Undergraduate Thesis

# LINGUISTIC AND TOKENIZATION ANALYSIS OF THE MELTEMI LLM FOR CYPRIOT GREEK

**Marios Koumides**

**UNIVERSITY OF CYPRUS**

**DEPARTMENT OF COMPUTER SCIENCE**

**May 2025**

# UNIVERSITY OF CYPRUS

## DEPARTMENT OF COMPUTER SCIENCE


**Linguistic and Tokenization analysis of the Meltemi LLM for Cypriot Greek**


**Marios Koumides**

Supervising Professor

Dr. Constantinos Pattichis


Thesis submitted in partial fulfilment of the requirements for the award of degree of Batchelor's in Computer Science from the University of Cyprus

May 2025

## Acknowledgments

Firstly, I would like to thank my supervisor professor Constantinos Pattichis, for giving me the opportunity to undertake this specific topic. I would also like to thank him for the feedback and valuable advice he provided throughout my work.

I also wish to express my thanks to Ms.Rebecca Pattichis for her invaluable support and expertise, which was especially enlightening during the course of my thesis. Her assistance and the guidance she offered proved to be crucial for the development and quality of my work.

## Abstract

This Thesis explores the use of the Meltemi Large Language Model (LLM) for the processing and analysis of Cypriot Greek language prompts, with the aim of identifying specific linguistic challenges and suggesting methods to evaluate the model’s performance. The research emphasizes the syntactic and semantic characteristics of the language and evaluates their impact on the performance of Meltemi. The study employs both qualitative and quantitative approaches, combining linguistic analysis with machine learning experiments to gain insights into model behavior. The key aspects of this work will use comprehensive data sets for Greek and Cypriot Greek prompts, an analysis of linguistic challenges faced by LLMs, and proposed methodologies to analyze language model responses, with the aim of advancing language technologies for underrepresented languages.

# Contents

## Graphs and tables list

## Abbreviations

| | |
|---|---|
| CG | Cypriot Greek |
| SMG | Standard Modern Greek |
| LLM | Large Language model |
| NLP | Natural Language Processing |
| AI | Artificial Intelligence |

# Chapter 1

## Introduction



### *1.1 Context and motivation*

The advent of large language models (LLMs), such as GPT and BERT, has revolutionized natural language processing (NLP) by enabling the generation of human-like text, answering questions, and performing complex linguistic tasks [10]. However, these models often face limitations when dealing with languages that are underrepresented in the training data. Cypriot Greek is a prime examples of such language, characterized by unique syntactic structures, rich morphology, and context-dependent meanings that pose challenges to LLMs. This project addresses the gap in NLP research for these languages, with the goal of improving model understanding and performance.

### *1.2 Research Objectives*

The primary objectives of this thesis are to investigate the performance of the Meltemi LLM on Cypriot Greek language prompts, identify linguistic characteristics that contribute to the models' limitations, and develop methods and tools for analysing the handling of Greek and Cypriot Greek language tasks. Understanding how Meltemi processes Cypriot Greek and investigating its tokenizer transferability, is crucial for enhancing its accuracy and effectiveness. By analyzing common failure points and linguistic nuances, this research aims to bridge the gap in NLP applications for Greek and Cypriot Greek. Furthermore, this study explores methodologies for evaluating the performance of the Meltemi model in underrepresented language varieties like Cypriot Greek, with a particular focus on metric design and linguistic analysis. The insights gained from this research aim to contribute to the broader field of NLP by improving our understanding of how language models handle less commonly studied dialects.

# Chapter 2

## Literature review



### *2.1 Large Language Models (LLMS)*

The field of Natural Language Processing (NLP) has significantly evolved with the advent of Large Language Models (LLMs). These models, such as GPT and BERT, have demonstrated remarkable capabilities in understanding and generating human-like text [10].However, their effectiveness varies across different languages and dialects. LLMs are deep learning-based models trained on vast amounts of text data to understand and generate human-like language. Key models include GPT (Generative Pre-trained Transformer), developed by OpenAI, which uses a transformer architecture and is trained on diverse datasets, enabling high adaptability across various tasks. BERT (Bidirectional Encoder Representations from Transformers), unlike GPT, processes text bidirectionally, making it highly effective for tasks requiring contextual understanding [10]. T5 (Text-to-Text Transfer Transformer) is another model that reformulates NLP tasks as text-to-text problems, allowing for unified processing of diverse linguistic inputs. Despite their advancements, these models exhibit limitations when handling languages with limited training data, such as Greek and its dialects.

