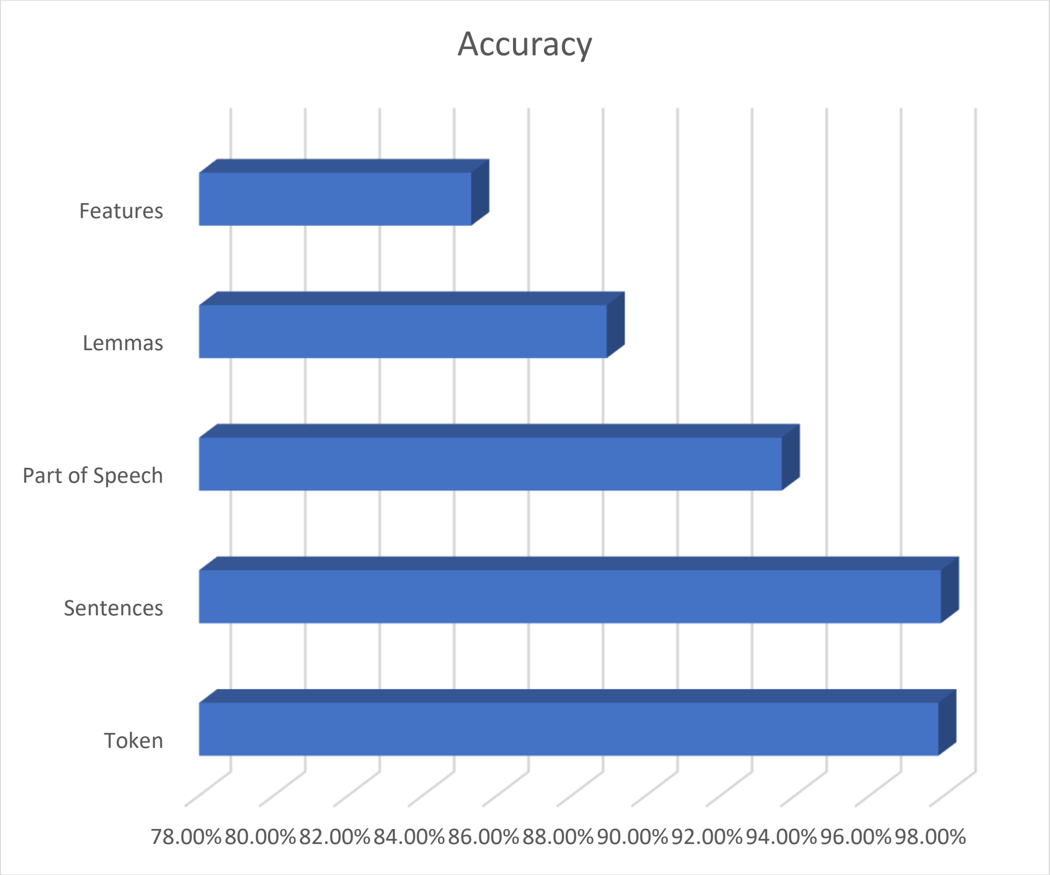
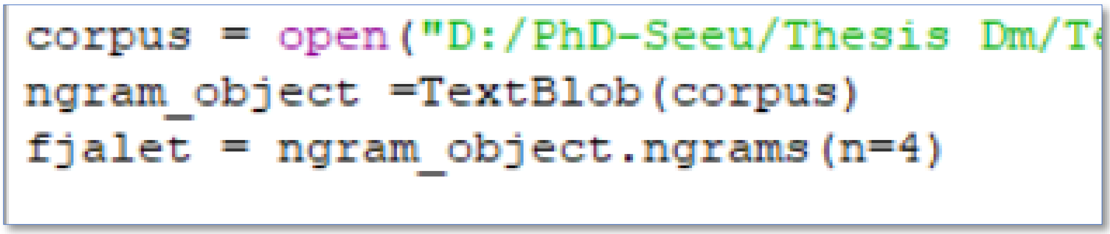


Figure 24. Analysis of the task of morphological tagging, text segmentation, and lemmatization of Albanian

The following Fig 24. presents statistics regarding the tagging of parts of speech. The frequency of occurrences of nouns, verbs, and adverbs in Albanian language texts.

5.1 Application of known unsupervised methods

In this section we present experiments and results generated by the Albanian dictionary. We have constructed a matrix in which the columns contain the 100 most frequently occurring words from the dictionary and the rows contain all words from the dictionary.



```
corpus = open("D:/PhD-Seeu/Thesis Dm/Te
ngram_object =TextBlob(corpus)
fjalet = ngram_object.ngrams(n=4)
```

Figure 25. Python script for ngrams

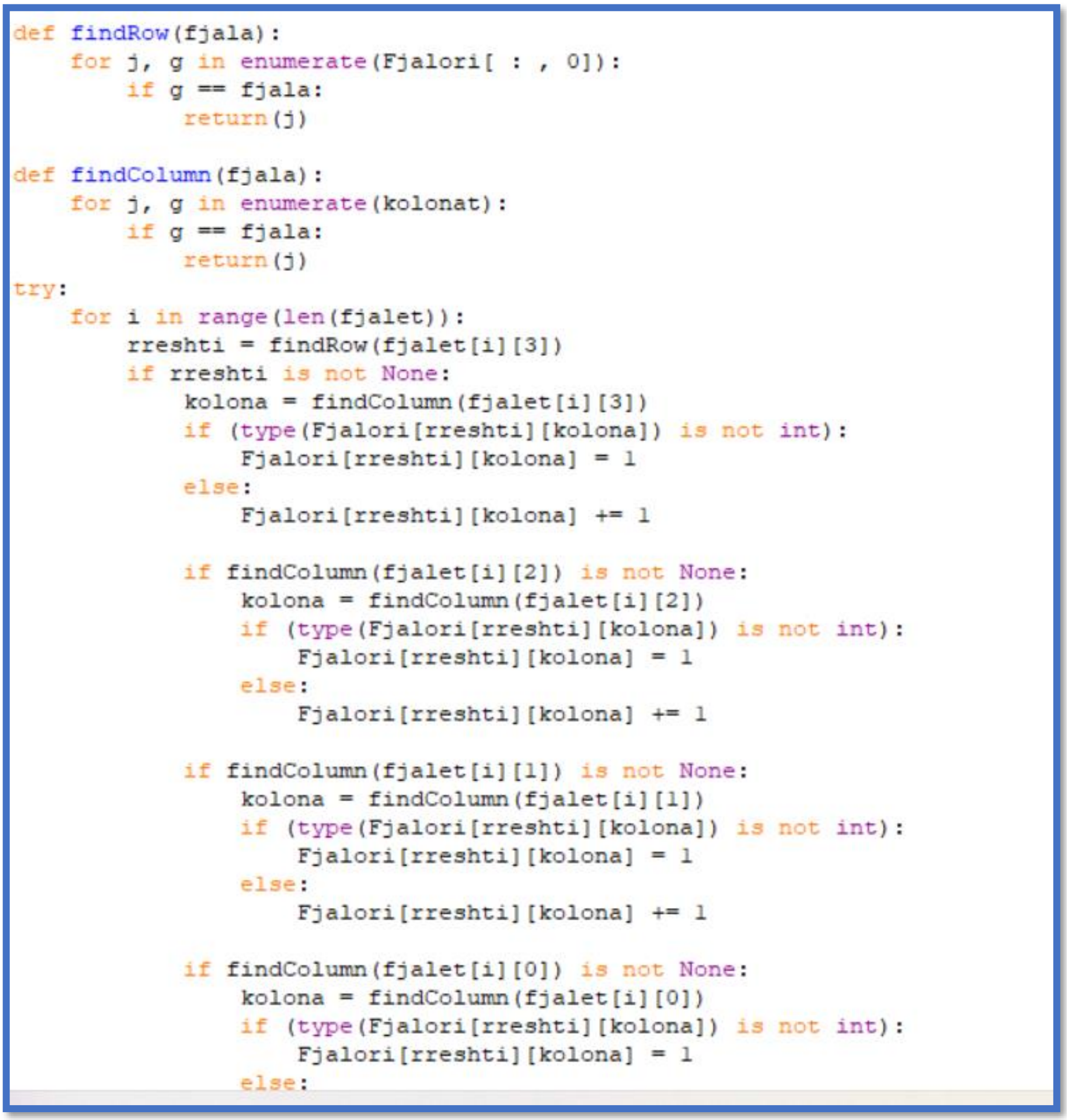
First, we gathered all of the material that had a capacity of 80 MB and then used N-gram to break it up into four sentences. Following the execution of the script indicated in Figure 25, we obtained the data shown in Table 42 below.

Table 42. Sentence using ngram=4

Sentence:
‘Me ndihmën e llogaritjeve komplekse që mund të bëhen vetëm me anë të kompjuterëve’
Results:
‘Me ndihmën e llogaritjeve’
‘ndihmën e llogaritjeve komplekse’
‘e llogaritjeve komplekse që’
‘llogaritjeve komplekse që mund’
‘komplekse që mund të’
‘që mund të bëhen’
‘mund të bëhen vetëm’
‘të bëhen vetëm me’

‘bëhen vetëm me anë’
‘vetëm me anë të’

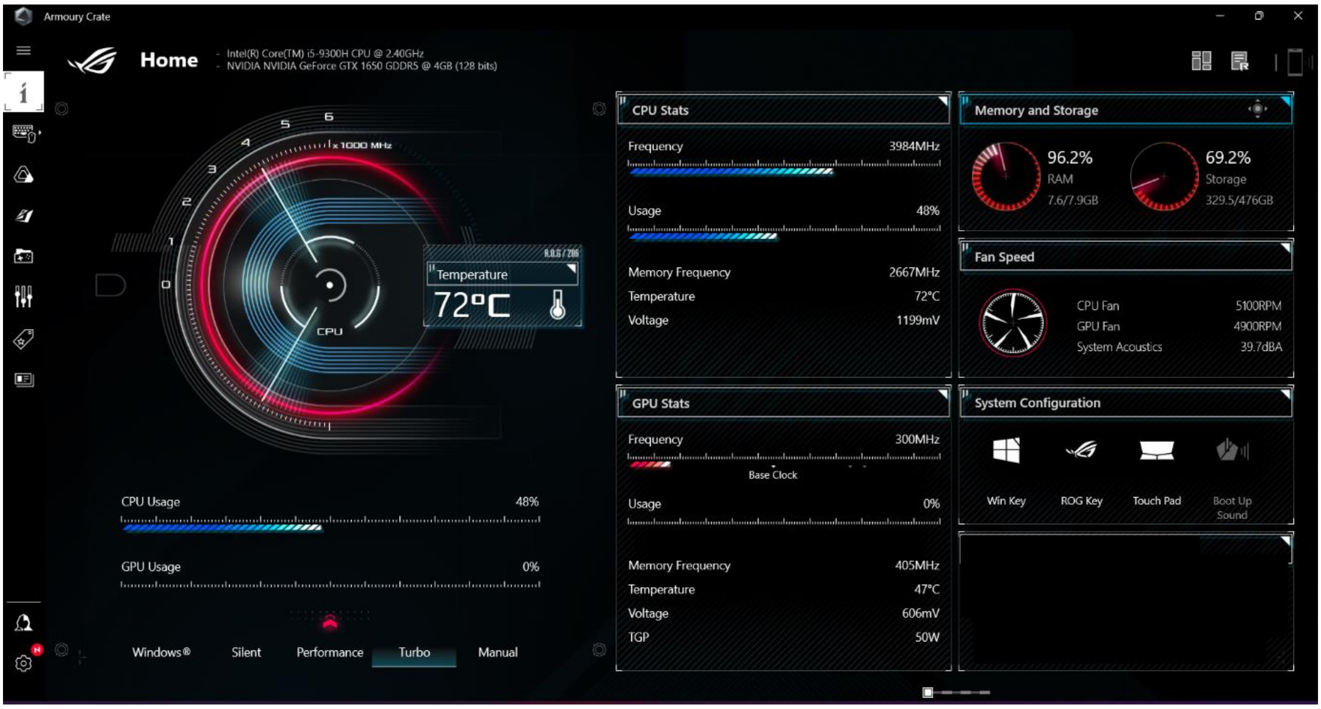
Python libraries such as pandas, NumPy, sys, and os have been used throughout the matrix figure 26 illustrates how the implementation process works. We have obtained the results after executing this script.



```
def findRow(fjala):
    for j, g in enumerate(Fjalori[ : , 0]):
        if g == fjala:
            return(j)

def findColumn(fjala):
    for j, g in enumerate(kolonat):
        if g == fjala:
            return(j)
try:
    for i in range(len(fjalet)):
        rreshti = findRow(fjalet[i][3])
        if rreshti is not None:
            kolona = findColumn(fjalet[i][3])
            if (type(Fjalori[rreshti][kolona]) is not int):
                Fjalori[rreshti][kolona] = 1
            else:
                Fjalori[rreshti][kolona] += 1

            if findColumn(fjalet[i][2]) is not None:
                kolona = findColumn(fjalet[i][2])
                if (type(Fjalori[rreshti][kolona]) is not int):
                    Fjalori[rreshti][kolona] = 1
                else:
                    Fjalori[rreshti][kolona] += 1

            if findColumn(fjalet[i][1]) is not None:
                kolona = findColumn(fjalet[i][1])
                if (type(Fjalori[rreshti][kolona]) is not int):
                    Fjalori[rreshti][kolona] = 1
                else:
                    Fjalori[rreshti][kolona] += 1

            if findColumn(fjalet[i][0]) is not None:
                kolona = findColumn(fjalet[i][0])
                if (type(Fjalori[rreshti][kolona]) is not int):
                    Fjalori[rreshti][kolona] = 1
                else:
```

Figure 26. Python script for implementing matrix

Our ability to manage the file collection was limited by the amount of RAM memory that we had available. The computer had the following specifications: Intel(R) Core(TM) i5-9300H 2.40GHZ with the capacity of RAM memory DDR4, 8GB, 266MHz. However, it was unable to perform the whole operation of the Python script due to the fact that it ran out of memory. The amount of CPU and Memory that was used while the script was being executed is seen in Figure 27.



- Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz
- NVIDIA NVIDIA GeForce GTX 1650 GDDR5 @ 4GB (128 bits)

Temperature 72°C

CPU Usage 48%

GPU Usage 0%

CPU Stats

Frequency 3984MHz

Usage 48%

Memory Frequency 2667MHz

Temperature 72°C

Voltage 1199mV

Memory and Storage

96.2% RAM 7.6/7.9GB

69.2% Storage 329.5/476GB

Fan Speed

CPU Fan 5100RPM

GPU Fan 4900RPM

System Acoustics 39.7dBA

GPU Stats

Frequency 300MHz

Usage 0%

Memory Frequency 405MHz

Temperature 47°C

Voltage 606mV

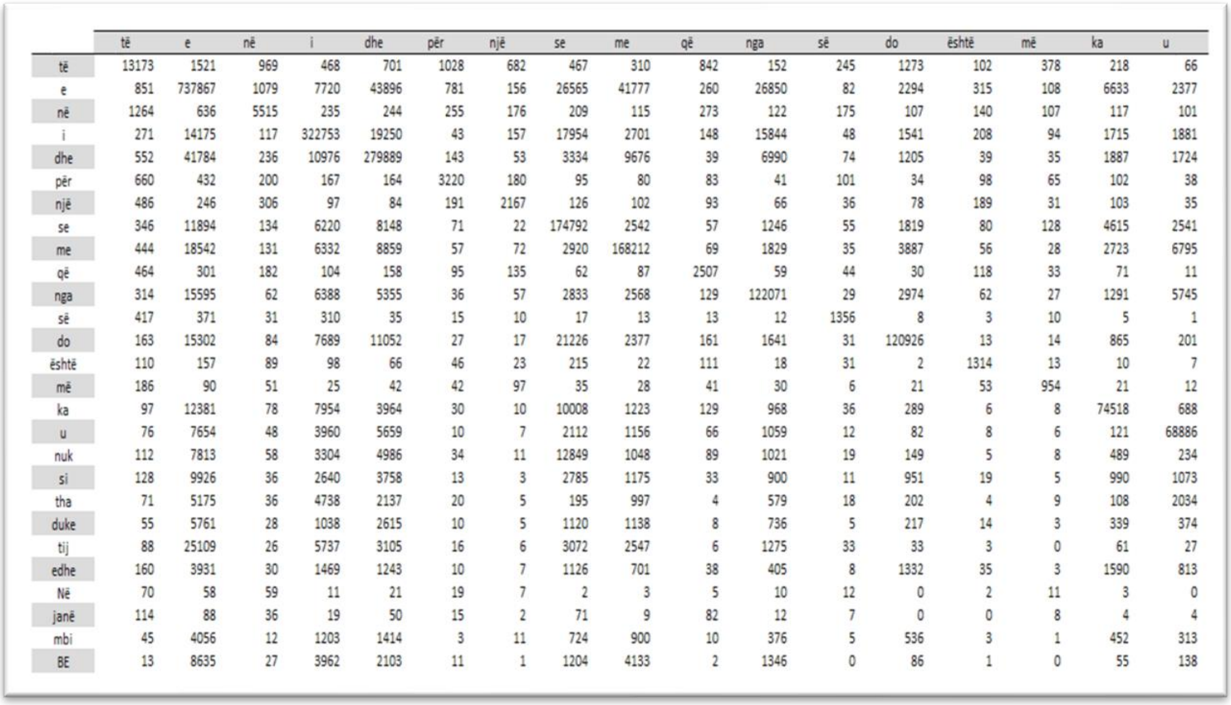
TGP 50W

System Configuration

Figure 27. Execution process in Intel(R) Core (TM) i5-9300H

All of the text collections that belong to the Albanian language are shown in the form of a matrix in Table 43. With these findings in hand, we can go on to the next stage of the process, which involves determining the cosine similarity between the terms found in the dictionary. As an example, the letter 'të' occurs 13173 times in the matrix, but the letter 'e' is found 737867 times.

Table 43. Matrix of the Albanian language dictionary



	të	e	në	i	dhe	për	një	se	me	që	nga	së	do	është	më	ka	u
të	13173	1521	969	468	701	1028	682	467	310	842	152	245	1273	102	378	218	66
e	851	737867	1079	7720	43896	781	156	26565	41777	260	26850	82	2294	315	108	6633	2377
në	1264	636	5515	235	244	255	176	209	115	273	122	175	107	140	107	117	101
i	271	14175	117	322753	19250	43	157	17954	2701	148	15844	48	1541	208	94	1715	1881
dhe	552	41784	236	10976	279889	143	53	3334	9676	39	6990	74	1205	39	35	1887	1724
për	660	432	200	167	164	3220	180	95	80	83	41	101	34	98	65	102	38
një	486	246	306	97	84	191	2167	126	102	93	66	36	78	189	31	103	35
se	346	11894	134	6220	8148	71	22	174792	2542	57	1246	55	1819	80	128	4615	2541
me	444	18542	131	6332	8859	57	72	2920	168212	69	1829	35	3887	56	28	2723	6795
që	464	301	182	104	158	95	135	62	87	2507	59	44	30	118	33	71	11
nga	314	15595	62	6388	5355	36	57	2833	2568	129	122071	29	2974	62	27	1291	5745
së	417	371	31	310	35	15	10	17	13	13	12	1356	8	3	10	5	1
do	163	15302	84	7689	11052	27	17	21226	2377	161	1641	31	120926	13	14	865	201
është	110	157	89	98	66	46	23	215	22	111	18	31	2	1314	13	10	7
më	186	90	51	25	42	42	97	35	28	41	30	6	21	53	954	21	12
ka	97	12381	78	7954	3964	30	10	10008	1223	129	968	36	289	6	8	74518	688
u	76	7654	48	3960	5659	10	7	2112	1156	66	1059	12	82	8	6	121	68886
nuk	112	7813	58	3304	4986	34	11	12849	1048	89	1021	19	149	5	8	489	234
si	128	9926	36	2640	3758	13	3	2785	1175	33	900	11	951	19	5	990	1073
tha	71	5175	36	4738	2137	20	5	195	997	4	579	18	202	4	9	108	2034
duke	55	5761	28	1038	2615	10	5	1120	1138	8	736	5	217	14	3	339	374
tij	88	25109	26	5737	3105	16	6	3072	2547	6	1275	33	33	3	0	61	27
edhe	160	3931	30	1469	1243	10	7	1126	701	38	405	8	1332	35	3	1590	813
Në	70	58	59	11	21	19	7	2	3	5	10	12	0	2	11	3	0
janë	114	88	36	19	50	15	2	71	9	82	12	7	0	0	8	4	4
mbi	45	4056	12	1203	1414	3	11	724	900	10	376	5	536	3	1	452	313
BE	13	8635	27	3962	2103	11	1	1204	4133	2	1346	0	86	1	0	55	138

5.1.1 Cosine Similarity

Cosine similarity is one of the metrics used in Natural Language Processing to evaluate the degree to which the text of two documents of different lengths are similar to one another. A word is equivalent to a vector in this representation. Word embeddings are vectors that are represented in n dimensions, and there are a total of n.

Calculating the cosine similarity between two n-dimensional vectors that are projected into a multi-dimensional environment is the purpose of the mathematical metric known as cosine similarity. When comparing two papers, the cosine similarity will fall somewhere between 0 and 1. When comparing two vectors, if the Similarity measure score is 1, it indicates that the direction of both vectors is identical. The greater the distance between the value and zero, the greater the difference in similarity between the two papers [7].

A well-known illustration of this is when we take the vector Mbret, remove the vector Burr, then add the vector Grua. Mbretresha is the closest matching vector towards the generated vector.

We may apply the same approach to lengthier sequences, such as sentences or paragraphs, and discover that identical meaning correlates to vector proximity/orientation.

The well-known arithmetic example in which Mbretëresha = Mbret — Burrë + Grua in the figure 28 below.

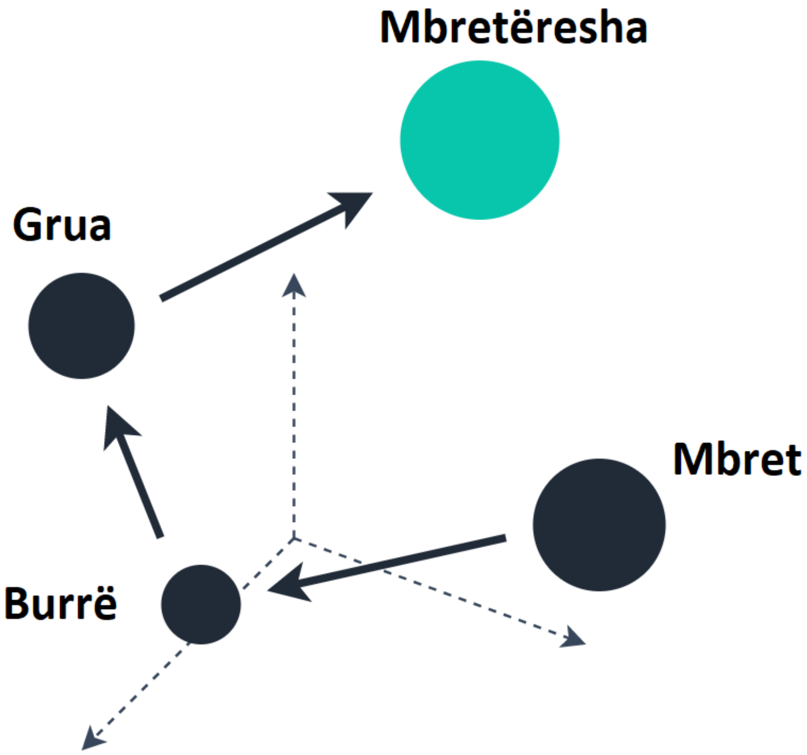


Figure 28. An example of a cosine similarity vector

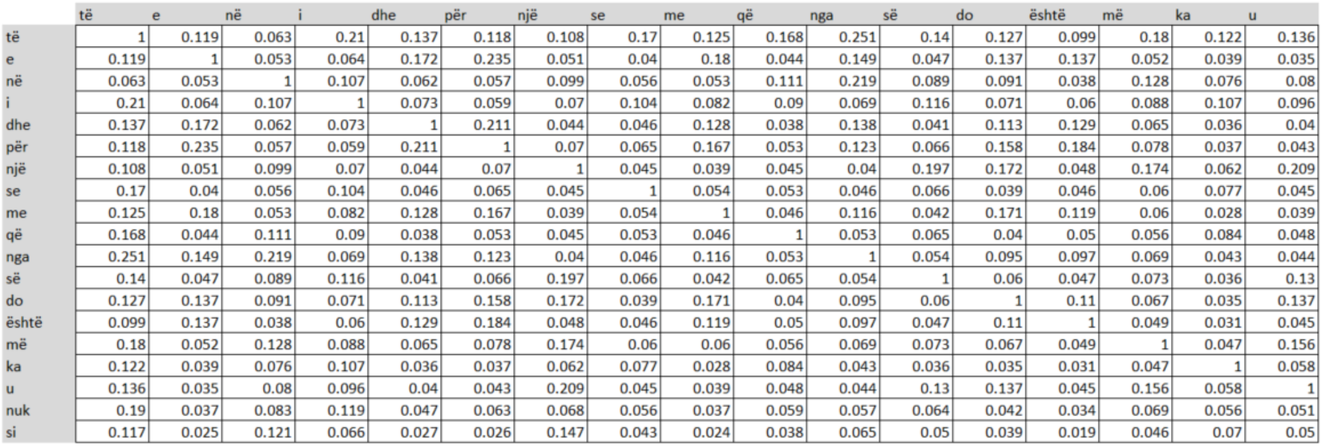
Both A and B are considered to be vectors in the first formula that was presented. The numerator shows the dot product, also known as the scalar product, while the denominator shows the magnitude of each vector. Since separating the linear combination by the magnitude, we arrive at the cosine of the angle that they make between themselves.

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

(1)

The mathematical expression for the cosine similarity between two vectors that are not zero is the formula (1). In natural language processing, cosine similarity may be highly helpful for a variety of different applications. This area includes a wide variety of operations like as question-answering, summarizing documents, and the Semantic Textual Similarity (STS) measurement. In the field of NLP, this is considered a core concept. Using Python as our programming language of choice, we were able to successfully construct the Cosine similarity formula described before. We determined that 10,000 words had a storage capacity of 548 megabytes by using a dictionary.

Table 44. Matrix calculating cosine similarity



	të	e	në	i	dhe	për	një	se	me	që	nga	së	do	është	më	ka	u
të	1	0.119	0.063	0.21	0.137	0.118	0.108	0.17	0.125	0.168	0.251	0.14	0.127	0.099	0.18	0.122	0.136
e	0.119	1	0.053	0.064	0.172	0.235	0.051	0.04	0.18	0.044	0.149	0.047	0.137	0.137	0.052	0.039	0.035
në	0.063	0.053	1	0.107	0.062	0.057	0.099	0.056	0.053	0.111	0.219	0.089	0.091	0.038	0.128	0.076	0.08
i	0.21	0.064	0.107	1	0.073	0.059	0.07	0.104	0.082	0.09	0.069	0.116	0.071	0.06	0.088	0.107	0.096
dhe	0.137	0.172	0.062	0.073	1	0.211	0.044	0.046	0.128	0.038	0.138	0.041	0.113	0.129	0.065	0.036	0.04
për	0.118	0.235	0.057	0.059	0.211	1	0.07	0.065	0.167	0.053	0.123	0.066	0.158	0.184	0.078	0.037	0.043
një	0.108	0.051	0.099	0.07	0.044	0.07	1	0.045	0.039	0.045	0.04	0.197	0.172	0.048	0.174	0.062	0.209
se	0.17	0.04	0.056	0.104	0.046	0.065	0.045	1	0.054	0.053	0.046	0.066	0.039	0.046	0.06	0.077	0.045
me	0.125	0.18	0.053	0.082	0.128	0.167	0.039	0.054	1	0.046	0.116	0.042	0.171	0.119	0.06	0.028	0.039
që	0.168	0.044	0.111	0.09	0.038	0.053	0.045	0.053	0.046	1	0.053	0.065	0.04	0.05	0.056	0.084	0.048
nga	0.251	0.149	0.219	0.069	0.138	0.123	0.04	0.046	0.116	0.053	1	0.054	0.095	0.097	0.069	0.043	0.044
së	0.14	0.047	0.089	0.116	0.041	0.066	0.197	0.066	0.042	0.065	0.054	1	0.06	0.047	0.073	0.036	0.13
do	0.127	0.137	0.091	0.071	0.113	0.158	0.172	0.039	0.171	0.04	0.095	0.06	1	0.11	0.067	0.035	0.137
është	0.099	0.137	0.038	0.06	0.129	0.184	0.048	0.046	0.119	0.05	0.097	0.047	0.11	1	0.049	0.031	0.045
më	0.18	0.052	0.128	0.088	0.065	0.078	0.174	0.06	0.06	0.056	0.069	0.073	0.067	0.049	1	0.047	0.156
ka	0.122	0.039	0.076	0.107	0.036	0.037	0.062	0.077	0.028	0.084	0.043	0.036	0.035	0.031	0.047	1	0.058
u	0.136	0.035	0.08	0.096	0.04	0.043	0.209	0.045	0.039	0.048	0.044	0.13	0.137	0.045	0.156	0.058	1
nuk	0.19	0.037	0.083	0.119	0.047	0.063	0.068	0.056	0.037	0.059	0.057	0.064	0.042	0.034	0.069	0.056	0.051
si	0.117	0.025	0.121	0.066	0.027	0.026	0.147	0.043	0.024	0.038	0.065	0.05	0.039	0.019	0.046	0.07	0.05

The cosine similarity of the words is shown in Table 44, which can be found by looking at the output; the value for the word "të" is 1, the similarity for the word "e" is 0.119, the similarity for the word "në" is 0.063, and so on. According to what we learned in the theory, if the cosine similarity is close to one, it means that the two vectors are quite similar to one another.

5.1.2 Experiments with Chinese Whispers Clustering Algorithm

During clustering, items are grouped according to their similarity. There are many different uses for clustering in the field of Natural Language Processing (NLP), which stands for Computational Linguistics. The most commonly used types are document clustering, which is used in retrieval applications, and word clustering, which locates clusters of similar words or idea hierarchies.

NLP typically has several thousand features, out of which only a small number are correlated at any given time - for example, compare the number of unique words to the number of words that appear in a sentence – dimensionality reduction techniques have the potential to significantly reduce complexity while maintaining a high level of accuracy.

Characterizing language objects has always relied on feature vectors. It is possible to think of these feature vectors as points in a multidimensional space. Clustering is based on distance metrics, such as the cosine of the angle formed by two vectors.

An alternative to the space-dimensioned representation is the graph representation. Graphs represent objects (represented as nodes) and their connections (represented as edges). The field of natural language processing (NLP) encompasses a wide variety of structures that can be represented organically as graphs. Lexicological-semantic wordnets, dependency trees, and co-occurrence graphs are some examples of these structures.

There is no distance metric when clustering items in a multidimensional space: rather, the similarities between objects are recorded in the edges of each of them. Since it is impossible to compare two objects that do not have a common edge, many optimization strategies have been developed. Because a graph does not have a centroid or a "average cluster member," approaches that rely on centroids cannot be used.

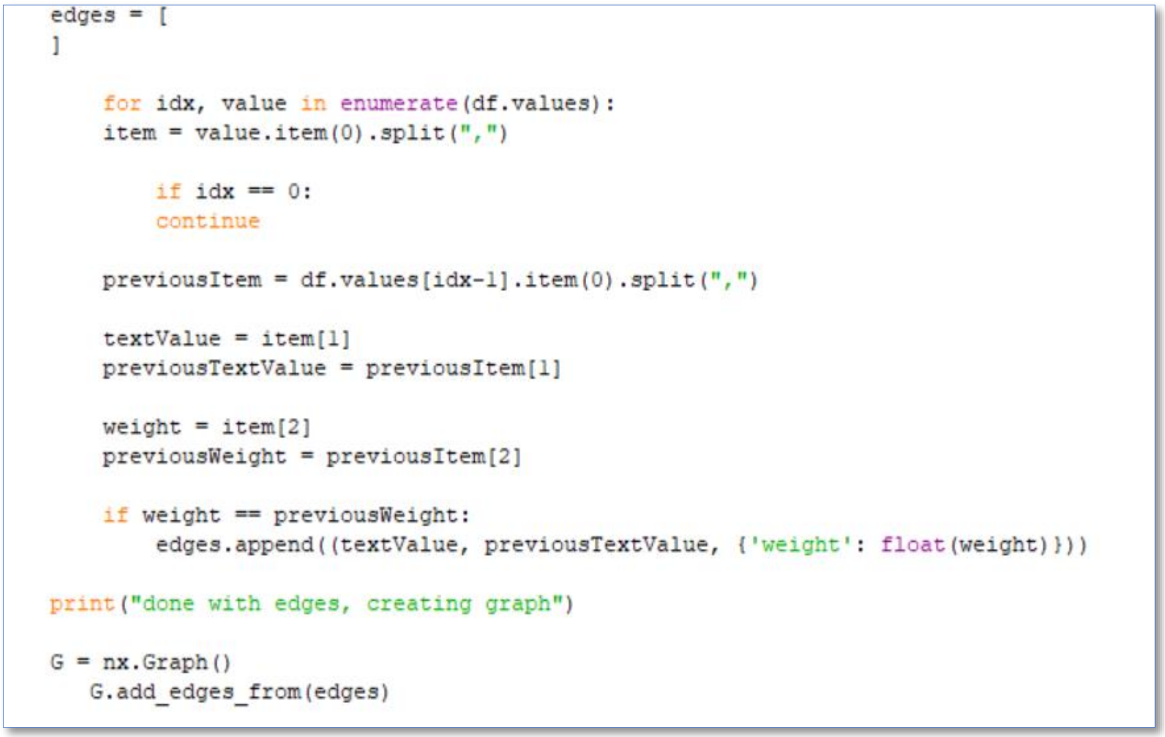
The chronological order in which Chinese whispers unfolds is perhaps its most notable advantage. This is possible due to the fact that the processing time rises linearly with the number of nodes in the network. As a result, the method is able to discover communities in a network in a fairly timely manner.

Because of this, Chinese whispers are an effective method for analysing community structures in graphs that have an extremely large number of nodes. There are a number of subfields within network science that make use of Chinese whispers. The majority of the time, concerns with natural language processing are brought up while discussing this topic. On the other hand, the technique may be used to solve any community identification issue that is connected to network architecture.

All of the experiments that were carried out using the unsupervised Chinese Whispers method are detailed in this section. In these studies, we made use of data from the corpus.
The section consists of:

1. Graph-clustering is an efficient algorithm for partitioning extremely large graphs quickly.
2. A Chinese Whispers game that is implemented as a randomised graph-clustering algorithm with such a time-linear number of edges.
3. Chinese Whispers Clustering is performed for all word’s categorization.
4. Each node has a class and communicates it to its neighbours
5. Nodes adopt the class of the majority of their neighbours if iterations are performed in random order

Python is the programming language that is utilised for the implementation of the Chinese Whispers algorithms. The libraries networkx, metplot, Chinese Whispers, subprocess, and shuffle are the ones that we've used to do this. The next figure, Figure 29, illustrates several components of the Python script.



```
edges = [
]

    for idx, value in enumerate(df.values):
    item = value.item(0).split(",")

        if idx == 0:
        continue

    previousItem = df.values[idx-1].item(0).split(",")

    textValue = item[1]
    previousTextValue = previousItem[1]

    weight = item[2]
    previousWeight = previousItem[2]

    if weight == previousWeight:
        edges.append((textValue, previousTextValue, {'weight': float(weight)}))

print("done with edges, creating graph")

G = nx.Graph()
   G.add_edges_from(edges)
```

Figure 29. Chinese Whispers Algorithm Python script

We have obtained the results after executing this script represented in the Figure 27. The following findings were obtained using an unsupervised Chinese Whispers Algorithm, and they are shown in Figure 30 below.

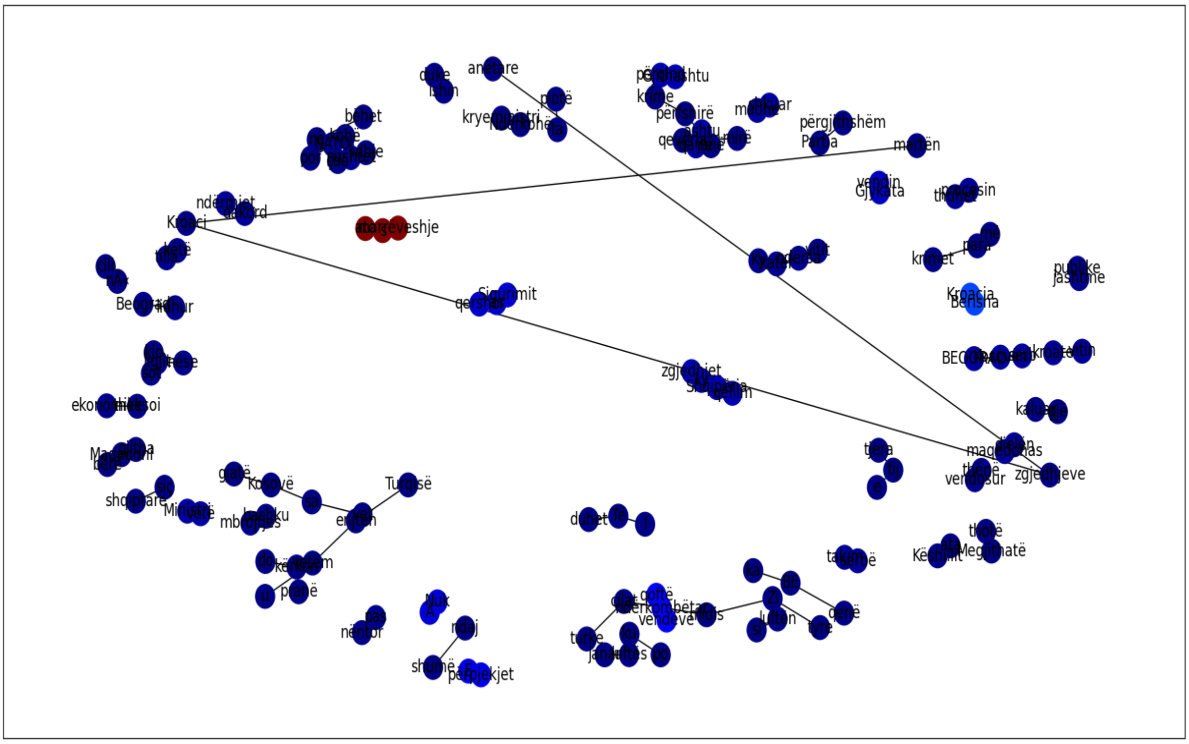


Figure 30. Chinese Whispers Algorithm implementation in the Albanian language corpus

A list of clusters is generated from the vocabulary of four hundred words, and it can be found in Table 45.

Table 45. Cluster List of 500 words

500 Words = 187 Clusters	
ID	Cluster
442	{'u', 'por', 'enjten', 'Republikës', 'kohë', 'sa', 'gjitha', 'së', 'dhe', 'Turqisë', 'vetëm', 'vet', 'Maqedoni', 'bërë', 'ai', 'gjatë', 'Kosovë'}

51	{'institucionet', 'integrimit', 'serbe', 'tani', 'Shkup', 'Gjykata', 'vendin', 'qeverisë', 'dhënë', 'grek', 'këtaj'}
298	{'SARAJEVË', 'rumun', 'ekonomike', 'prill', 'theksoi', 'jenë', 'marrëdhëniet', 'përpara', 'takua', 'ajo', 'çështjet'}
	………………
208	{'partive', 'krimeve', 'Ky', 'Komisionit', 'vitit', 'ndërsa', 'milion', 'katër', 'numër', 'njerëzit'}
5	{'marrëveshje', 'ish', 'vendeve', 's', 'ndërkombëtar', 'ato', 'qoftë', 'të', 'Kroacia'}
65	{'qe', 'ky', 'qëllim', 'përqind', 'Shqipëria', 'zgjedhjet', 'Gjithashtu', 'jemi', 'kërkuar'}
	………………
106	{'tu', 'kam', 'njohur', 'Por', 'Beogradit', 'reja', 'Shqipëri', 'vende', 'iu'}
166	{'Jashtëm', 'autoritetet', 'mos', 'aq', 'partisë', 'tek', 'tre', 'shtuar', 'miratoi'}
171	{'vitin', 'maqedonase', 'BUKURESHT', 'Tadiç', 'publik', 'diskutuar', 'Kjo', 'kroate', 'cila'}
	………………
390	{'sot', 'njoftoi', 'kjo', 'plotë', 'kryeministri', 'ta', 'Maqedonisë', 'lufte', 'Ndërkohë'}
458	{'pa', 'tjetër', 'thotë', 'do', 'me', 'mundur', 'kërkon', 'pranë', 'Megjithatë'}
468	{'po', 'qenë', 'pasi', 'tjera', 'luftën', 'ku', 'OSBE', 'ndaj', 'pjesë'}

In the above list, there are 187 clusters. For example, the ID 442 includes the cluster of words {'u', 'por', 'enjten', 'Republikës', 'kohë', 'sa', 'gjitha', 'së', 'dhe', 'Turqisë', 'vetëm', 'vet', 'Maqedoni', 'bërë', 'ai', 'gjatë', 'Kosovë'}while the ID 116 includes the cluster of words such as {'Jashtëm', 'autoritetet', 'mos', 'aq', 'partisë', 'tek', 'tre', 'shtuar', 'miratoi'}', and the ID 468 includes the words {'po', 'qenë', 'pasi', 'tjera', 'luftën', 'ku', 'OSBE', 'ndaj', 'pjesë'} etc. Nodes have classes and communicate them with their neighbors. If iterations are performed in random order, nodes adopt the class of the majority of their neighbors.

Table 46. Cluster List of 1000 words

1000 Words = 263 Clusters	
ID	Cluster
623	{'Kroaci', 'anëtare', 'martën', 'shpejt', 'zgjedhjeve', 'rritje', 'Ministria'}
553	{'tyre', 'turke', 'midis', 'cilat', 'Zi', 'mbështetje'}
9	{'përfaqësuesit', 'qoftë', 'ndërkombëtar', 'vendeve', 'mbështet'}
…………	
88	{'lehtë', 'ky', 'zgjedhjet', 'qëllim', 'Shqipëria'}
142	{'gazetarëve', 'Po', 'megjithatë', 'rol', 'masat'}
243	{'katër', 'ndërsa', 'rrugë', 'Ky', 'vitit'}
…………	
1	{'marrëveshje', 'atij', 's', 'ato'}
36	{'pavarësisht', 'anëtarët', 'ministrisë', 'opozitës'}
56	{'paqes', 'Ka', 'ardhmen', 'Nëse'}
………….	
65	{'kryesisht', 'zyrtarët', 'emër', 'nëpërmjet'}
74	{'rinj', 'Shkup', 'integrimit', 'serbe'}
75	{'persona', 'Gjykata', 'vendin', 'Solana'}

In the above list, there are 263 clusters from 1000 words. For example, the ID 623 includes the cluster of words {'Kroaci', 'anëtare', 'martën', 'shpejt', 'zgjedhjeve', 'rritje', 'Ministria'} while the ID 88 includes the cluster of words such as {'lehtë', 'ky', 'zgjedhjet', 'qëllim', 'Shqipëria'}', and the ID 56 includes the words {'paqes', 'Ka', 'ardhmen', 'Nëse'}, etc.