### *2.2 Prompt Engineering & Analysis*

Prompt engineering involves designing inputs that guide LLMs to generate accurate and relevant outputs. Effective prompt crafting includes zero-shot prompting, where a direct query is provided without examples, few-shot prompting, which includes a few examples to guide the model, and chain-of-thought prompting, which encourages the model to explain its reasoning step by step [6]. Challenges in prompt engineering include biases in training data, difficulty in controlling model responses, and inconsistencies in performance across different languages. Studies have shown that English-dominant training skews LLM performance, making it crucial to explore prompt strategies for underrepresented languages. In this context, our work involves prompting the Meltemi model using Cypriot Greek prompts to evaluate how well the model can interpret and respond to inputs in a low-resource dialect.

### *2.3 Greek & Cypriot Greek in NLP*

Greek NLP research has focused on syntactic parsing, machine translation, and sentiment analysis. However, Cypriot Greek, a dialect with significant linguistic differences, remains underexplored. Key challenges include limited annotated datasets, as few high-quality datasets exist for Cypriot Greek. Additionally, the morphological complexity of Greek and its dialects, which exhibit rich inflectional variations, complicates model training. Another major challenge is code-switching, where many speakers mix Greek and English in informal contexts, further complicating NLP tasks.

### *2.4 Gaps in Existing Research*

While LLMs perform well in high-resource languages, their effectiveness in Greek and Cypriot Greek remains uncertain. One of the main gaps in existing research is the limited understanding of how well Standard Greek language models generalize or transfer to regional dialects such as Cypriot Greek. While LLMs have shown impressive results in Standard Modern Greek, their ability to process and represent dialectal variation remains largely unexplored. In particular, there is a lack of evaluation-focused studies that assess the linguistic alignment between model outputs and dialect-specific features. Addressing these gaps will contribute to a more inclusive and effective NLP framework for underrepresented languages.

# Chapter 3

## The Meltemi chatbot



### *3.1 Introduction to Meltemi*

Meltemi represents the first significant attempt at developing an open LLM specifically designed for the Greek language. Developed by the Institute for Speech and Language Processing of the Athena Research Center, Meltemi aims to address the notable underrepresentation of Greek in LLMs, typically dominated by languages such as English and Chinese [1]. This chapter explores the development process, capabilities, and implications of Meltemi, highlighting its importance in supporting Greek linguistic and cultural representation in Artificial Intelligence (AI).

### *3.2 Development of Meltemi*

The development of Meltemi involved the adaptation and continuous pretraining of the Mistral 7B model [6]. Mistral 7B is based on a Transformer architecture consisting of 32 layers, designed initially for general-purpose language modeling tasks, exclusively trained on English datasets. The process of adapting this model to Greek included several key phases.

### 3.2.1 Data Acquisition and Preprocessing

To build Meltemi, researchers compiled a comprehensive corpus totaling approximately 54.5 billion tokens, which included Greek monolingual data, English monolingual data, and parallel English-Greek translation data [1]. Sources ranged from Wikipedia, academic repositories, ELRC-SHARE (European Language Resource Coordination Sharing), a platform developed to facilitate the sharing of multilingual resources collected from European public administrations, aiming to support language technologies across Europe and CulturaX, a multilingual dataset containing large volumes of cleaned web-based texts in various languages, designed specifically for training large-scale language models. Extensive preprocessing was carried out, including removal of low-quality data, using MinHashLSH to remove duplicate or near-duplicate textual content, and filtering for irrelevant content [1].

### 3.2.2 Tokenizer and Embedding Expansion

A significant challenge in adapting Mistral 7B for Greek was the inefficiency of its original tokenizer for handling Greek language texts [1]. Initially, the tokenizer showed higher computational costs for Greek, necessitating its expansion from 32,000 tokens to 61,362 tokens. This strategic adaptation substantially improved computational efficiency, reducing resource requirements and speeding up training and inference processes.