Nodes have classes and communicate them with their neighbors. If iterations are performed in random order, nodes adopt the class of the majority of their neighbors.

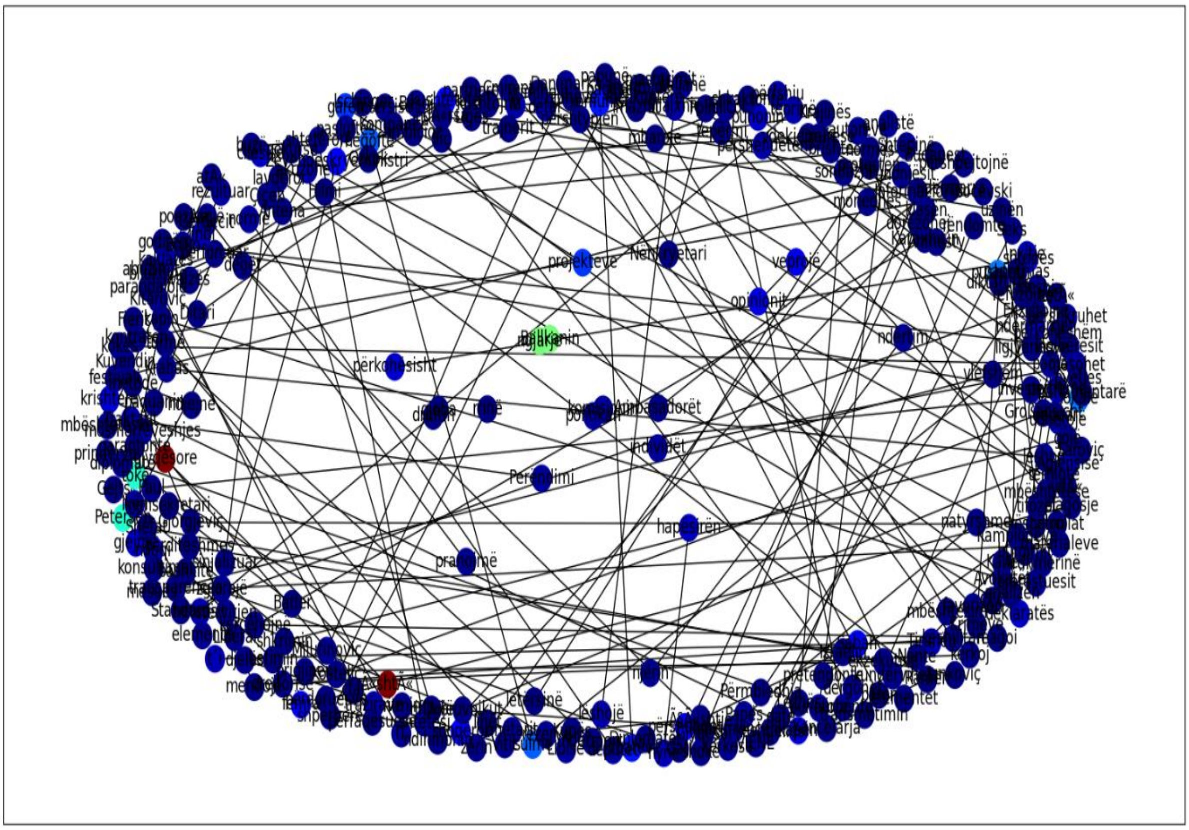


Figure 31. Chinese Whispers Algorithm implementation in the Albanian language corpus

In addition, a second version, which is based on the Albanian lexicon and contains 10,000 words, produces 1,352 clusters. Figure 31 depicted the nodes' relationships to one another via the use of arrows.

Table 47. 1352 Clusters of the Albanina vocabluary

10000 Words = 1,352 Clusters	
ID	Cluster
9036	{'katrore', 'keqen', 'parazgjedhore', 'merrni', 'juridiksionin', 'parregullsive', 'Splitit', 'kamion', 'madhore', 'fuqive', 'verdhë', 'bruto', 'arrestuarit', 'ndodheshin', 'formave', 'lagje', 'pirë', 'gjykimi', 'angazhohen', 'besimeve', 'verore', 'ashtuquajtura', 'Lisbonës', 'sekretarja', 'quhen'}
9936	{'pretendonte', 'përdoruesve', 'Boshnjake', 'këshillon', 'simboleve', 'thelluar', 'trafikimi', 'version', 'ligjvënësve', 'plagosje', 'kriminelë', 'dëbuar', 'shkaktonte', 'mbarojë', 'celularë', 'lajmi', 'pagesës', 'telefonit', 'Fierit', 'raketat', 'ndërmjetës', 'ndërmerr', 'Pirro', 'vizitonte', 'mashtrimit'}
8918	{'Traktatit', 'veprave', 'Nobel', 'Mina', 'njohu', 'Rice', 'idesë', 'mësohet', 'kuptojmë', 'parimeve', 'Perëndimi', 'miq', 'individët', 'popullsi', 'Çka', 'dispozitat', 'zhvillimeve', 'përshpejtimin', 'k', 'argëtuese', 'kthehej', 'autoritetit', 'kurdoherë', 'shkëmbim'}
…………	
2787	{'pastra', 'Grupin', 'vërtetës', 'Fëmijët', 'ulen', 'kundërshtarëve', 'gjuha', 'mbulojnë', 'Kombet', 'pikash', 'Post', 'shoqja', 'bashkinë', 'pozitat', 'Afati', 'Traktatin', 'Stojkoviç', 'pensionet', 'letrën', 'vendoste', 'jugut', 'njëanëshme'}
8089	{'Sistemi', 'shpresë', 'këngët', 'shprehen', 'Mladen', 'Europën', 'materiale', 'Bosnjës', 'kërkimin', 'paraqit', 'sigurisht', 'shkoi', 'gjendja', 'vizituar', 'zhvillimi', 'ska', 'faktorët', 'biznes', 'takojnë', 'tërhoqi'}
8353	{'përmbushen', 'nivelit', 'Çlirimtare', 'refugjatët', 'varur', 'Njeriu', 'pos', 'ndikojë', 'Kthimi', 'skandal', 'nënshkruan', 'Zëri', 'veçse', 'përmirësime', 'tërhoqën', 'krimi', 'ICJ', 'nevojave', 'sugjeron', 'mujor'}
…………	
8666	{'shënuan', 'skandali', 'celular', 'izraelit', 'standarde', 'investim', 'ndihmuan', 'dënohet', 'Batiç', 'shoqëritë', 'fosile', 'tremujorin', 'penalisht', 'dobishëm', 'paparë', 'zjarrin', 'studimeve', 'shpie', 'stilin', 'ndërhyrjen'}
9828	{'rralla', 'miratonte', 'imazhit', 'diktaturës', 'mbështetëse', 'mbështesim', 'meritojnë', 'kostos', 'Cotidianul', 'tepërmi', 'varfëra', 'vizitorët', 'Prania', 'protestën', 'pyetjes', 'Erhard', 'Krimeve', 'Kumanovës', 'tërhiqte', 'letërsinë'}
8723	{'dhuna', 'Sheshelj', 'Shqiptarët', 'shpërthim', 'programet', 'Asambleja', 'Demokratëve', 'Labush', 'prag', 'bashkua', 'shitur', 'votuan', 'hetuar', 'dërgon', 'dëme', 'diskuton', 'vendimeve', 'urdhër', 'Bakojanis'}
…………	
8873	{'jehonë', 'titulluar', 'pagave', 'shkelur', 'panjohur', 'sistemeve', 'përkohësisht', 'Kartës', 'STRASBURG', 'duheshin', 'fizike', 'drejtuesi', 'nisjes', 'dorëzohen', 'krahina', 'kryejë', 'largimin', 'dërgojnë', 'shtunës'}

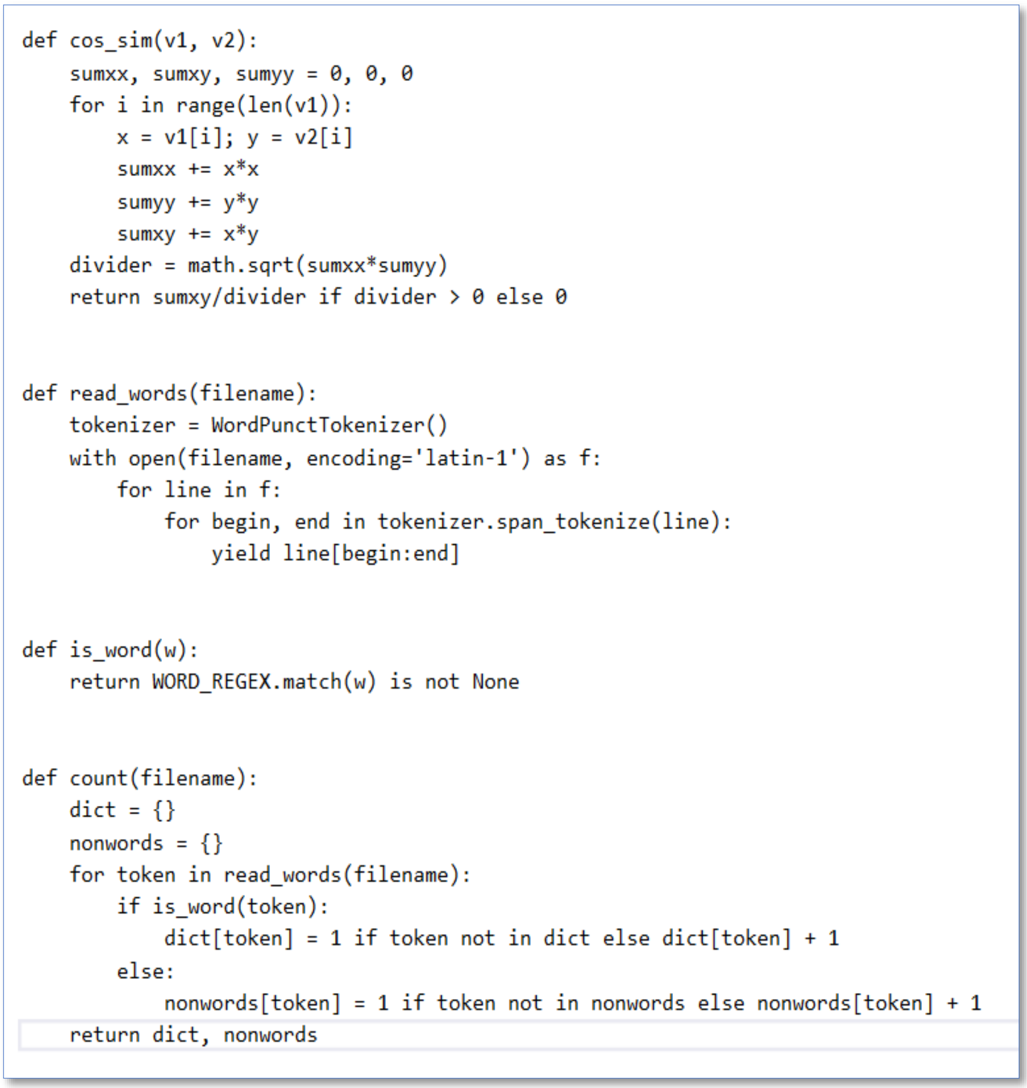
9743	{'Kancelarja', 'Momçillo', 'partnere', 'Ekzistojnë', 'lavdëroi', 'buxhetor', 'rifillojë', 'Hartman', 'normës', 'Antena', 'mbajtura', 'juglindor', 'vajzës', 'qëlluan', 'Ramën', 'monedhës', 'MekIlhani', 'transmetimin', 'Mbretërisë'}
8835	{'vesh', 'sy', 'trajtimin', 'larg', 'shënoi', 'vazhdueshëm', 'ekzistojnë', 'raundin', 'shoqërisë', 'genocid', 'Shkupit', 'plani', 'Presidentin', 'kërkesave', 'ushtrinë', 'Government', 'humanitare', 'tri'}
8996	{'thatë', 'shtynë', 'Sokol', 'Ambasadorët', 'shpirt', 'helikopterë', 'Gordon', 'humbasë', 'bllokojnë', 'rezervat', 'përshpejtojnë', 'gjysmës', 'Aliu', 'shkëmbimit', 'shkurtra', 'meta', 'Përse', 'radikal'}
………….	
9346	{'Gërdecit', 'DPS', 'premtuan', 'arrinin', 'njëherësh', 'zotuan', 'versionin', 'shpjegoi', 'moshës', 'drejtohen', 'rezultuar', 'pozitë', 'Ligjit', 'mosmarrëveshjes', 'Ministrinë', 'Denktash', 'partneritet', 'prokuroria'}
8046	{'rajoni', 'falenderoi', 'avaz', 'shpenzimet', 'konsiderohen', 'pagesë', 'pozitën', 'tanishme', 'shtyn', 'thirrjen', 'këmbëngul', 'moral', 'qytetar', 'dënimit', 'mjek', 'dhjetë', 'Bagdad'}
8612	{'rregullt', 'KosovÃ«s', 'dielës', 'botëror', 'ndihmë', 'Vetëm', 'komisioneri', 'referendum', 'dyja', 'bashkimin', 'vit', 'lëvizje', 'perëndimor', 'turizmit', 'identifikuar', 'raportin', 'hyri'}
………….	
9193	{'Allahun', 'incident', 'kushtetuta', 'lejohet', 'gazetarë', 'bashkiake', 'humbën', 'vepër', 'firma', 'optimist', 'treguan', 'Shekulli', 'mërkurës', 'Aleancën', 'akuzohet', 'lajm', 'Majko'}
9383	{'delegacionin', 'shihen', 'deputet', 'Vojvodinës', 'Lulzim', 'kryesues', 'cilësi', 'ngrirë', 'filmave', 'djalin', 'normë', 'gjeneralëve', 'dërgonte', 'tru', 'traktatit', 'parregullsi', 'llogaritur'}
3571	{'privatizimit', 'ngrenë', 'KB', 'pohojnë', 'islamik', 'Jugosllavinë', 'përcakton', 'incidentit', 'mira', 'hyn', 'ekipeve', 'Bashkimit', 'paralajmëruan', 'komitetin', 'përcaktuara', 'mundi'}

The list represents ID clusters for example the ID 9036 include list of words such as {'katrore', 'keqen', 'parazgjedhore', 'merrni', 'juridiksionin', 'parregullsive', 'Splitit', 'kamion', 'madhore', 'fuqive', 'verdhë', 'bruto', 'arrestuarit', 'ndodheshin', 'formave', 'lagje', 'pirë', 'gjykimi', 'angazhohen', 'besimeve', 'verore', 'ashtuquajtura', 'Lisbonës', 'sekretarja', 'quhen'}. ID 8666 includes the words such as 'shënuan', 'skandali', 'celular', 'izraelit', 'standarde', 'investim', 'ndihmuan', 'dënohet', 'Batiç', 'shoqëritë', 'fosile', 'tremujorin', 'penalisht', 'dobishëm', 'paparë', 'zjarrin', 'studimeve', 'shpie', 'stilin', 'ndërhyrjen'}', etc.

Natural language processing (NLP) problems are particularly well-suited to this approach since class distributions are often highly skewed and the number of classes (for example, in WSI) is not known in advance. Chinese Whispers can also manage clusters of different sizes, like some other graph clustering algorithms.

Therefore, CW can be built up into a hierarchical version as follows: Nodes of the same class are linked via hyper-nodes.

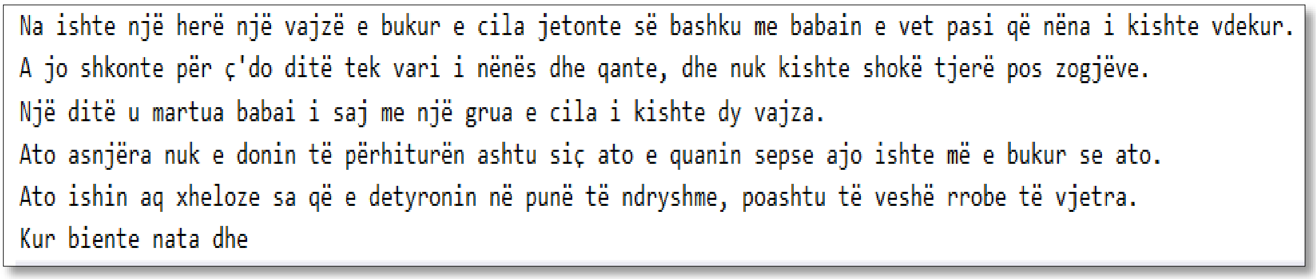
The number of inter-class edges that connect two matching nodes is taken into account when determining the value of an edge's weight between hyper-nodes. This results in flatter hierarchies. According to the 1000 and 10,000 and words list results we had there, these results are invalid.



```python
def cos_sim(v1, v2):
    sumxx, sumxy, sumyy = 0, 0, 0
    for i in range(len(v1)):
        x = v1[i]; y = v2[i]
        sumxx += x*x
        sumyy += y*y
        sumxy += x*y
    divider = math.sqrt(sumxx*sumyy)
    return sumxy/divider if divider > 0 else 0


def read_words(filename):
    tokenizer = WordPunctTokenizer()
    with open(filename, encoding='latin-1') as f:
        for line in f:
            for begin, end in tokenizer.span_tokenize(line):
                yield line[begin:end]


def is_word(w):
    return WORD_REGEX.match(w) is not None


def count(filename):
    dict = {}
    nonwords = {}
    for token in read_words(filename):
        if is_word(token):
            dict[token] = 1 if token not in dict else dict[token] + 1
        else:
            nonwords[token] = 1 if token not in nonwords else nonwords[token] + 1
    return dict, nonwords
```

Figure 32. Script of the implemantation in Python language

We did not get good results from the tests of 500 words. Although Chinese Whispers has good results for other languages, like German language [8], Polish language [9] but for our language we found anomalies in the texts that made us doubt its suitability for the Albanian language.
As a result, we began implementing the algorithm for texts we are familiar with. Since these results are dubious, we have attempted smaller texts in order to find out what the results will be for Chinese Whispers.



```
Na ishte një herë një vajzë e bukur e cila jetonte së bashku me babain e vet pasi që nëna i kishte vdekur.
A jo shkonte për ç'do ditë tek vari i nënës dhe qante, dhe nuk kishte shokë tjerë pos zogjëve.
Një ditë u martua babai i saj me një grua e cila i kishte dy vajza.
Ato asnjëra nuk e donin të përhiturën ashtu siç ato e quanin sepse ajo ishte më e bukur se ato.
Ato ishin aq xheloze sa që e detyronin në punë të ndryshme, poashtu të veshë rrobe të vjetra.
Kur biente nata dhe
```

Figure 33. The text that includes 100 words of the Albanian language

The text of figure 33 contains 100 words in Albanian. As part of the implementation process of Chinese Whispers, the text is analyzed, the cosine similarity between the terms in the text is determined, and the Chinese Whispers algorithm is used.

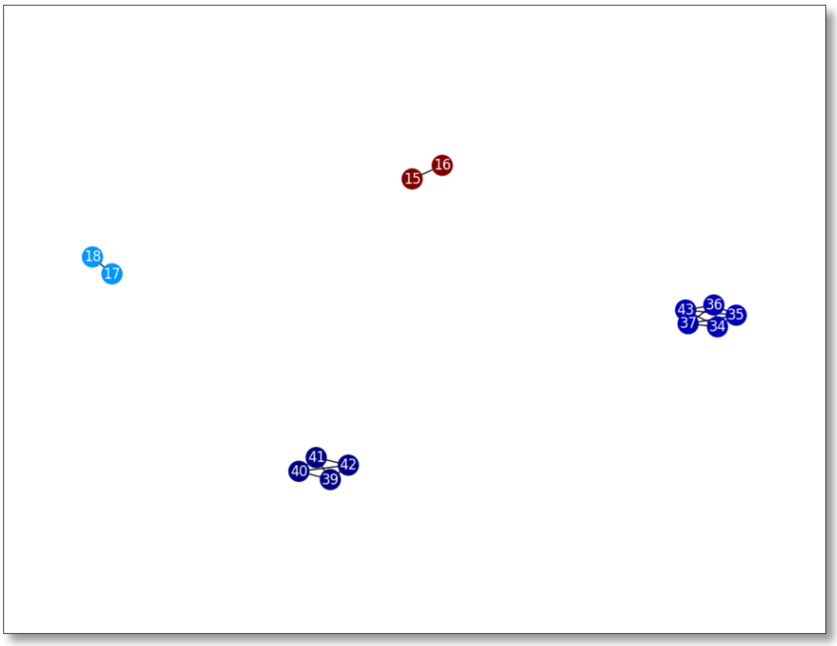


Figure 34. The results of the 100 words of Albanian language using Chinese Whispers

In addition, we can observe the outcomes of using Chinese Whispers to identify the nouns, verbs, adjectives, and other parts of speech by looking at figure 34.

Table 48. Part of speech tagging of the 100 words in the Albanian language

Verbs	Nouns	Adjective	Pronoun
ishte	herë	bukur	Na
jetonte	vajzë	shkonte	Ato
kishte	ditë		
ishin	vajza		
punë			

As a result, we've increased the number of Albanian words to 500. The results are shown in figure 35.

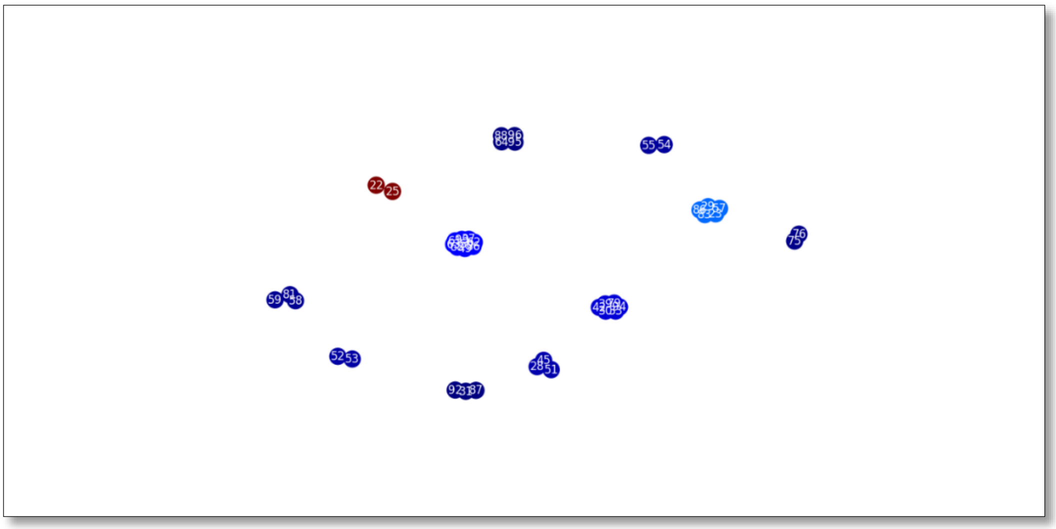


Figure 35. The results of the 500 words of Albanian language using Chinese Whispers

The list of words that are identified as shown in the table 49 below

Table 49. Part of speech tagging of the 500 words in the Albanian language

Verbs	Nouns	Adjective	Numeral	Pronoun

ishte	herë	bukur	një	Na
jetonte	vajzë	shkonte	dy	Ato
kishte	ditë	xheloze	gjasht	e cila
ishin	vajza	vjetra		
veshë	punë	martua		
binte	nata	përfunduar		
qante	xheloze	ulej		
përfundoj	punët	morrën		
mbathi	filluan			
takonte	njerka			
këpute	pëhitura			
pyeste	princi			
humbën	festë			
largu	zana			

We claim that it works well when the verbs contain identifiers or nouns, but when the text is introduced, we argue that it does not function. As a result, a study needs to be conducted on an additional variant of the Albanian language.

5.2 Effect of injecting additional language features

Since the experiments for 100 words were spell-checked, we did not have to change the results when we verified them manually. However, if there are 10,000 words, there is no need for manual intervention hence, there is no way to do it.

Due to the fact that POS tagging is easier when there are fewer words in the language, we can verify and find misspelled checks and divide them into verbs and nouns in the correct way. So, there has been no need for injecting additional language features. Furthermore, we cannot determine manually if any misspelled word is in a larger number of words.

5.3 Evaluation

We have included all of the results and comments based on the experimental data collection for the Albanian language here in this Chapter. According to the findings of the most recent phase, it would seem to be able to construct a lexicon for low-resource languages such as Albanian using unsupervised learning techniques.

In the next section, "Section 5.1," we describe an implementation of certain well-known unsupervised approaches, in which we have implemented N-grams, which are continuous sequences of words, symbols, or tokens found in a text collection. Implementing a matrix as part of the continuation of phase two of the part-of-speech tagging process for the Albanian lexicon. Section 5.1.1 presents the final matrix that was obtained by the computation of the cosine similarity from the previous matrix. This calculation was performed on the previous matrix.

All of the experiments that were conducted via the use of the unsupervised Chinese Whispers approach are described in full in this part, which can be found in Section 5.1.2 The data from the corpus were used in these several research that we conducted. 2. A Chinese Whispers game that is implemented as a randomised graph-clustering technique with a time-linear number of edges. Every node is assigned a class, and they share that information with their neighbours. In addition, the clusters for 400 words have been implemented in 187 lists, and the clusters for 10,000 words have been represented in 1,352 totals.

Because POS tagging is easier with fewer words in the vocabulary, we can find and categorize misspelled checks as verbs and nouns correctly presented in the Section 5.2. Therefore, no additional language features were required. We cannot manually determine whether a misspelled word occurs in a larger number of words.

These findings cannot be trusted since they contradict the results that we obtained from the 1000- and 10,000-word lists that we had there. The results consisting of 500 words did not provide us with satisfactory results. Although Chinese Whispers has decent results for other languages, such as the German and the Polish language, we discovered irregularities in the texts when we used it to translate into our language, which led us to question whether or not it was appropriate for the Albanian language.

6

Chapter

6. INTERPRETATION OF RESULTS

This chapter, are represented all of the experiments that we carried out, as well as the data that we gleaned from the textual analysis that we carried out as part of our research.

The fundamental purpose of this project is a dictionary for the Albanian language that has 643 thousand phrases and covers topics including law, economics, literature, science, medicine, and tourism.

The information for the tourism-domain-based corpus was taken from websites written in Albanian that are relevant to tourism. Internet resources are also used to collect data for domain-based corpora in the economics and literature fields. The utilization of these corpora paves the way for the creation of natural language processing resources that have been utilized for the Albanian language.

A collection of raw text files has been collected, and all special characters have been removed such as ‘.’, ‘-’, ‘\’, ‘&’, ‘%’, ‘(‘, ‘)’, ‘”‘, ‘+’, ‘_’ etc. The first step for text analysis is Tokenization - which is the process of breaking down a text paragraph into smaller chunks such as words.

The statistic in table shows some frequently used words out of 631,008 ones that are used more often in all selected textbooks in the Albanian language for this research

Based on the findings, we can conclude that letter 'të' appears 49992 times, followed by letter 'e' which appears 37165 times, followed by letter 'në' which appears 19577 times, etc. The most often used word is "të." In Table 50, the term is shown together with the frequency with which it is used.

Table 50. Frequencies and appearances of 200 words used more frequently

TOTAL WORD COUNT		631.008
WORD	**Appearance**	**Frequency**
të	49992	7.92%
e	37165	5.89%
në	19577	3.10%
drejtës	626	0.10%
	……..	
ndryshme	628	0.10%
juridike	620	0.10%
evropian	575	0.09%
ndërkombëtare	482	0.08%
	……..	
rëndësishme	280	0.04%
ekziston	281	0.04%
jetën	281	0.04%
Kështu	279	0.04%

In addition to this, we have also provided the findings in figure 36. Here, we are able to acquire the most common word, which is 'të,' continuing with the frequency of the rest of the words from the whole text collection of the Albanian language.

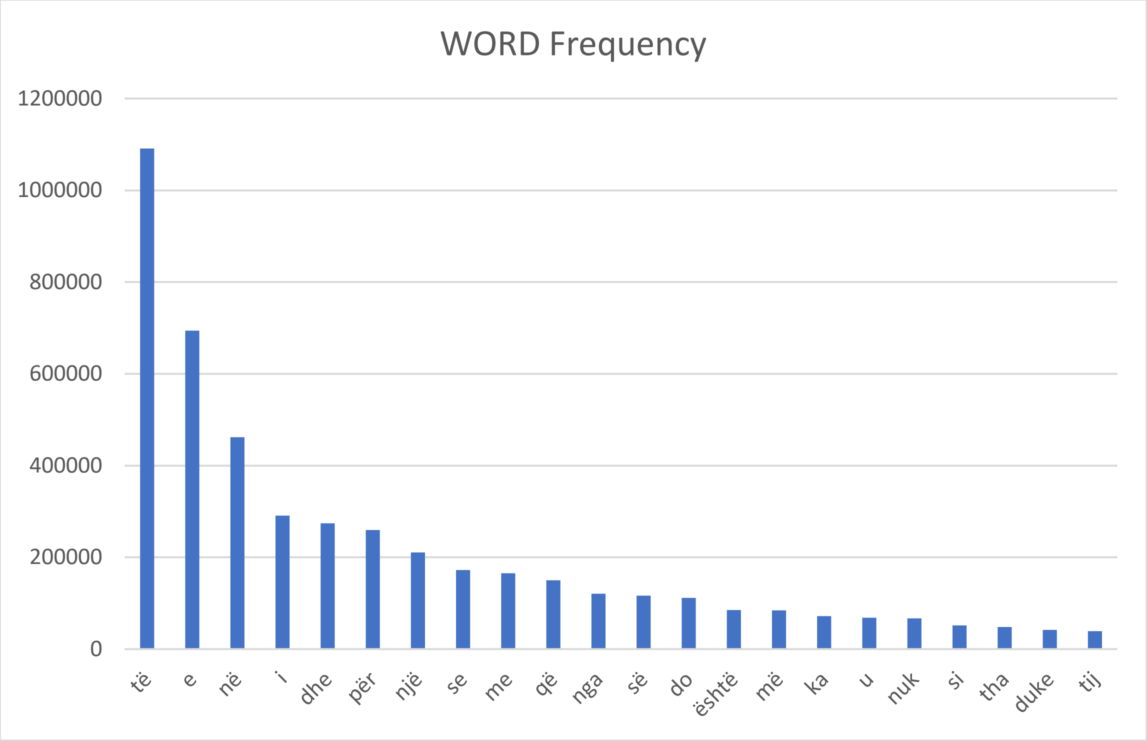


Figure 36. Graph for word frequency

Furthermore, we compared the frequency of the terms from Source 1 to that of the words in all of the source's files. The term "kolizionit" appears the most often, with a frequency of 11.0 percent, while the word "Evropian" appears the least, with a frequency of 0.9 percent. As shown in Table 51 below, "kolizionit" has the highest relative frequency.

Table 51. Words from Source 1 and all sources' frequencies

Word	Source1(AB_01)	All	Frequency
kolizionit	54	491	11.0%
afërmi	2	19	10.5%
alternative	2	23	8.7%
ndryshme	46	628	7.3%
ekonomike	9	142	6.3%
shoqërore	8	162	4.9%
ana	7	150	4.7%

afërt	2	56	3.6%
ai	20	884	2.3%

Furthermore, we have also provided the findings in figure 38. Here, we are able to acquire the most common word, which is 'të,' continuing with the frequency of the rest of the words from the whole text collection of the Albanian language.

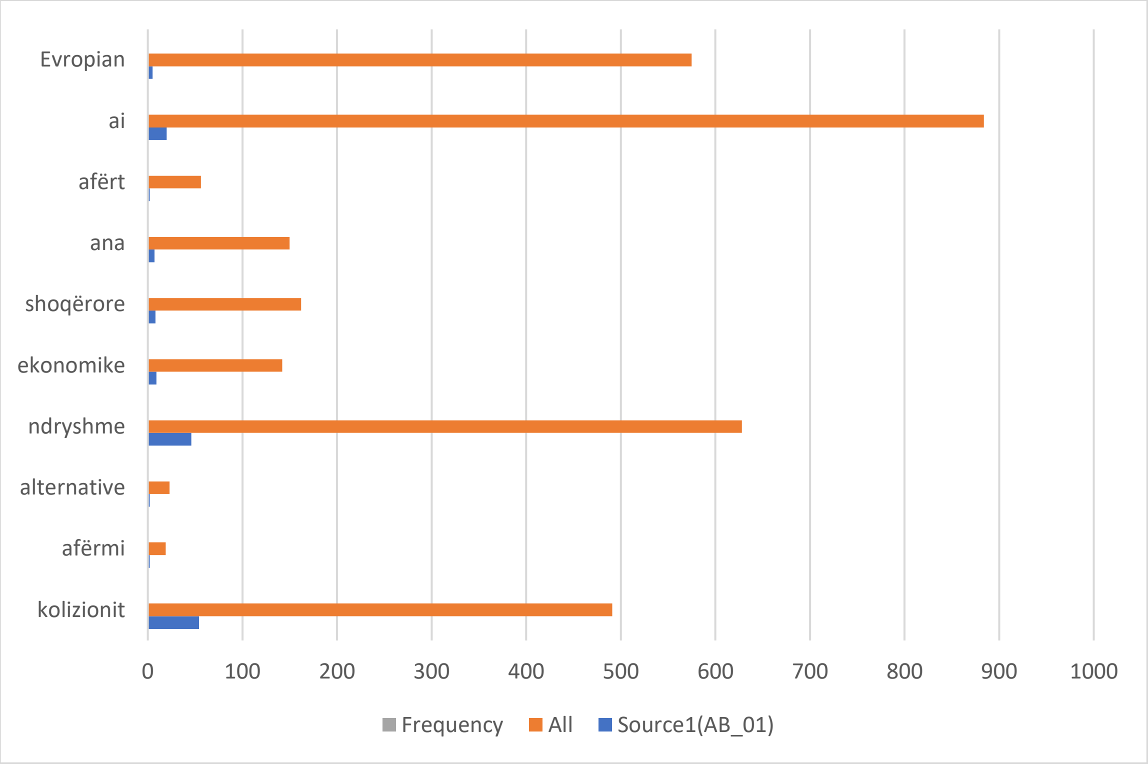


Figure 37. The frequency of words in source 1 and all the texts in the collection

In the places where hyphenated words are used, we have counted each one as two separate words. For instance, the phrase "juridiko-civile" is counted as two separate words since we have counted both the prefix "juridiko-" and the suffix "civile" as separate tokens. Table 68 presents the terms that include a hyphen, such as "juridiko-," which have been found in various sources 205 times and have a frequency of 0.568 percent. Figure 38 of a funnel representing hyphenated words as a single token.

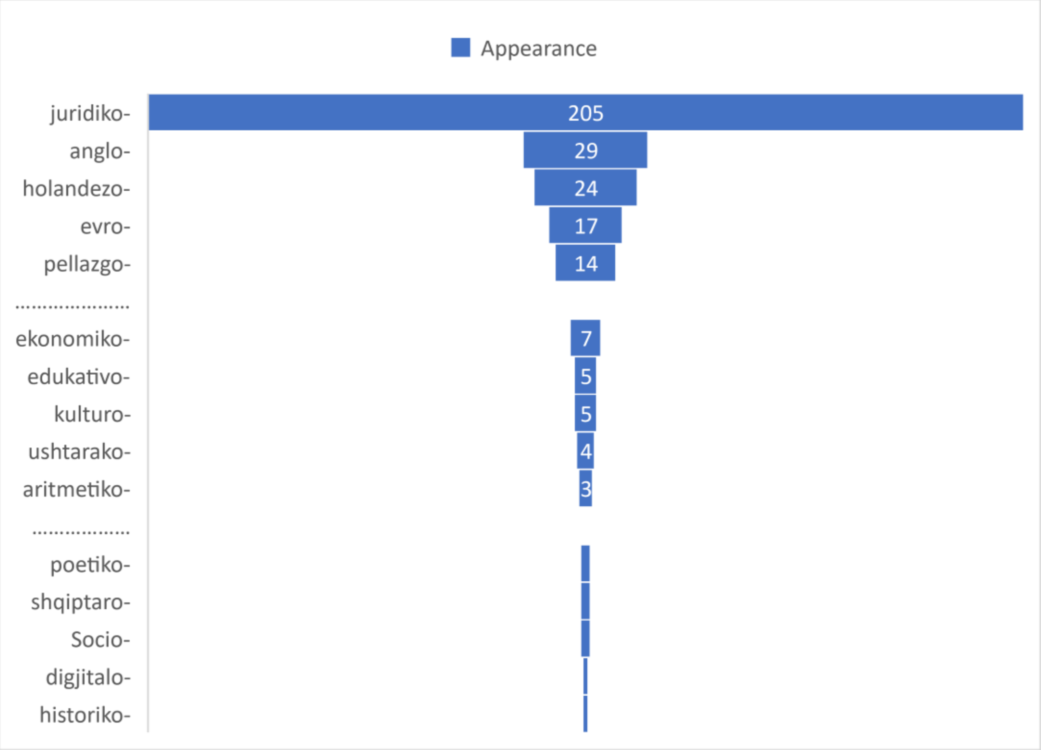


Figure 38. Funnel chart for hyphenated words as one token

During the parsing of the text, we have noticed words that use special characters such as the hyphenated words, presented in table 52.

Example: ‘juridiko-civile’:

juridiko, civile

juridiko- , civile

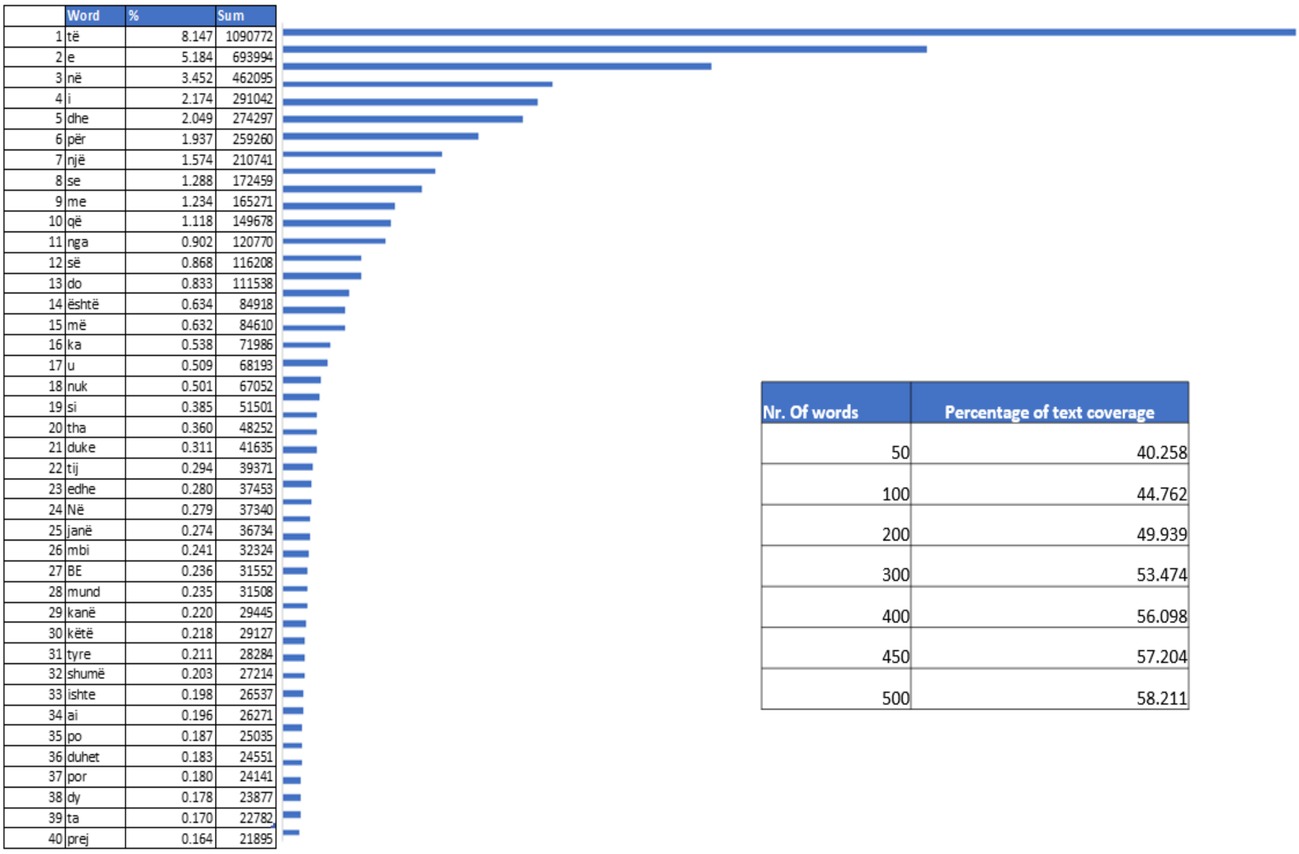
juridiko-civile

Table 52. Frequency of hyphenated words from the sources out of 643k

Word	Number of Appearances	Frequency %
juridiko-civile	113	0.0179
juridiko-private	38	0.006
juridiko-civil	24	0.0038
holandezo-flamane	20	0.0031
ushtarako-politike	3	0.0004
………		
Maqedono-Kosovare	3	0.0004
formalo-juridikisht	2	0.0003
partizano-çetnikët	1	0.0001

Calculations have been done using the terms shown in figure 53 below to determine the percentage and total. In Albanian, 50 words have a coverage percentage of 40.25 percent, 100 words have a coverage percentage of 44.76 percent, 200 words have a coverage percentage of 49.93 percent, 300 words have a coverage percentage of 53.47 percent, 400 words have a coverage percentage of 56 percent, 450 words have a coverage percentage of 57.20 percent, and 500 words have a coverage percentage of 58.211 percent.

Table 53. An analysis of different word counts and percentages of text coverage



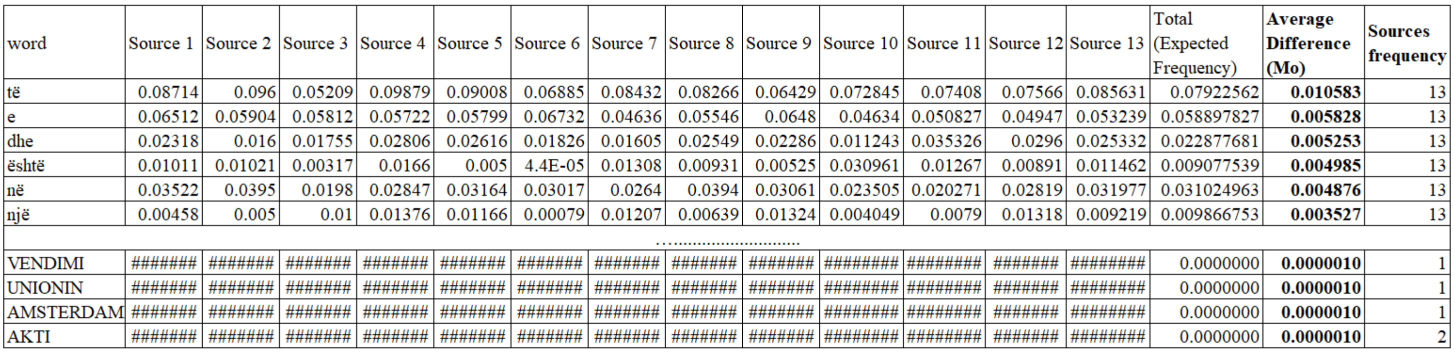
	Word	%	Sum
1	të	8.147	1090772
2	e	5.184	693994
3	në	3.452	462095
4	i	2.174	291042
5	dhe	2.049	274297
6	për	1.937	259260
7	një	1.574	210741
8	se	1.288	172459
9	me	1.234	165271
10	që	1.118	149678
11	nga	0.902	120770
12	së	0.868	116208
13	do	0.833	111538
14	është	0.634	84918
15	më	0.632	84610
16	ka	0.538	71986
17	u	0.509	68193
18	nuk	0.501	67052
19	si	0.385	51501
20	tha	0.360	48252
21	duke	0.311	41635
22	tij	0.294	39371
23	edhe	0.280	37453
24	Në	0.279	37340
25	janë	0.274	36734
26	mbi	0.241	32324
27	BE	0.236	31552
28	mund	0.235	31508
29	kanë	0.220	29445
30	këtë	0.218	29127
31	tyre	0.211	28284
32	shumë	0.203	27214
33	ishte	0.198	26537
34	ai	0.196	26271
35	po	0.187	25035
36	duhet	0.183	24551
37	por	0.180	24141
38	dy	0.178	23877
39	ta	0.170	22782
40	prej	0.164	21895

Nr. Of words	Percentage of text coverage
50	40.258
100	44.762
200	49.939
300	53.474
400	56.098
450	57.204
500	58.211

Following a simple calculation of frequency, we calculate the frequencies of 13 sources and compare them with the total frequency. In addition, we calculated the average difference by taking the words from the different sources and the total, which is the expected frequency.