### 3.2.3 Continuous Pretraining

Continuous pretraining refers to the process of further training an existing pre-trained language model on new and specific datasets. This approach helps the model adapt effectively to new languages [6].

Continuous pretraining for Meltemi occurred in two phases [1]. Initially, the focus was only on training the new embeddings, which are the numerical representations of the newly added Greek tokens. The rest of the model was not changed during this phase. The purpose was to help these new tokens integrate smoothly with the model's existing knowledge without causing significant disruption. In the second phase, the complete model was retrained extensively to fully adapt it to Greek language usage. This comprehensive retraining employed carefully managed learning rates and gradient to ensure optimal adaptation and performance [1].

### 3.3 Instruction Tuning and Alignment

Meltemi 7B Instruct, the chatbot variant of Meltemi, was created by aligning the model with human preferences through instruction tuning. This involved translating and curating twelve preference datasets, ensuring the absence of toxic or biased responses, and training the model on a final dataset of approximately 97,072 preference instances [1]. The instruction tuning leveraged special tokens to denote different conversational roles, enabling Meltemi 7B Instruct to effectively engage in meaningful and context-aware interactions.

### 3.4 Evaluation and Performance

Meltemi underwent extensive evaluation across a range of benchmarks to measure its language understanding, reasoning, and task-specific performance:

#### 3.4.1 Greek Evaluation Benchmarks

Meltemi demonstrated significant improvements over its predecessor, Mistral 7B, across various Greek datasets, including ARC Greek, Truthful QA Greek, HellaSwag Greek, MMLU Greek, Belebele, and Greek Medical Multiple Choice QA. The average performance increase on Greek tasks was approximately 20.2% [1].

#### 3.4.2 Comparative English Benchmarks

Although primarily optimized for Greek, Meltemi also underwent evaluation on English benchmarks, demonstrating a moderate performance decline of approximately 6% compared to the original Mistral 7B [1]. This reduction in English performance aligns with trends observed in other language-specific adaptations of foundational models.

### 3.5 Discussion and Implications

Meltemi represents a pivotal step in the democratization of AI, particularly for languages with smaller linguistic communities. Its development highlights the feasibility and effectiveness of continuous pretraining approaches for adapting existing models to underrepresented languages. Despite the moderate decline in multilingual performance, Meltemi successfully addresses critical linguistic and cultural representation needs in AI, providing a foundation for further research and development.

# Chapter 4

## The Naïve Bayes classifier



### *4.1 Text Classification*

Text classification refers to the task of assigning a label to a given piece of text. For example, we might want to determine whether a sentence expresses a positive or negative opinion, or in our case, whether it is written in Cypriot Greek or Standard Modern Greek. To achieve this, we train a model on many examples of labeled text so that it can learn the patterns that are typical of each category. Once the model is trained, it can be given a new sentence and decide which category it most likely belongs to. This type of task is common in various real-world applications, including spam detection, language identification, and automatic news classification.

### *4.2 The Naive Bayes Classifier*

The Naive Bayes classifier is a probability based method. It works by looking at which words, or tokens appear in a sentence and comparing them to what it has learned from the training data. The idea is to figure out which category is most likely, based on the presence of certain words. For example, if words like “εν”, “μαναμου” or “κάμνω” appear in a sentence, and these words were often seen in Cypriot Greek during training, the model might conclude that the sentence is likely to be Cypriot Greek. Similarly, if the sentence contains more standard Greek expressions, it will lean towards classifying it as Standard Modern Greek. What makes this model “naive” is that it assumes each word contributes to the final decision independently from the others. In reality, words are often related to one another in context, but this simplifying assumption makes the model fast and easy to use and in many cases, accurate enough for practical purposes.