In the text tokenization process, 250,152 words were generated – for which a frequency analysis and difference average analysis were conducted. (△M) frequency was calculated, which deduced four different word categorizations: (1) candidate words for the dictionary, (2) extreme cases, (3) nonextreme cases, and (4) rare words

Table 54. Average Difference for 13 sources



word	Source 1	Source 2	Source 3	Source 4	Source 5	Source 6	Source 7	Source 8	Source 9	Source 10	Source 11	Source 12	Source 13	Total (Expected Frequency)	**Average Difference (Mo)**	**Sources frequency**
të	0.08714	0.096	0.05209	0.09879	0.09008	0.06885	0.08432	0.08266	0.06429	0.072845	0.07408	0.07566	0.085631	0.07922562	**0.010583**	13
e	0.06512	0.05904	0.05812	0.05722	0.05799	0.06732	0.04636	0.05546	0.0648	0.04634	0.050827	0.04947	0.053239	0.058897827	**0.005828**	13
dhe	0.02318	0.016	0.01755	0.02806	0.02616	0.01826	0.01605	0.02549	0.02286	0.011243	0.035326	0.0296	0.025332	0.022877681	**0.005253**	13
është	0.01011	0.01021	0.00317	0.0166	0.005	4.4E-05	0.01308	0.00931	0.00525	0.030961	0.01267	0.00891	0.011462	0.009077539	**0.004985**	13
në	0.03522	0.0395	0.0198	0.02847	0.03164	0.03017	0.0264	0.0394	0.03061	0.023505	0.020271	0.02819	0.031977	0.031024963	**0.004876**	13
një	0.00458	0.005	0.01	0.01376	0.01166	0.00079	0.01207	0.00639	0.01324	0.004049	0.0079	0.01318	0.009219	0.009866753	**0.003527**	13
								…...........................								
VENDIMI	#######	#######	#######	#######	#######	#######	#######	#######	#######	########	########	#######	########	0.0000000	**0.0000010**	1
UNIONIN	#######	#######	#######	#######	#######	#######	#######	#######	#######	########	########	#######	########	0.0000000	**0.0000010**	1
AMSTERDAM	#######	#######	#######	#######	#######	#######	#######	#######	#######	########	########	#######	########	0.0000000	**0.0000010**	1
AKTI	#######	#######	#######	#######	#######	#######	#######	#######	#######	########	########	#######	########	0.0000000	**0.0000010**	2

Case 1: definite candidate words for the dictionary

When sorting the variables from the Mw in the first instance, words that provide an index greater than 1 are chosen as strong contenders for inclusion in the dictionary. The experiment produced a very encouraging result of 55.49, symbolised by the letter "t," which is utilised the most in the text sources consulted for this work.

Prepositions, conjunctions, and adverbs are frequently utilised in simple sentence construction because of the linguistic characteristics of languages, in this example, the Albanian language. Table 55 provides examples of these words. 48% or so across all sources.

Table 55. Words from the first case

WORDS	**M_W [>1.0]**
të	55.4942
gjitha	9.0986
këtë	8.7162
……..	
disa	5.6283
tjetër	3.9182

janë	3.3404
……..	
vend	2.1712
brenda	1.9539
veçantë	1.7344
……..	
kështu	1.5950
fundit	1.5563
vërtetë	1.4946

Case 2: potential candidate words for the dictionary

When sorting for the variables from the Mw in the second case, the group of words that produce an index greater than 0.1 but less than 1 are taken into consideration as potential candidate words to be added to the dictionary. These words, which are primarily nouns, verbs, and adjectives, are often used less frequently than the grammatical groups listed in Table 56. 37% total, across all sources.

Table 56. Words from the second case

WORDS	**Mw [0.1-1]**
lartë	1.2973
qenë	1.1180
pastaj	1.0625
……..	
punën	0.8838
jetën	0.8385

kombëtare	0.7443
……..	
shtëpi	0.6945
veçanta	0.6564
unike	0.6152
……..	
qytetarëve	0.3762
Evropën	0.2940
zgjedhjet	0.1899

Case 3: rare words

When sorting for the variables from the Mw in the third case, the group of words that produce an index lower than 0.1 are regarded as rare words and are not included in the dictionary as a result. Linguists should pay closer attention to these terms since, if more sources are added, the word may start to appear more frequently, increasing its index value.

The experiment in this instance produced a marginally favourable result of 0.0993, or "progress." This word (progress (Eng. : progress) - in Albanian: prparim) is typical for Case 3 because it is not an official word but an unofficial adaption from other languages. More instances of these words are provided in Table 57. 15% from all sources, roughly.

Table 57. Words from the third case

WORDS	**M_W [<0.1]**
progres	0.0993
vazhdueshëm	0.0981
kompjuterëve	0.0893

…….	
edukimi	0.0865
inteligjencave	0.0851
Shpirtërore	0.0743
…….	
zotësitë	0.0621
dyanshëm	0.0544
ushtrimeve	0.0427
…….	
Kuptosh	0.0227
balansuara	0.0167
supozimet	0.0187

Additionally, a current algorithm that aids in the processing of uncommon words ought to be used. When it comes to unusual words, we advise giving linguists the list so they can assess which of these words are foreign and which are genuine words in the Albanian language. Based on 13 sources, Table 58 has representations for all three categories.

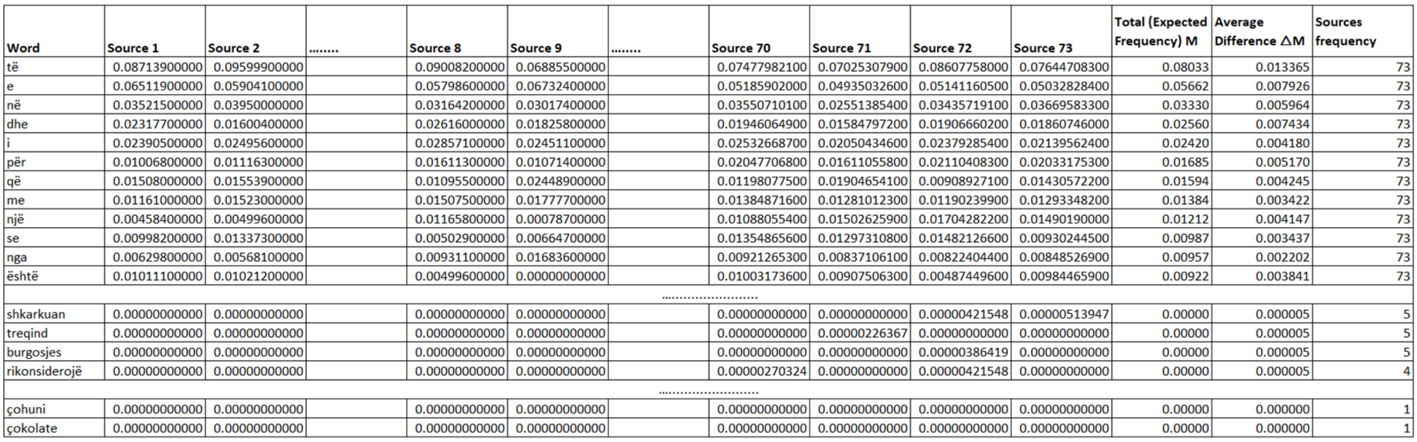
Table 58. The frequency of all sources

Case	***13 Sources***
(1)	48%
(2)	37%
(3)	15%

However, just 49,514 of the 631.008 words chosen using tokenization were accurate, while 581,494 contained extra letters or were misspelt. In this instance, the linguists would review those words and choose which should be added to the dictionary of the Albanian language.

Following that, 77MB additional of data from 60 more sources totaling 631,008 words were added to the data source. The experiment has thus far used 73 sources and 250,152 words in total. A sample of terms from various sources are shown in Table 59, together with their expected frequency and average differences.

Table 59. Average Difference for all sources



Word	Source 1	Source 2	……..	Source 8	Source 9	……..	Source 70	Source 71	Source 72	Source 73	Total (Expected Frequency) M	Average Difference △M	Sources frequency
të	0.08713900000	0.09599900000		0.09008200000	0.06885500000		0.07477982100	0.07025307900	0.08607758000	0.07644708300	0.08033	0.013365	73
e	0.06511900000	0.05904100000		0.05798600000	0.06732400000		0.05185902000	0.04935032600	0.05141160500	0.05032828400	0.05662	0.007926	73
në	0.03521500000	0.03950000000		0.03164200000	0.03017400000		0.03550710100	0.02551385400	0.03435719100	0.03669583300	0.03330	0.005964	73
dhe	0.02317700000	0.01600400000		0.02616000000	0.01825800000		0.01946064900	0.01584797200	0.01906660200	0.01860746000	0.02560	0.007434	73
i	0.02390500000	0.02495600000		0.02857100000	0.02451100000		0.02532668700	0.02050434600	0.02379285400	0.02139562400	0.02420	0.004180	73
për	0.01006800000	0.01116300000		0.01611300000	0.01071400000		0.02047706800	0.01611055800	0.02110408300	0.02033175300	0.01685	0.005170	73
që	0.01508000000	0.01553900000		0.01095500000	0.02448900000		0.01198077500	0.01904654100	0.00908927100	0.01430572200	0.01594	0.004245	73
me	0.01161000000	0.01523000000		0.01507500000	0.01777700000		0.01384871600	0.01281012300	0.01190239900	0.01293348200	0.01384	0.003422	73
një	0.00458400000	0.00499600000		0.01165800000	0.00078700000		0.01088055400	0.01502625900	0.01704282200	0.01490190000	0.01212	0.004147	73
se	0.00998200000	0.01337300000		0.00502900000	0.00664700000		0.01354865600	0.01297310800	0.01482126600	0.00930244500	0.00987	0.003437	73
nga	0.00629800000	0.00568100000		0.00931100000	0.01683600000		0.00921265300	0.00837106100	0.00822404400	0.00848526900	0.00957	0.002202	73
është	0.01011100000	0.01021200000		0.00499600000	0.00000000000		0.01003173600	0.00907506300	0.00487449600	0.00984465900	0.00922	0.003841	73
……………………													
shkarkuan	0.00000000000	0.00000000000		0.00000000000	0.00000000000		0.00000000000	0.00000000000	0.00000421548	0.00000513947	0.00000	0.000005	5
treqind	0.00000000000	0.00000000000		0.00000000000	0.00000000000		0.00000000000	0.00000226367	0.00000000000	0.00000000000	0.00000	0.000005	5
burgosjes	0.00000000000	0.00000000000		0.00000000000	0.00000000000		0.00000000000	0.00000000000	0.00000386419	0.00000000000	0.00000	0.000005	5
rikonsiderojë	0.00000000000	0.00000000000		0.00000000000	0.00000000000		0.00000270324	0.00000000000	0.00000421548	0.00000000000	0.00000	0.000005	4
……………………													
çohuni	0.00000000000	0.00000000000		0.00000000000	0.00000000000		0.00000000000	0.00000000000	0.00000000000	0.00000000000	0.00000	0.000000	1
çokolate	0.00000000000	0.00000000000		0.00000000000	0.00000000000		0.00000000000	0.00000000000	0.00000000000	0.00000000000	0.00000	0.000000	1

The word "të," which is followed by the letters "në," "që," "e," "dhe," and so forth, has the biggest anticipated frequency and average differences. Additionally, the frequency of the source was identified for each word that featured in the studies. You will receive more information on the terms that are uncommon but used in all sources as a result.

Increasing the number of sources to 73, we have conducted different experiments for all word categorizations.

Case 1: Definite candidate words for the dictionary

First, words that provide an index greater than 1 while sorting the MW's variables are taken into consideration as strong contenders for dictionary inclusion. Roughly 63% in total.

Table 60. The first case's words

WORDS	**M_W [>1.0]**
të	14891.29
gjitha	186.86
janë	567.67
……..	
shumë	407.28
disa	161.89
tjetër	145.47
vend	58.50
brenda	16.68
veçantë	8.35
ndihma	3.98
gjenetikës	1.64
klimë	1.15

Case 2: Potential candidate words for the dictionary

The group of words with an index higher than 0.1 but lower than 1 is regarded as a viable candidate for inclusion in the dictionary when sorting for the variables from the Mw. Based on 73 sources, the overall rate is estimated to be around 29%.

Table 61. Examples of words from the second case

WORDS	MW [0.1-1]
sociale	1.00
lutjen	0.99
…..	
dijetarë	0.74
firme	0.54
shkarkohet	0.42
……..	
arriturave	0.27
falëm	0.18
diskutoje	0.12
……..	
verzion	0.11
shqiptaro	0.11
biologë	0.10

Case 3: rare words

In the third case, the group of words whose index is lower than 0.1 when sorted by the variables from the MW are regarded as rare words, and as a result, they are not included in the dictionary. This is because their index is lower than the threshold for inclusion in the dictionary. 8 percent, based on 73 different sources.

Table 62. Examples of third case

WORDS	**M_W [<0.1]**
fëminore	0.099
multietnicitetit	0.086
sezonet	0.077
……..	
etnikësh	0.063
mjerimi	0.056
prushi	0.052
……..	
pararoje	0.049
titiani	0.043
esnafëve	0.039
……..	
përfaqësoja	0.027
tyrqet	0.015
investigohen	0.003

Based on a comparison of 13 sources and 73, Table 63 shows the results. In addition to that, some examples are given for some of the categories. In the research that was conducted using 13 different sources, a total of 48% (119.894) of the definite candidate words for the dictionary, 37% (92.418) of the potential candidate words, and 15% (37.469) of the rare words were found. This indicates that the contribution to the dictionary will likely be beneficial.

Table 63. Comparison percentage from different sources

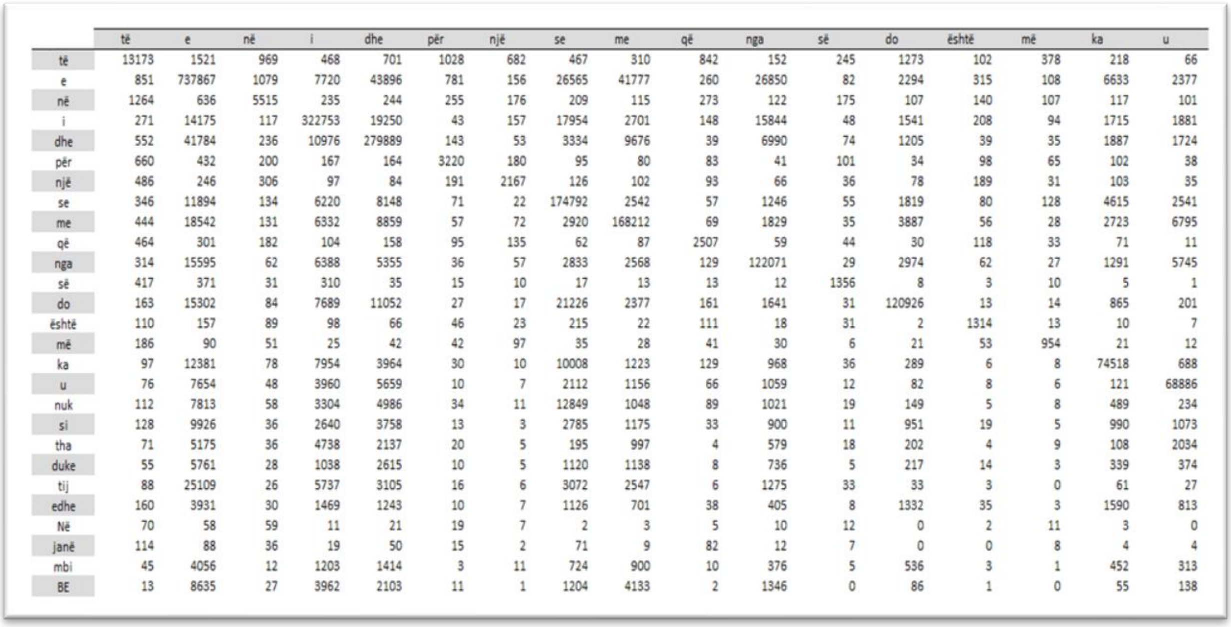
Case	***13 Sources***	***73 Sources***
(1)	48% (119.894)	63% (157.362)
(2)	37% (92.418)	29% (72.436)
(3)	15% (37.469)	8% (19.983)

When we raised the number of courses to 73, however, the findings showed that 63% (157.362) of the words fell into the first category, 29% (72.436) fell into the category of possible candidate words, and 8% (19.983) fell into the category of unusual terms.

We have created a matrix in which the columns list the 100 words from the dictionary that are found in the dictionary the most frequently, and the rows list all of the terms that are found in the dictionary.

After collecting all of the data that could fit into 80 MB, we utilized N-gram to segment it into four sentences so that it could be more easily analyzed.

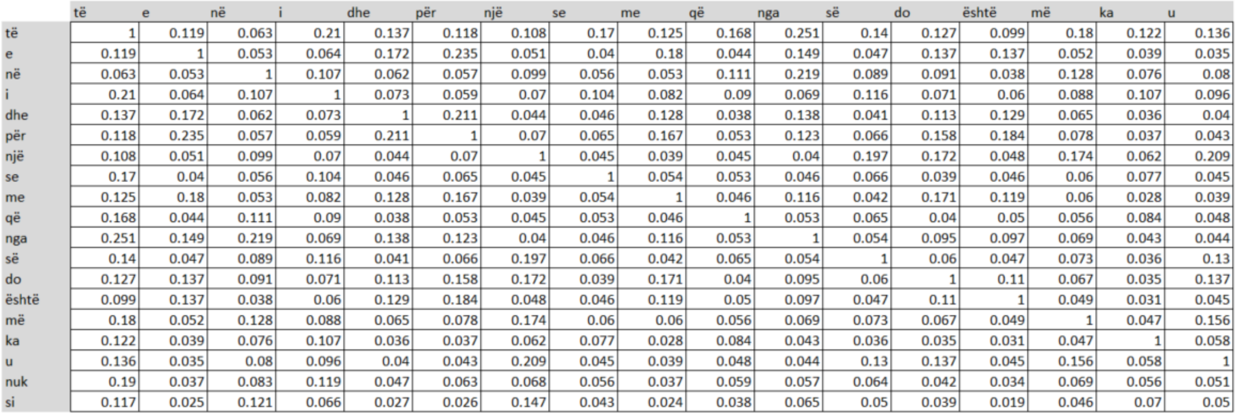
Table 64. Matrix of the Albanian language dictionary



	të	e	në	i	dhe	për	një	se	me	që	nga	së	do	është	më	ka	u
të	13173	1521	969	468	701	1028	682	467	310	842	152	245	1273	102	378	218	66
e	851	737867	1079	7720	43896	781	156	26565	41777	260	26850	82	2294	315	108	6633	2377
në	1264	636	5515	235	244	255	176	209	115	273	122	175	107	140	107	117	101
i	271	14175	117	322753	19250	43	157	17954	2701	148	15844	48	1541	208	94	1715	1881
dhe	552	41784	236	10976	279889	143	53	3334	9676	39	6990	74	1205	39	35	1887	1724
për	660	432	200	167	164	3220	180	95	80	83	41	101	34	98	65	102	38
një	486	246	306	97	84	191	2167	126	102	93	66	36	78	189	31	103	35
se	346	11894	134	6220	8148	71	22	174792	2542	57	1246	55	1819	80	128	4615	2541
me	444	18542	131	6332	8859	57	72	2920	168212	69	1829	35	3887	56	28	2723	6795
që	464	301	182	104	158	95	135	62	87	2507	59	44	30	118	33	71	11
nga	314	15595	62	6388	5355	36	57	2833	2568	129	122071	29	2974	62	27	1291	5745
së	417	371	31	310	35	15	10	17	13	13	12	1356	8	3	10	5	1
do	163	15302	84	7689	11052	27	17	21226	2377	161	1641	31	120926	13	14	865	201
është	110	157	89	98	66	46	23	215	22	111	18	31	2	1314	13	10	7
më	186	90	51	25	42	42	97	35	28	41	30	6	21	53	954	21	12
ka	97	12381	78	7954	3964	30	10	10008	1223	129	968	36	289	6	8	74518	688
u	76	7654	48	3960	5659	10	7	2112	1156	66	1059	12	82	8	6	121	68886
nuk	112	7813	58	3304	4986	34	11	12849	1048	89	1021	19	149	5	8	489	234
si	128	9926	36	2640	3758	13	3	2785	1175	33	900	11	951	19	5	990	1073
tha	71	5175	36	4738	2137	20	5	195	997	4	579	18	202	4	9	108	2034
duke	55	5761	28	1038	2615	10	5	1120	1138	8	736	5	217	14	3	339	374
tij	88	25109	26	5737	3105	16	6	3072	2547	6	1275	33	33	3	0	61	27
edhe	160	3931	30	1469	1243	10	7	1126	701	38	405	8	1332	35	3	1590	813
Në	70	58	59	11	21	19	7	2	3	5	10	12	0	2	11	3	0
janë	114	88	36	19	50	15	2	71	9	82	12	7	0	0	8	4	4
mbi	45	4056	12	1203	1414	3	11	724	900	10	376	5	536	3	1	452	313
BE	13	8635	27	3962	2103	11	1	1204	4133	2	1346	0	86	1	0	55	138

Table 64 presents a matrix representation of all of the text collections that are associated with the Albanian language. These collections include: Now that we have these discoveries in hand, we are able to move on to the next stage of the procedure, which entails determining the cosine similarity between the terms that are found in the dictionary. For illustration purposes, the letter 't' appears 13173 times in the matrix, whereas the letter 'e' may be found 737867 times.

Table 65. Matrix calculating cosine similarity



	të	e	në	i	dhe	për	një	se	me	që	nga	së	do	është	më	ka	u
të	1	0.119	0.063	0.21	0.137	0.118	0.108	0.17	0.125	0.168	0.251	0.14	0.127	0.099	0.18	0.122	0.136
e	0.119	1	0.053	0.064	0.172	0.235	0.051	0.04	0.18	0.044	0.149	0.047	0.137	0.137	0.052	0.039	0.035
në	0.063	0.053	1	0.107	0.062	0.057	0.099	0.056	0.053	0.111	0.219	0.089	0.091	0.038	0.128	0.076	0.08
i	0.21	0.064	0.107	1	0.073	0.059	0.07	0.104	0.082	0.09	0.069	0.116	0.071	0.06	0.088	0.107	0.096
dhe	0.137	0.172	0.062	0.073	1	0.211	0.044	0.046	0.128	0.038	0.138	0.041	0.113	0.129	0.065	0.036	0.04
për	0.118	0.235	0.057	0.059	0.211	1	0.07	0.065	0.167	0.053	0.123	0.066	0.158	0.184	0.078	0.037	0.043
një	0.108	0.051	0.099	0.07	0.044	0.07	1	0.045	0.039	0.045	0.04	0.197	0.172	0.048	0.174	0.062	0.209
se	0.17	0.04	0.056	0.104	0.046	0.065	0.045	1	0.054	0.053	0.046	0.066	0.039	0.046	0.06	0.077	0.045
me	0.125	0.18	0.053	0.082	0.128	0.167	0.039	0.054	1	0.046	0.116	0.042	0.171	0.119	0.06	0.028	0.039
që	0.168	0.044	0.111	0.09	0.038	0.053	0.045	0.053	0.046	1	0.053	0.065	0.04	0.05	0.056	0.084	0.048
nga	0.251	0.149	0.219	0.069	0.138	0.123	0.04	0.046	0.116	0.053	1	0.054	0.095	0.097	0.069	0.043	0.044
së	0.14	0.047	0.089	0.116	0.041	0.066	0.197	0.066	0.042	0.065	0.054	1	0.06	0.047	0.073	0.036	0.13
do	0.127	0.137	0.091	0.071	0.113	0.158	0.172	0.039	0.171	0.04	0.095	0.06	1	0.11	0.067	0.035	0.137
është	0.099	0.137	0.038	0.06	0.129	0.184	0.048	0.046	0.119	0.05	0.097	0.047	0.11	1	0.049	0.031	0.045
më	0.18	0.052	0.128	0.088	0.065	0.078	0.174	0.06	0.06	0.056	0.069	0.073	0.067	0.049	1	0.047	0.156
ka	0.122	0.039	0.076	0.107	0.036	0.037	0.062	0.077	0.028	0.084	0.043	0.036	0.035	0.031	0.047	1	0.058
u	0.136	0.035	0.08	0.096	0.04	0.043	0.209	0.045	0.039	0.048	0.044	0.13	0.137	0.045	0.156	0.058	1
nuk	0.19	0.037	0.083	0.119	0.047	0.063	0.068	0.056	0.037	0.059	0.057	0.064	0.042	0.034	0.069	0.056	0.051
si	0.117	0.025	0.121	0.066	0.027	0.026	0.147	0.043	0.024	0.038	0.065	0.05	0.039	0.019	0.046	0.07	0.05

Table 65, which can be obtained by glancing at the output, displays the cosine similarity of the words; the value for the word "t" is 1, the similarity for the word "e" is 0.119, the similarity for the word "n" is 0.063, and so on. The output also displays the value for the word "n", which is 0.063. If the cosine similarity is relatively close to one, as was discussed in the theory, this indicates that the two vectors are quite comparable to one another.

The text of figure 39 contains 100 words in Albanian. As part of the implementation process of Chinese Whispers, the text is analyzed, the cosine similarity between the terms in the text is determined, and the Chinese Whispers algorithm is used.

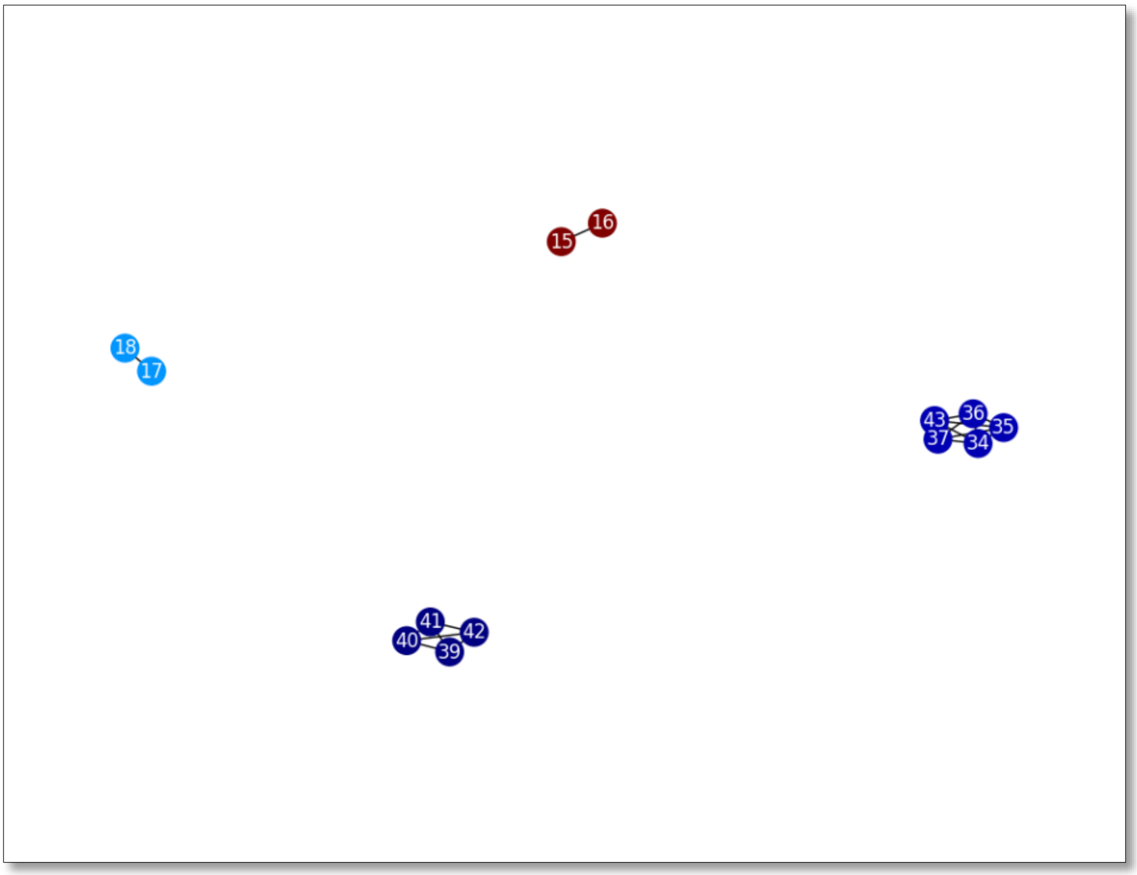


Figure 39. The results of the 100 words of Albanian language using Chinese Whispers

In addition, we can observe the outcomes of using Chinese Whispers to identify the nouns, verbs, adjectives, and other parts of speech by looking at Table 66.

Table 66. Part of speech tagging of the 100 words in the Albanian language

Verbs	Nouns	Adjective	Pronoun
ishte	herë	bukur	Na
jetonte	vajzë	shkonte	Ato
kishte	ditë		
ishin	vajza		
punë			

As a result, we've increased the number of Albanian words to 500. The results are shown in figure 40.

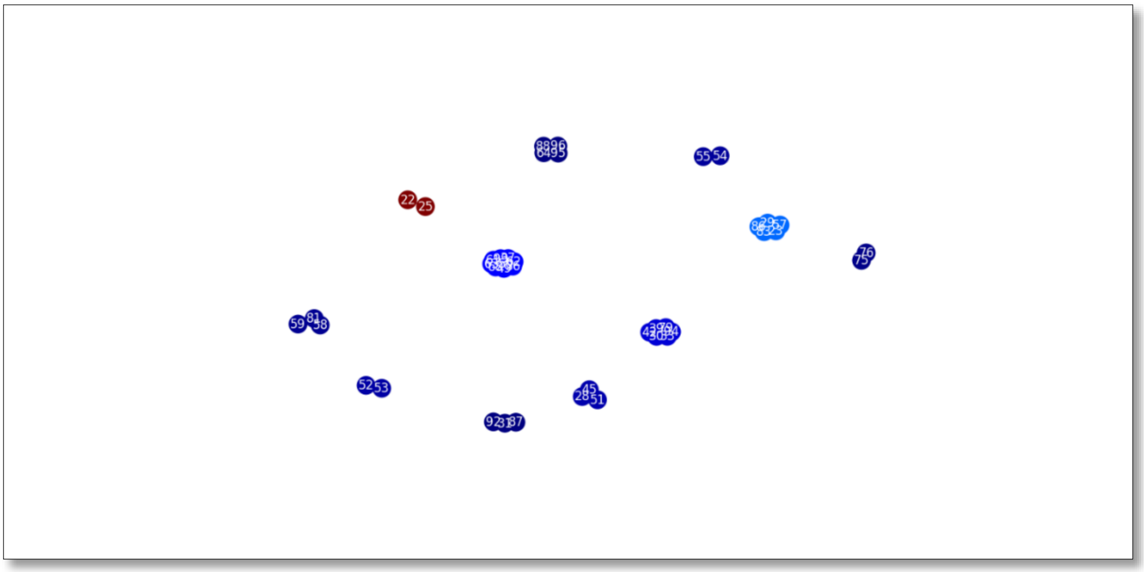


Figure 40. The results of the 500 words of Albanian language using Chinese Whispers

The table contains a list of the words that are identified, which may be found below in table 67.

Table 67. Part of speech tagging of the 500 words in the Albanian language

Verbs	Nouns	Adjective	Numeral	Pronoun
ishte	herë	bukur	një	Na
jetonte	vajzë	shkonte	dy	Ato
kishte	ditë	xheloze	gjasht	e cila
ishin	vajza	vjetra		
veshë	punë	martua		
binte	nata	përfunduar		
qante	xheloze	ulej		
përfundoj	punët	morrën		
mbathi	filluan			
takonte	njerka			
këpute	pëhitura			
pyeste	princi			
humbën	festë			
largu	zana			

When the verbs contain identifiers or nouns, it functions well, according to our assertions; however, when the text is brought into play, it does not function properly. As a direct consequence of this, it is necessary to carry out research on an extra variety of the Albanian language.

7

Chapter

7. CONCLUSION AND FUTURE WORK

In general, there is a lack of academic study and effort related to NLP for the Albanian language, which has few word etymologies in common with other higher resource languages and belongs to the linguistic group of low resources. Through this Ph.D. we have targeted the five research questions:

1. Is it possible to generate spell-check dictionaries from the raw text?
2. Is it possible to detect misspelled words based on usage differences?
3. What algorithms can be used to find rarely used words?
4. What results can be obtained by applying unsupervised POS tagging to a large text collection in Albanian?
5. How increasing the text collection affects the accuracy?

We came to the conclusion that it is feasible to construct a spell-check dictionary using the row text that is supplied in chapter 4, which relates to Research Question 1. In the collection of Albanian language texts, 73 sources were used, in which the spell-check dictionary was generated using NLP phases, including tokenization of sentences, chunking them into words, removing special characters, and finding frequency and usage from the total number of words. A dictionary with 250,000 unique words is compiled from the row text collection for Albanian.

As part of our investigation into Research Question No. 2, we came to the conclusion that, due to differences in the way words are used, it is possible to find misspelt terms in the text collection for the Albanian language. The term that only appears once in one source is examined in Section 4.1.3. Only about 20.001 words, or the third group of words with an 8% frequency, are found. In addition, the terms supplied appear around 3.638 times in Case 2, and 15.128 times in Case 3. 117.371 words appear twice in Case 2.

In order to answer Research question #3, we conducted an analysis on many different algorithms. Some of these algorithms are described in chapter 3, such as Chinese Whispers, K-Means, and K-The frequency of each phrase was given in section 4.1.4, and an analysis of the data was done using the comparison algorithm. Regarding each category, this procedure was used. There are currently 220,604 words in the dictionary, an 11% reduction from the previous total (89 percent).

For Research Question #4, we assessed various from the given data, we can conclude that not all clusters were as expected due to the usage of the unsupervised approach, but some of the clusters were appropriately displaying terms with the same morphological analysis. The section 5.1.2, provides a detailed description of every experiment that was carried out using the unsupervised Chinese Whispers method. We did a number of studies using the data from the corpus. A Chinese Whispers that uses a time-linear number of edges and a randomised graph clustering algorithm. Each node has a class assigned to it, and they all communicate this information with one another. Additionally, 187 lists have used the clusters for 400 words, and 1,352 totals have used the clusters for 10,000 words.

As a direct consequence of the fifth research question, we have expanded the number of sources we use to sixty. These new sources have been added to the data source, which now has a total capacity of seventy-seven megabytes and a total of 631,008 words. Therefore, as of right now, the experiment has included a total of 73 sources and 250,152 words, all of which are shown in Table 23. As a consequence of an increase in text collections in the Albanian language, the accuracy of dictionaries has increased.

Additionally, through this Ph.D. thesis, we have shown how low-resource languages may catch up to the majority language group in terms of value.
Based on these hypotheses, a framework has been established that can be applied to other languages and low-resource environments as well.

Null: Large text collection can be effectively used to improve language feature extraction for low resources languages

H1: Word usage differences can be used to detect misspelled words in automatically built dictionaries
H2: Structural characteristics of text can contribute to dictionary completion
H3:H1 and H2 results help improve automated POS tagging in low resources languages.

As a result of building a dictionary for the Albanian language, we can conclude that the Null hypothesis has been confirmed. The use of text collections from different fields has improved this low-resource language's features.

As a result of different usage between sources, we were able to detect the misspelled words in our dictionary, and based on the usage we were able to determine which words were misspelled in the text, so we can conclude that Hypothesis H1 has been completed.

Furthermore, based on hypothesis H2 we can conclude that structural characteristics have been very helpful during our process of building the dictionary for the Albanian language. For the last hypothesis H3, we can conclude that using few sources POS tagging has been improved for low-resource languages, the Albanian language database was limited; nonetheless, if there was a richer dataset, the results would undoubtedly be different.

In the first chapter, we went over the background information and the difficulty with the study, as well as the premise, the research questions, and the technique. As was shown, the Albanian language is a good example of a language that has limited access to resources. Furthermore, almost every language has a dictionary; here, we explore the relevance of resolving this issue as well as the benefits that come with doing so.

The study of linguistics is going to benefit enormously from the inclusion of this new information. In addition, we emphasized that the outcomes of this research would be used to develop this low-resource language. This is especially important considering that the Albanian language does not have a considerable amount of digital vocabulary.

In Chapter 3, we provided the findings from our qualitative study, which included a literature evaluation of over 205 publications. This study was given after we had conducted our qualitative research. The domains of Natural Language Processing, Machine Learning, and Language Identification were represented by the papers.

It was provided that an overview of unsupervised learning approaches as well as N-gram was given. Additionally, an introduction to Natural Language Processing was provided in Chapter 2, which may be found here. As a direct consequence of this, we were in a position to derive theoretical insights, which aided our process of selecting the implementation model.

In Chapter 4, we presented all experimental and empirical results for all the cases of building the vocabulary. The results from the categorization of words from 73 sources were promising in terms of accuracy. On other hand, it increased in execution time and challenges when the size of the dataset increased.

In addition, as a component of chapter 4, illustrates reinforcement learning using data that has been explicitly annotated. Random selection was used to choose, from within each group, a total of 150 properly spelt words. In the first scenario, there is not a single misspelt word; in the second scenario, there are around 24 percent of incorrect words; and in the third scenario, 76 percent of the words have incorrect spellings.

In Chapter 5 we presented all the findings and discussions based on the experimental data collection for the Albanian language. Based on the results of the last phase, it appears possible to build a lexicon for low-resource languages like Albanian using unsupervised learning approaches.

In addition, as part of Chapter 5, there is shown the use of well-known unsupervised approaches, in which we have implemented N-grams, which are unbroken sequences of words, symbols, or tokens that are found in a text collection. Implementing a matrix as part of the continuation of phase two of the part-of-speech tagging process for the Albanian lexicon. The ultimate matrix that was produced after the computation of the cosine similarity from the prior matrix, which can be found in section 5.1.1.

The findings and outcomes of our research are interpreted in Chapter 6, which is the last chapter of our doctoral thesis. All of the tests that were carried out using various text collections and applying them to a variety of scenarios contributed to the development of a vocabulary for the Albanian language. The vocabulary consists of two hundred and fifty thousand words, making it the most extensive corpus of the Albanian language to date. The official documents of Albania would have been translated accurately into the Albanian language if the country had been a member of the European Union. Digitization of the Albanian language, then we would have more collection of texts. As a result of digitizing libraries, dictionaries could be created that would be more accurate and precise as there would be a large collection of text.

Because there were few sources for low-resource languages, the Albanian language database was limited; nonetheless, if there was a richer dataset, the results would undoubtedly be different. Due to the fact that the model has been developed, it will be available to all other interested parties to continue research in this area.

Other candidates that are affiliated with the SEEU database will have access to our dataset, allowing them to contribute to the Albanian language.

In future work:

- Having a large collection of Albanian texts
- The implementation of other algorithms

An analysis of how different algorithms perform differently in different languages.

PUBLICATIONS AND PRESENTATIONS

Conference Proceedings:

Diellza Nagavci Mati, Mentor Hamiti, Jaumin Ajdari, Besnik Selimi, and Bujar Raufi, “A Systematic Mapping Study of Language Features Identification from Large Text Collection”, Proceedings of the MECO'2019, 8th Mediterranean Conference on Embedded Computing (MECO 2019), IEEE Conference Publications [https://ieeexplore.ieee.org/document/8760042], DOI: 10.1109/MECO.2019. 8760042, Pages: 242 – 246, 10-14 June 2019, Budva, Montenegro

Diellza Nagavci Mati, Mentor Hamiti, Jaumin Ajdari, Bujar Raufi & Besnik Selimi, “Review of Natural Language Processing tasks in Albanian language”, 3rd International Scientific Conference on Business and Economics, "From Transition to Development: Emerging Challenges and Perspectives", Conference Proceedings [https://conf.seeu.edu.mk/archive/], SEEU, ISBN: 978-608-248-031-2, Pages: 317-323,
13-15 June 2019, Skopje, North Macedonia

Diellza Nagavci Mati, Mentor Hamiti, Besnik Selimi & Jaumin Ajdari, "Building Spell-Check Dictionary for Low-resource Language by Comparing Word Usage", Proceedings of the mipro 2021, 44th International Convention on Information, Communication and Electronic Technology (MIPRO), IEEE Conference Publications [https://ieeexplore.ieee.org/document/9597183], ISSN: 2623-8764, pp. 225-262,
Sept. 27 – Oct. 01, 2021, Opatija, Croatia

International Journals:

Diellza Nagavci Mati, Mentor Hamiti, Arsim Susuri, Besnik Selimi & Jaumin Ajdari, “Unsupervised Learning Challenges of Building Dictionaries for Low Resource Languages”, Annals of Emerging Technologies in Computing (**AETiC**), Vol. 5, No. 3, pp. 52-58, **2021**, ISSN 2516-0281. [https://aetic.theiaer.org/archive/v5/v5n3/p5.html]

Diellza Nagavci Mati, Mentor Hamiti, Elissa Mollakuqe, “Morphological Tagging and Lemmatization in the Albanian Language”, Seeu Review Vol. 16, Issue 2, pp. 3-16, 29, Dec,**2021**, DOI: 10.2478/seeur-2021-0015 [https://doi.org/10.2478/seeur-2021-0015]

ACKNOWLEDGEMENT

A journey of a thousand miles begins with a single step” is a maxim that many acknowledge but a few seek to truly understand. Embarking on this journey has weathered me through a spectrum of life colors: I have enjoyed exuberance and vigor when the tide of science aligned with my path, and I have endured despair and stress when time and energy did not translate immediately into success. Though it has covered but a fraction of my life, the countless steps to this ‘thousand-mile’ journey have made me wiser, not only as a researcher but rather as a person.

The beacon that has shined my path forward in this endeavor has undoubtedly been my supervisor, mentor, and guide, Prof. Dr. Mentor Hamiti, who spent countless hours guiding, advising, and encouraging me to take those ‘steps’ one after the other, and understand that they will ultimately lead to this point. Also, I like to express immense gratitude to Prof. Besnik Selimi, and Prof. Jamuin Ajdari for providing invaluable insight, guidance, and support on the thesis.

I dedicate this work to my parents, who have instilled in me the work ethic, relentless attitude, and courage to boldly face the unknown, who ultimately made me the person I am today. I am grateful to my husband Baton who shared with me the moments of joy, and comforted me on my strides of difficulty; for his continuous support, motivation, and empathy throughout this journey.

Finally, my shining star, Gea, I hope that you will be motivated by me as much as you have been my motivation since day one, and one day, you will be as proud of me as I am of my parents.

I still do not claim to understand the full depth of Lao Tzu's wisdom, but I have come to acknowledge that a journey is the sum of experiences, knowledge, and lessons one learns along the way.

PROOFREADING DECLARATION

I, Zhaklina Lipoveci

Hereby declare that I have proofread PhD thesis with title

“Language Feature Identification from Large Text Collections”

To the best of my ability, and such, it meets the criteria for being defended and published.

Respectfully,

Prof. ______

Gerichtsdolmetsch - Interprete Giudiziale - Interprete Judiciaire * Court Interpreter -
Interpret Gjyqësor - Sudski Tumač *
Prof.
Zhaklina Lipoveci
* GJAKOVË - DJAKOVICA *

BIBLIOGRAPHY

[1] A. Jain, G. Kulkarni and V. Shah, “Natural Language Processing,” *International Journal of Computer Sciences and Engineering,* vol. 6, no. 1, pp. 161-167, 2018.

[2] A. K. K. S. S. Diksha Khurana, Natural Language Processing: State of The Art, Current Trends and Challenges, India: IEEE, 2017.

[3] Y. Zhao and X. Zhou, “K-means Clustering Algorithm and Its Improvement Research,” *2nd International Workshop on Electronic communication and Artificial Intelligence (IWECAI 2021) Nanjing, China,* pp. 62-68, 12-14 March 2021.

[4] R. P. Vaishali and M. Rupa G. , “Impact of Outlier Removal and Normalization Approach,” *IJCSI International Journal of Computer Science,* vol. 2, no. 5, pp. 331-336, 2011.

[5] S. Bird, E. Klein and E. Loper, Natural Language Processing with Python, USA: O'Reilly Media, 2009.

[6] M. Hamiti, V. Shehu and A. Dika, “Automatic Syllable Identification for the Albanian Language,” *SEEU Review,* Vols. Volume 5, No. 2, pp. 173-192, 2009.

[7] N. Kote, M. Biba and E. Trandafili, “A Thorough Experimental Evaluation of Algorithms for Opinion Mining in Albanian,” in *Proceedings of the International Conference on Emerging Internetworking, Data & Web Technologies*, Thailand,, 2018.

[8] C. Biemann, “Chinese Whispers - an Efficient Graph Clustering Algorithm and its Application to Natural Language Processing Problems,” in *Association for Computational Linguistics*, New York City, 2006.

[9] Ł. Degórski, “Fine-tuning Chinese Whispers algorithm for a Slavonic language POS tagging task and its evaluation,” in *Computer and Science* , Poland, 2013.

[10] C. Biemann, Unsupervised Part-of-Speech Tagging in the Large, Germany: Res on Lang and Comput, 2016.

[11] V. H. Walter Daelemans, “Evaluation of Machine Learning Methods for Natural Language Processing Tasks,” *IEEE,* 2016.

[12] N. J. a. I. M. Vaishali Gupta, “Advanced Machine Learning Techniques in Natural Language Processing for Indian Languages,” *Springer Link,* 2019.

[13] S. P. Dipanjan Das, “Unsupervised Part-of-Speech Tagging with Bilingual Graph-Based Projections,” *Proceedings of the 49th Annual Meeting of the Association for Computational*

Linguistics, , p. pages 600–609, 2011.

[14] M. B. a. E. T. Nelda Kote, “An Experimental Evaluation of Algorithms for Opinion Mining in Multi-domain Corpus in Albanian,” *© Springer Nature Switzerland AG ,* 2018.

[15] G. P. Petasis?, “Machine Learning in Natural Language Processing,” *IEEE,* 2017.

[16] P. S. S. Langley, “Learning context-free grammars with a simplicity bias,” *Proceedings of the 11th European Conference on Machine Learning. ECML,* 2016.

[17] M. C. Karl Stratos, “Unsupervised Part-Of-Speech Tagging with Anchor Hidden Markov Models,” *Transactions of the Association for Computational Linguistics, vol. 4, Action Editor: Hinrich Schutze.,* p. pp. 245–257, 2016.

[18] K. L. Odile Piton, “Morphological study of Albanian words, and processing with NooJ,” *Barcelone, Spain. Cambridge Scholar Publishing, ,* pp. pp.189-205,, 2015.

[19] A. C. Shervin Malmasi, “Measuring Feature Diversity in Native Language Identification,” *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications, ,* p. pages 49–55, 2015.

[20] R. Collobert, “A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning,” *IEEE,* 2016.

[21] D. Zhai, X. Liu, X. Ji, D. Zhao, S. Satoh and W. Gao, “Supervised Distributed Hashing for Large-Scale Multimedia Retrieval,” *IEEE,* 2018.

[22] Y. D. X. L. T. N. Y. L. Zexian Zeng, “Natural Language Processing for EHR-Based Computational Phenotyping,” *IEEE,* 2019.

[23] M. S. A. F. V. a. X.-w. C. Itauma Itauma∗1, “Unsupervised Learning and Image Classification in High Performance Computing Cluster,” *IEEE 14th International Conference on Machine Learning and Applications,* 2015.

[24] Y. Park and S. Kang, “Natural Language Generation Using Dependency Tree Decoding for Spoken Dialog Systems,” *IEEE,* 2019.

[25] W. Xiang, H. Zhang, R. Cui, X. Chu, K. Li and W. Zhou, “Pavo: A RNN-Based Learned Inverted Index, Supervised or Unsupervised?,” *IEEE,* 2019.

[26] S. Gharge and M. Chavan, “An integrated approach for malicious tweets detection using NLP,” *IEEE,* 2017.

[27] T. Ohanian, “How Artificial Intelligence and Machine Learning May Eventually Change Content

Creation Methodologies," *IEEE,* 2019.

[28] D. Goldberg, " Genetic Algorithms in Search, Optimization,," *IEEE,* 2015.

[29] B. W. Erik Cambria, "Jumping NLP Curves: A Review of Natural," *Digital Object Identifier 10.1109/MCI.,* 2015.

[30] S. R. Gunn, "Support Vector Machines for Classification and Regression," 2015.

[31] T. Hutchinson, "Protecting Privacy in the Archives: Supervised Machine Learning and Born-Digital Records," *IEEE,* 2018.

[32] R. S. Mahajan and M. A. Zaveri, "Machine learning based paraphrase identification system using lexical syntactic features," *IEEE,* 2017.

[33] Y. Song, Y. Wang and J. Viventi, "Unsupervised Learning of Spike Patterns for Seizure Detection and Wavefront Estimation of High Resolution Micro Electrocorticographic ($\mu $ ECoG) Data," *IEEE,* 2017.

[34] A. Shrestha, L. Kaati and K. Cohen, "A Machine Learning Approach towards Detecting Extreme Adopters in Digital Communities," *IEEE,* 2017.

[35] T. J. Sefara, M. J. Manamela and P. T. Malatji, "Text-based language identification for some of the under-resourced languages of South Africa," *IEEE,* 2016.

[36] M. P. SkÃ«nduli and M. Biba, "A Named Entity Recognition approach for Albanian," *IEEE,* 2016.

[37] A. E. Youssef, A. S. Ibrahim and A. L. Abbott, "Automated gender identification for Arabic and English handwriting," *IEEE,* 2015.

[38] J. Cai, J. Li, W. Li and J. Wang, "Deeplearning Model Used in Text Classification," *IEEE,* 2018.

[39] Z. Xuanyu, L. Xiao, P. Xin, L. XiangSi and Xiangyi, "Number Type Identification of Chinese Personal Noun Phrases via Piece-wise Linear Support Vector Machine," *IEEE,* 2018.

[40] S. B. Bodapati, S. Ramaswamy and G. Narayanan, "A Machine Learning Approach to Detecting Start Reading Location of eBooks," *IEEE,* 2018.

[41] S. B. Bodapati, S. Ramaswamy and G. Narayanan, "A Machine Learning Approach to Detecting Start Reading Location of eBooks," *IEEE,* 2018.

[42] J. S. Anjana and S. S. Poorna, "Language Identification From Speech Features Using SVM and LDA," *IEEE,* 2018.

[43] P. Suri and N. R. Roy, "Comparison between LDA & NMF for event-detection from large text

stream data," *IEEE,* 2017.

[44] H. Alam and A. Kumar, "Multi-lingual author identification and linguistic feature extraction â€" A machine learning approach," *IEEE,* 2015.

[45] M. A. Shahin, B. Ahmed and K. J. Ballard, "Automatic classification of unequal lexical stress patterns using machine learning algorithms," *IEEE,* 2014.

[46] W. S. Lee, N. C. Kim and I. H. Jang, "Texture feature-based language identification using wavelet-domain BDIP, BVLC, and NRMA features," *IEEE,* 2015.

[47] D. Kucuk and M. T. Yondem, "Automatic identification of pronominal Anaphora in Turkish texts," *IEEE,* 2015.

[48] W. Liu and L. Wang, "Unsupervised ensemble learning for Vietnamese multisyllabic word extraction," *IEEE,* 2016.

[49] M. Atzeni and M. Atzori, "Translating Natural Language to Code: An Unsupervised Ontology-Based Approach," *IEEE,* 2018.

[50] Y. Zaki, H. Hajjar, M. Hajjar and G. Bernard, "Towards the development of a statistical parser of Arabic language," *IEEE,* 2017.

[51] Q. Zhang and J. H. L. Hansen, "Language/Dialect Recognition Based on Unsupervised Deep Learning," *IEEE,* 2018.

[52] T. T. Urmi, J. J. Jammy and S. Ismail, "A corpus based unsupervised Bangla word stemming using N-gram language model," *IEEE,* 2016.

[53] S. a. L. P. IEEE/ACM Transactions on Audio, "Supervised Detection and Unsupervised Discovery of Pronunciation Error Patterns for Computer-Assisted Language Learning," *IEEE,* 2015.

[54] A. Celikyilmaz, R. Sarikaya, M. Jeong and A. Deoras, "An Empirical Investigation of Word Class-Based Features for Natural Language Understanding," *IEEE,* 2016.

[55] O. Güngör and E. Yıldız, "Linguistic features in Turkish word representations," *IEEE,* 2017.

[56] D. Wanvarie, S. Ek-atchariya and T. Kaewwipat, "Unsupervised construction of a word list on tourism from Wikipedia," *IEEE,* 2015.

[57] A. Kumar, L. Padró and A. Oliver, "Unsupervised learning of agglutinated morphology using nested Pitman-Yor process based morpheme induction algorithm," *IEEE,* 2015.

[58] H. Walia, A. Rana and V. Kansal, "A Naïve Bayes Approach for working on Gurmukhi Word

Sense Disambiguation," *IEEE,* 2017.

[59] M. J. Carbajal, A. Dawud, R. Thiollière and E. Dupoux, "The "language filter" hypothesis: A feasibility study of language separation in infancy using unsupervised clustering of I-vectors," *IEEE,* 2016.

[60] L. Nandanwar, "Graph connectivity for unsupervised Word Sense Disambiguation for HINDI language," *IEEE,* 2015.

[61] W. Lopez, J. Merlino and P. Rodriguez-Bocca, "Vector representation of internet domain names using a word embedding technique," *IEEE,* 2017.

[62] C. Ni, C.-C. Leung, L. Wang, N. F. Chen and B. Ma, "Unsupervised data selection and word-morph mixed language model for tamil low-resource keyword search," *IEEE,* 2015.

[63] D. H. Sasmita, A. F. Wicaksono, S. Louvan and M. Adriani, "Unsupervised aspect-based sentiment analysis on Indonesian restaurant reviews," *IEEE,* 2017.

[64] A. Caranica, H. Cucu and A. Buzo, "Exploring an unsupervised, language independent, spoken document retrieval system," *IEEE,* 2016.

[65] P. Louridas and C. Ebert, "Machine Learning," *IEEE,* 2016.

[66] Z. Zong and C. Hong, "On Application of Natural Language Processing in Machine Translation," *IEEE,* 2018.

[67] T. Peng, I. Harris and Y. Sawa, "Detecting Phishing Attacks Using Natural Language Processing and Machine Learning," *IEEE.*

[68] D. T. Nguyen, "Interactive document retrieval system based-on natural language query processing," *IEEE,* 2015.

[69] P. Kłosowski, "Deep Learning for Natural Language Processing and Language Modelling," *IEEE,* 2018.

[70] E. Khan, "Machine Learning Algorithms for Natural Language Semantics and Cognitive Computing," *IEEE,* 2016.

[71] a. T. Mylavarapu, "A Deep Learning Approach for sleuthing Disease-Treatment Relations in brief Texts," *IEEE,* 2018.

[72] M. B. L. Virtucio, J. A. Aborot, J. K. C. Abonita, R. S. Aviñante, R. J. B. Copino, M. P. Neverida, V. O. Osiana, E. C. Peramo, J. G. Syjuco and G. B. A. Tan, "Predicting Decisions of the Philippine Supreme Court Using Natural Language Processing and Machine Learning," *IEEE,* 2018.

[73] C.-K. Hung, “Making machine-learning tools accessible to language teachers and other non-techies: T-SNE-lab and rocanr as first examples,” *IEEE,* 2017.

[74] H. Bais, M. Machkour and L. Koutti, “Querying database using a universal natural language interface based on machine learning,” *IEEE,* 2016.

[75] Y.-F. Liao and Y.-R. Wang, “Some Experiences on Applying Deep Learning to Speech Signal and Natural Language Processing,” *IEEE,* 2018.

[76] N. Mukai, N. Harada and Y. Chang, “Japanese Fingerspelling Recognition Based on Classification Tree and Machine Learning,” *IEEE,* 2017.

[77] M. F. Kabir, K. Abdullah-Al-Mamun and M. N. Huda, “Deep learning based parts of speech tagger for Bengali,” *IEEE,* 2016.

[78] S. S. Dodal and P. V. Kulkarni, “Multi-Lingual Information Retrieval Using Deep Learning,” *IEEE,* 2018.

[79] Z. S. Ritu, N. Nowshin, M. M. H. Nahid and S. Ismail, “Performance Analysis of Different Word Embedding Models on Bangla Language,” *IEEE,* 2018.

[80] A. Rokade, B. Patil, S. Rajani, S. Revandkar and R. Shedge, “Automated Grading System Using Natural Language Processing,” *IEEE,* 2018.

[81] L. Dostert, “The Georgetown-I.B.M. experiment.,” in *Machine translation of languages*, Mass, M.I.T.Press, 1955, p. 124–135..

[82] M. a. H. N. Popovic, “POS-based Word Reorderings for Statistical Machine Translation,” in *LREC*, 2006.

[83] N. a. H. N. Ueffing, “Using pos information for statistical machine translation into morphologically rich languages,” in *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*, 2003.

[84] D. W. G.-A. L. a. C. I. C. Oard, “CLEF experiments at Maryland: Statistical stemming and backoff translation,” in *Workshop of the Cross-Language Evaluation Forum for European Languages*, Berlin, Heidelberg, 2000.

[85] J. a. D. K. Trommer, “A morphological tagger for standard Albanian,” in *Proceedings of LREC*, 2004.

[86] K. a. A. B. Hoxha, “An Automatically Generated Annotated Corpus for Albanian Named Entity Recognition,” *Cybernetics and Information Technologies,* vol. 18, no. 1, pp. 95-108, 2018.

[87] M. P. a. M. B. Skënduli, “A named entity recognition approach for albanian,” in *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, 2013.

[88] G. a. K. H. Kono, “Named Entity Recognition in Albanian Based on CRFs Approach,” in *RTA-CSIT*, 2016.

[89] B. Kabashi, “Building an Albanian Text Corpus for Linguistic Research,” in *Corpus-Based Approaches to the Balkan Languages and Dialects*, 2016.

[90] J. e. a. Sylak-Glassman, “ A universal feature schema for rich morphological annotation and fine-grained cross-lingual part-of-speech tagging,” in *International Workshop on Systems and Frameworks for Computational Morphology*, Cham, 2015.

[91] N. R. v. W. a. T. S. Aepli, “Part-of-speech tag disambiguation by cross-linguistic majority vote,” in *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, 2014.

[92] A. Kadriu, “NLTK tagger for Albanian using iterative approach,” in *Proceedings of the ITI 2013 35th International Conference on Information Technology Interfaces*, 2013.

[93] B. a. T. P. Kabashi, “Albanian Part-of-Speech Tagging: Gold Standard and Evaluation,” in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, 2018.

[94] J. e. a. Nivre, “Universal Dependencies v1: A Multilingual Treebank Collection.,” in *LREC*, 2016.

[95] A. D. Alban Rashiti, “Adaption of Levenshtein Algorithm for Albanian Language,” *2017 International Conference on Computational Science and Computational Intelligence,* 2017.

[96] B. Raufi and I. Xhaferri, “Application of Machine Learning Techniques for Hate Speech Detection in Mobile Applications,” *2018 International Conference on Information Technologies (InfoTech),* 2018.

[97] M. B. M.Skënduli, “A Named Entity Recognition Approach for Albanian,” *IEEE,* 2013.

[98] B. Hasanaj, “A Part of Speech Tagging Model for Albanian: An innovative solution.,” *LAP LAMBERT Academic Publishing,,* 2012.

[99] J. e. a. Nivre, “Universal Dependencies v1: A Multilingual Treebank Collection.,,” *IEEE,* 2016.

[100] *. H. A. BAXHAKU., “ALBANIAN LANGUAGE IDENTIFICATION IN TEXT DOCUMENTS,” *IEEE,* 2017.

[101] K. T. M. G. a. K. Z. Christopher Schreiner, "Using Machine Learning Techniques to Reduce Data Annotation Time," in *Human Factors and Ergonomics Society Annual Meeting Proceedings*, USA, 2014.

[102] M. H. J. A. B. S. B. R. Diellza Nagavci Mati, "A Systematic Mapping Study of Language Features Identification from Large Text Collection," in *2019 8th MEDITERRANEAN CONFERENCE ON EMBEDDED COMPUTING (MECO), 10-14 JUNE 2019*, BUDVA, 2019.

[103] M. H. J. A. B. R. B. S. Diellza Nagavci Mati, "Review of Natural Language Processing tasks in Albanian language," in *3rd International Scientific Conference on Business and Economics*, Tetovo, 2019.

[104] D. Dahlmeier, "On the Challenges of Translating NLP Research into Commercial Products," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, 2017.

[105] S. B. R. S. &. N. G. Bodapati, "A Machine Learning Approach to Detecting Start Reading Location of eBooks.," in *IEEE*, 2018.

[106] S. P. Dipanjan Das, "Unsupervised Part-of-Speech Tagging with Bilingual Graph-Based Projections.," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics,*, 2011, 2011.

[107] B. W. (. Erik Cambria, "Jumping NLP Curves: A Review of Natural. Digital Object Identifier," in *MCI*, 2015.

[108] A. M. H. a. A. D. Susuri, "Machine learning based detection of vandalism in Wikipedia across languages.," in *5th Mediterranean Conference on Embedded Computing (MECO).*, Budva, 2016.

[109] M. H. &. G. Fliedl, "Text Preparation Through Extended Tokenization," *IEEE,* vol. 36, p. 9, 2016.

[110] K. L. Odile Piton, "Morphological study of Albanian words, and processing with NooJ," *Barcelone, Spain,* pp. pp.189-205, 2011.

[111] C. Biemann, "Chinese Whispers - an Efficient Graph Clustering Algorithm and its Application to Natural Language Processing Problems," *IEEE,* 2017.

[112] S. Bordag, "Word Sense Induction:Triplet-Based Clustering and Automatic Evaluation," *Natural Language Processing Department,* 2006.

[113] K. K. W. K. A. K. a. B. S. Atta Ur Rahman, "Unsupervised Machine Learning based Documents Clustering in Urdu," *EAI Endorsed Transactions on Scalable Information Systems,* vol. Volume

5 , no. 19, pp. 1-14, 2018.

[114] Stephen M. Stigler, “Darwin, Galton and the Statistical Enlightenment,” *Journal of the Royal Statistical Society, Series A,* pp. 469-482, 2010.

[115] Dekking, M, “A Modern Introduction to Probability and Statistics.,” *London : Springer ,* pp. pp. 181-190, 2005.

[116] P. B.P and H. Dhami, “Application of Natural Language Processing Tools in Stemming,” *International Journal of Computer Applications (0975 – 8887),* pp. 50-57, 2011.

[117] . C. T.T, L. C. E. and R. R. L., “Intrudation to algorithms,” in *MIT Press*, Cambridge, 1990.

[118] A. M Robertson and P. Willett, “Applications of N-grams in textual information systems,” *Journal of Documentation,* vol. 54, pp. 50-63, 1998.

[119] A. G. Jivani, “A Comparative Study of Stemming Algorithms,” *Int. J. Comp. Tech. Appl.,,* vol. 2 (6), pp. 1930-1938, 2011.

[120] S. Millaku, “The Albanian words , the morphemes and the prefixes,” *European Journal of English Language Teaching,* vol. 2, no. 4, p. 15, 2020.

[121] I. Çollaku and E. Adali, “Morphological parsing of Albanian language: a different approach to Albanian verbs,” in *University of Business and Technology in Kosovo*, Kosovo, 2015.

[122] B. Kabashi, “Collecting Collocations for the Albanian Language,” in *Proceedings of eLex* , Portugal, 2019.

[123] C. Biemann, S. Bordag, G. Heyer, U. Quasthoff and C. Wolff, “Language-independent Methods for Compiling Monolingual Lexical Data,” in *CICLing*, Germany, 2004.

[124] I. C. Education, “Natural Language Processing (NLP),” IBM, 2 July 2020. [Online]. Available: https://www.ibm.com/cloud/learn/natural-language-processing. [Accessed 07 07 2020].

[125] L. Edward , E. Klein and B. Standford, “Preface,” Natural Language Processing, 15 10 2012. [Online]. Available: https://www.nltk.org/book_1ed/ch00.html. [Accessed 07 06 2020].

[126] D. Jurafsky and J. H. Martin, “Regular Expressions, Text Normalization, Edit Distance,” *Speech and Language Processing,* pp. 1-30, 2021.

[127] B. Li and L. Han , “Distance Weighted Cosine Similarity Measure for Text Classification,” 2013.

APPENDIX

Experimental Results

Table 68. Number of apperance and frequency in the Albanian language sources

Words	Nr appearance	Frequency	Total word count from the file
të	49992	7.923%	631008
e	37165	5.890%	
në	19577	3.102%	
i	15194	2.408%	
dhe	14436	2.288%	
që	10510	1.666%	
me	10316	1.635%	
për	9226	1.462%	
një	6226	0.987%	
është	5728	0.908%	
se	5542	0.878%	
nga	5520	0.875%	
më	5242	0.831%	
së	4319	0.684%	
do	4051	0.642%	
nuk	3955	0.627%	
si	3703	0.587%	
edhe	3433	0.544%	
u	3079	0.488%	
ka	2537	0.402%	
janë	2534	0.402%	
mund	2412	0.382%	
shumë	2144	0.340%	
ta	2083	0.330%	
por	2069	0.328%	
Në	1769	0.280%	
duke	1762	0.279%	
tyre	1557	0.247%	
këtë	1538	0.244%	
duhet	1433	0.227%	
ose	1430	0.227%	
kanë	1345	0.213%	
vetëm	1318	0.209%	
politike	1227	0.194%	

Table 69 Hyphenated tokens from all sources

Hyphenated Words	Nr appearance	Frequency %	Total word count from the file
			631008
UE-së	974	0.1544%	
UE-ja	143	0.0227%	
juridiko-civile	113	0.0179%	
BEE-së	93	0.0147%	
UE-në	67	0.0106%	
PE-së	43	0.0068%	
juridiko-private	38	0.0060%	
SHBA-ve	34	0.0054%	
PE-ja	34	0.0054%	
njëri-tjetrit	33	0.0052%	
BEE-	28	0.0044%	
njëra-tjetrën	27	0.0043%	
njëri-tjetrin	26	0.0041%	
juridiko-civil	24	0.0038%	
UNMIK-ut	22	0.0035%	
20-të	22	0.0035%	
NATO-s	21	0.0033%	
holandezo-flamane	20	0.0032%	
BETHÇ-së	18	0.0029%	
PE-në	15	0.0024%	
njëra-tjetrës	14	0.0022%	
anglo-saksone	13	0.0021%	
RSFJ-së	12	0.0019%	
ish-komuniste	12	0.0019%	
OKB-së	12	0.0019%	
evro-atlantike	11	0.0017%	
ECJ-së	11	0.0017%	
SHBA-të	10	0.0016%	
ish-Jugosllavi	10	0.0016%	
dy-tre	10	0.0016%	
pluralo-partiake	10	0.0016%	
19-të	10	0.0016%	
dy-tri	10	0.0016%	
AKU-së	10	0.0016%	
dita-ditës	9	0.0014%	
ish-Jugosllavisë	9	0.0014%	
Belkiz-hanmi	9	0.0014%	
NATO-ja	9	0.0014%	
stabilizim-asocimit	9	0.0014%	
BE-së	9	0.0014%	

Table 70. Apostrophes tokens in the Albanian language sources

Words	Nr appearance	Frequency %	Total word count from the file
t'	772	1.223439	631008
s'	343	0.543575	
ç'	87	0.137875	
m'	29	0.045958	
S'	27	0.042789	
Ç'	21	0.033280	
d'	12	0.019017	
n'	10	0.015848	
T'	6	0.009509	
N'	4	0.006339	
D'	3	0.004754	
gjith'	2	0.003170	
l'	2	0.003170	
Lal'	2	0.003170	
Rek'	2	0.003170	
shtetasit'	2	0.003170	
Association'	1	0.001585	
Dhiat'	1	0.001585	
dit'	1	0.001585	
dor'	1	0.001585	
erdh'	1	0.001585	
flak'	1	0.001585	
J'	1	0.001585	
Kopenhagen'	1	0.001585	
kur'	1	0.001585	
lajn'	1	0.001585	
Mit'	1	0.001585	
Or'	1	0.001585	
përher'	1	0.001585	
përshëndet'	1	0.001585	

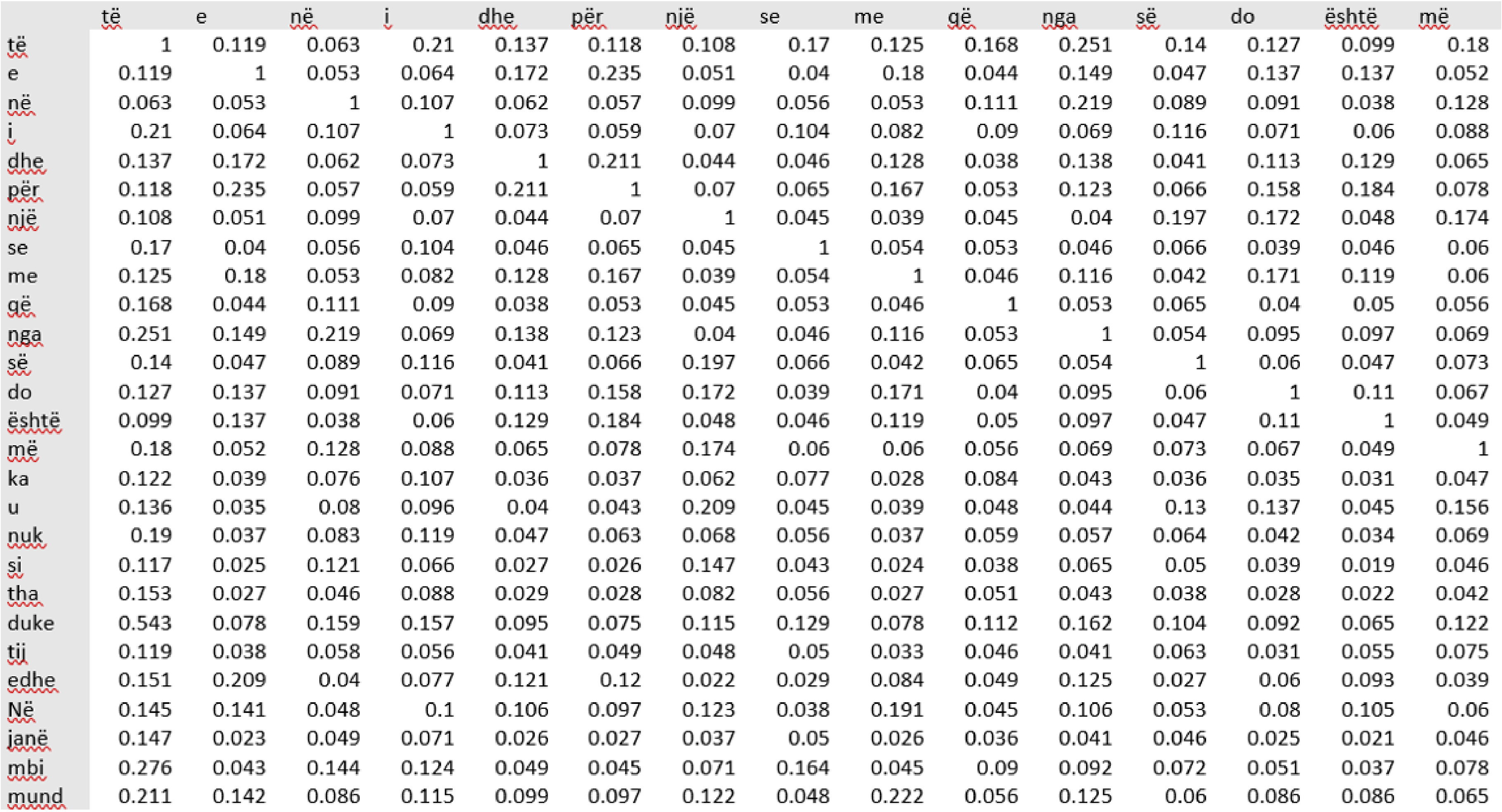
Table 71. Tokens from left apostrophe in the Albanian language sources

Words	Nr appearance	Frequency %	Total word count from the file
i	606	0.960368	631008
u	193	0.305860	
ia	49	0.077654	
është	47	0.074484	
ka	40	0.063391	
mund	35	0.055467	
ua	30	0.047543	
do	28	0.044373	
ju	23	0.036450	
ishte	22	0.034865	
e	20	0.031695	
kam	20	0.031695	
më	20	0.031695	
kishin	15	0.023771	
t	15	0.023771	
rast	13	0.020602	
iu	10	0.015848	
kemi	10	0.015848	
ishin	9	0.014263	
na	9	0.014263	
kishte	8	0.012678	
kish	7	0.011093	
di	6	0.009509	
dinin	6	0.009509	
janë	6	0.009509	
kanë	6	0.009509	
po	6	0.009509	
dinte	5	0.007924	
keni	5	0.007924	
kuptonte	5	0.007924	
masë	5	0.007924	
duhet	4	0.006339	
Esten	4	0.006339	
jemi	4	0.006339	

Table 72. Matrix from the 10.000 words of the Albanian language

	të	e	në	i	dhe	për	një	se	me	që	nga	së	do	është	më	ka	u	nuk	si	tha
të	13173	1521	969	468	701	1028	682	467	310	842	152	245	1273	102	378	218	66	308	144	31
e	851	737867	1079	7720	43896	781	156	26565	41777	260	26850	82	2294	315	108	6633	2377	7616	9415	4051
në	1264	636	5515	235	244	255	176	209	115	273	122	175	107	140	107	117	101	45	45	25
i	271	14175	117	322753	19250	43	157	17954	2701	148	15844	48	1541	208	94	1715	1881	3634	7290	4999
dhe	552	41784	236	10976	279889	143	53	3334	9676	39	6990	74	1205	39	35	1887	1724	777	7605	465
për	660	432	200	167	164	3220	180	95	80	83	41	101	34	98	65	102	38	39	51	11
një	486	246	306	97	84	191	2167	126	102	93	66	36	78	189	31	103	35	29	77	12
se	346	11894	134	6220	8148	71	22	174792	2542	57	1246	55	1819	80	128	4615	2541	1527	313	22727
me	444	18542	131	6332	8859	57	72	2920	168212	69	1829	35	3887	56	28	2723	6795	1238	1487	357
që	464	301	182	104	158	95	135	62	87	2507	59	44	30	118	33	71	11	29	16	8
nga	314	15595	62	6388	5355	36	57	2833	2568	129	122071	29	2974	62	27	1291	5745	918	1612	276
së	417	371	31	310	35	15	10	17	13	13	12	1356	8	3	10	5	1	1	3	25
do	163	15302	84	7689	11052	27	17	21226	2377	161	1641	31	120926	13	14	865	201	9020	1041	2528
është	110	157	89	98	66	46	23	215	22	111	18	31	2	1314	13	10	7	118	7	19
më	186	90	51	25	42	42	97	35	28	41	30	6	21	53	954	21	12	21	9	2
ka	97	12381	78	7954	3964	30	10	10008	1223	129	968	36	289	6	8	74518	688	7623	701	877
u	76	7654	48	3960	5659	10	7	2112	1156	66	1059	12	82	8	6	121	68886	1789	545	372
nuk	112	7813	58	3304	4986	34	11	12849	1048	89	1021	19	149	5	8	489	234	67307	377	1601
si	128	9926	36	2640	3758	13	3	2785	1175	33	900	11	951	19	5	990	1073	547	57228	110
tha	71	5175	36	4738	2137	20	5	195	997	4	579	18	202	4	9	108	2034	185	390	48359
duke	55	5761	28	1038	2615	10	5	1120	1138	8	736	5	217	14	3	339	374	195	236	1061
tij	88	25109	26	5737	3105	16	6	3072	2547	6	1275	33	33	3	0	61	27	4	131	110
edhe	160	3931	30	1469	1243	10	7	1126	701	38	405	8	1332	35	3	1590	813	282	3405	79
Në	70	58	59	11	21	19	7	2	3	5	10	12	0	2	11	3	0	1	4	6
janë	114	88	36	19	50	15	2	71	9	82	12	7	0	0	8	4	4	63	3	5
mbi	45	4056	12	1203	1414	3	11	724	900	10	376	5	536	3	1	452	313	254	179	68
BE	13	8635	27	3962	2103	11	1	1204	4133	2	1346	0	86	1	0	55	138	32	126	220
mund	66	3558	26	1599	2002	8	12	5173	600	53	408	10	1327	4	3	196	71	5597	448	360
kanë	79	127	40	41	51	9	1	64	10	98	12	14	0	1	5	0	6	45	3	5
këtë	61	92	141	8	25	90	6	28	41	29	2	7	12	5	3	11	5	2	1	3
tyre	58	19434	31	1826	1719	21	5	740	971	8	824	9	12	2	0	27	29	9	111	5
shumë	91	49	36	18	30	21	41	39	28	29	5	3	14	52	163	50	4	20	6	4
ishte	59	2865	18	2404	870	4	1	4369	336	17	365	12	2977	2	5	50	184	1825	374	1098

Table 73. Matrix obtained results from the calculation of cosine similarity.



	të	e	në	i	dhe	për	një	se	me	që	nga	së	do	është	më
të	1	0.119	0.063	0.21	0.137	0.118	0.108	0.17	0.125	0.168	0.251	0.14	0.127	0.099	0.18
e	0.119	1	0.053	0.064	0.172	0.235	0.051	0.04	0.18	0.044	0.149	0.047	0.137	0.137	0.052
në	0.063	0.053	1	0.107	0.062	0.057	0.099	0.056	0.053	0.111	0.219	0.089	0.091	0.038	0.128
i	0.21	0.064	0.107	1	0.073	0.059	0.07	0.104	0.082	0.09	0.069	0.116	0.071	0.06	0.088
dhe	0.137	0.172	0.062	0.073	1	0.211	0.044	0.046	0.128	0.038	0.138	0.041	0.113	0.129	0.065
për	0.118	0.235	0.057	0.059	0.211	1	0.07	0.065	0.167	0.053	0.123	0.066	0.158	0.184	0.078
një	0.108	0.051	0.099	0.07	0.044	0.07	1	0.045	0.039	0.045	0.04	0.197	0.172	0.048	0.174
se	0.17	0.04	0.056	0.104	0.046	0.065	0.045	1	0.054	0.053	0.046	0.066	0.039	0.046	0.06
me	0.125	0.18	0.053	0.082	0.128	0.167	0.039	0.054	1	0.046	0.116	0.042	0.171	0.119	0.06
që	0.168	0.044	0.111	0.09	0.038	0.053	0.045	0.053	0.046	1	0.053	0.065	0.04	0.05	0.056
nga	0.251	0.149	0.219	0.069	0.138	0.123	0.04	0.046	0.116	0.053	1	0.054	0.095	0.097	0.069
së	0.14	0.047	0.089	0.116	0.041	0.066	0.197	0.066	0.042	0.065	0.054	1	0.06	0.047	0.073
do	0.127	0.137	0.091	0.071	0.113	0.158	0.172	0.039	0.171	0.04	0.095	0.06	1	0.11	0.067
është	0.099	0.137	0.038	0.06	0.129	0.184	0.048	0.046	0.119	0.05	0.097	0.047	0.11	1	0.049
më	0.18	0.052	0.128	0.088	0.065	0.078	0.174	0.06	0.06	0.056	0.069	0.073	0.067	0.049	1
ka	0.122	0.039	0.076	0.107	0.036	0.037	0.062	0.077	0.028	0.084	0.043	0.036	0.035	0.031	0.047
u	0.136	0.035	0.08	0.096	0.04	0.043	0.209	0.045	0.039	0.048	0.044	0.13	0.137	0.045	0.156
nuk	0.19	0.037	0.083	0.119	0.047	0.063	0.068	0.056	0.037	0.059	0.057	0.064	0.042	0.034	0.069
si	0.117	0.025	0.121	0.066	0.027	0.026	0.147	0.043	0.024	0.038	0.065	0.05	0.039	0.019	0.046
tha	0.153	0.027	0.046	0.088	0.029	0.028	0.082	0.056	0.027	0.051	0.043	0.038	0.028	0.022	0.042
duke	0.543	0.078	0.159	0.157	0.095	0.075	0.115	0.129	0.078	0.112	0.162	0.104	0.092	0.065	0.122
tij	0.119	0.038	0.058	0.056	0.041	0.049	0.048	0.05	0.033	0.046	0.041	0.063	0.031	0.055	0.075
edhe	0.151	0.209	0.04	0.077	0.121	0.12	0.022	0.029	0.084	0.049	0.125	0.027	0.06	0.093	0.039
Në	0.145	0.141	0.048	0.1	0.106	0.097	0.123	0.038	0.191	0.045	0.106	0.053	0.08	0.105	0.06
janë	0.147	0.023	0.049	0.071	0.026	0.027	0.037	0.05	0.026	0.036	0.041	0.046	0.025	0.021	0.046
mbi	0.276	0.043	0.144	0.124	0.049	0.045	0.071	0.164	0.045	0.09	0.092	0.072	0.051	0.037	0.078
mund	0.211	0.142	0.086	0.115	0.099	0.097	0.122	0.048	0.222	0.056	0.125	0.06	0.086	0.086	0.065

Table 74. POS tagging for the Albanian language obtained results from the 500 words.

Verbs	**Nouns**	**Adjective**	**Numeral**	**Pronoun**
ishte	herë	bukur	një	Na
jetonte	vajzë	shkonte	dy	Ato
kishte	ditë	xheloze	gjasht	e cila
agjëroj	Athinë	administrativ	Pese	I cili
absolutizoj	konferencë	poshtërues	shtatë	Ne
frenoj	stadium	komik	dhjetë	Ata
ishin	vajza	vjetra		
veshë	punë	martua		
binte	nata	përfunduar		
qante	xheloze	ulej		
përfundoj	punët	morrën		
adaptoj	stacion	pozitiv		
mbathi	filluan			
takonte	njerka			
këpute	pëhitura			
pyeste	princi			
humbën	festë			
largu	zana			
nënshtroj	fenomen			

Table 75. POS tagging results from 100 words of the Albanian language

Verbs	**Nouns**	**Adjective**	**Pronoun**
ishte	herë	bukur	Na
jetonte	vajzë	shkonte	Ato
nënvlerësoj	Afrikë	konsumues	Ne
tregoj	ditë	premtues	Ajo
lazdroj	Angli	problematik	i cili
kishte	nata		
ishin	vajza		
punë			

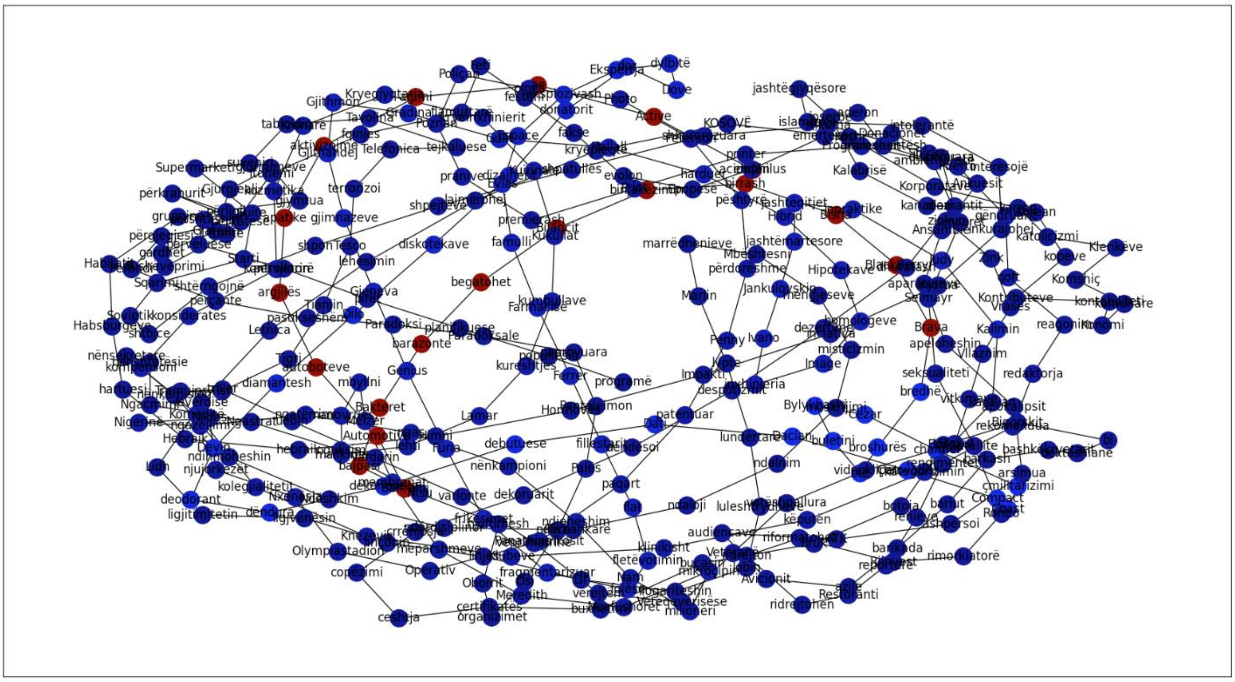


Figure 41. Chinese Whispers algorithm in 500 words of the Albanian language.

Table 76. Different sources of percentage comparison

Case	13 Sources	73 Sources
M_w [>1.0]	48% (119.894)	63% (157.362)
M_w [0.1-1]	37% (92.418)	29% (72.436)
M_w [<0.1]	15% (37.469)	8% (19.983)