### *4.3 How the Classifier Learns*

During the training phase, the Naive Bayes model reads many example sentences that have already been labeled as either Cypriot Greek or Standard Modern Greek. It then learns how common each word or token is in each type of text and how often each category appears in the training data overall. With this information, when the model sees a new sentence, it looks at each word in the sentence and checks whether those words were more commonly found in Cypriot or Standard Greek texts during training. Based on that, it decides which label is the best match [2]. This process follows the principles of Bayes' Theorem, adapted for text classification using the Multinomial Naive Bayes model. The model estimates the class ccc that maximizes the probability [2]:

$$\hat{c} = \arg\max_{c \in C} P(c) \prod_{i=1}^{n} P(w_i \mid c)$$

Where:

- c^: the predicted class (CG or SMG)
- P(c): the prior probability of class c
- $P(w_i \mid c)$: the probability of word/token $w_i$ given class c
- n: total number of tokens in the sentence
- The product is over all tokens in the input sentence

Each token's probability $P(w_i \mid c)$ is calculated using Laplace smoothing, to avoid assigning zero probability to unseen words. This is given by [2]:

$$P(\text{word} \mid \text{class}) = \frac{\text{count of word in class} + \alpha}{\text{total word count in class} + \alpha \cdot V}$$

Where:

- a is the smoothing parameter (typically 1)
- V is the vocabulary size (total number of unique tokens)
- The numerator counts how often the word appears in the class, with smoothing
- The denominator counts the total sum of words across the vocabulary

This smoothed estimate is critical when working with tokenized text from language models like Meltemi, where many sub-word tokens may be rare or unique to a specific dialect. By combining frequency information and smoothing, the model can classify unseen sentences robustly, based on their token composition and similarity to known examples from each dialect.

### *4.4 Why Naive Bayes is Useful*

There are several reasons why Naive Bayes is a popular and powerful choice for text analysis tasks like ours. First of all, it is very fast, both to train and to make predictions, it works well even with limited data, which is important in dialectal research where large datasets are rare. It also provides interpretable results, meaning we can easily see which words or tokens the model considers important for its decisions. This is especially valuable in our project, where understanding the difference between Cypriot and Standard Greek is just as important as being able to classify correctly.

### *4.5 Application in our research*

In this study, we used the Naive Bayes classifier to distinguish between sentences written in Cypriot Greek and Standard Modern Greek. What makes our approach unique is that instead of using traditional words or character-based features, we used tokens generated by the Meltemi tokenizer, which breaks text into sub-word units in a way that reflects how large language models process text. We trained the model using two sets of sentences: one from Cypriot Greek texts and one from Standard Greek texts. Each sentence was tokenized using the Meltemi model, and then those tokens were counted and used as input features for the Naive Bayes classifier. The model learned which tokens are more likely to appear in one variety of Greek versus the other. After training, we tested the classifier on new, unseen sentences. The model was able to correctly identify the variety of Greek in most cases. We also analysed which tokens were the most "informative", that is, the ones that the model relied on most when making its decision. This helped us better understand which linguistic features are most distinctive of Cypriot Greek, even at the token level.

### 4.6 Limitations of the Naive Bayes Approach

While Naive Bayes is a powerful tool, it is not without its limitations. Its main weakness lies in the assumption that all words or tokens are independent from each other. In natural language, this is rarely true. The meaning of a sentence often depends on how words are combined and ordered. Additionally, Naive Bayes does not take context or word sequences into account. For tasks where word order is important or where subtle differences matter, more advanced models such as deep neural networks or transformer-based models may perform better. However, for our task, which focuses on vocabulary and token patterns, Naive Bayes remains a strong and reliable choice.

# Chapter 5

## Tools used and data analysis methodology



### *5.1 Introduction*

This chapter presents the tools, resources, and methodologies used in the analysis and classification of Cypriot Greek (CG) and Standard Modern Greek (SMG) texts. The study aims to evaluate the effectiveness of modern language models, specifically the Meltemi model, in identifying and distinguishing between these two languages. The work conducted builds upon an existing body of research in dialect classification and incorporates new approaches using large language model (LLM) tokenization and scoring metrics.

### *5.2 Programming Environment*

The implementation of this research was carried out entirely in Python. All experiments and data analyses were conducted using Jupyter Notebooks. Several key Python libraries were used throughout the study. The Natural Language Toolkit (NLTK) provided essential functionality for processing text, including tokenization and sentence segmentation. Scikit-learn was employed for implementing machine learning models, including the Multinomial Naive Bayes classifier, as well as for handling data vectorization and performance evaluation. For interaction with large language models, the HuggingFace Transformers library was used.

This library enabled access to the Meltemi tokenizer, allowing us to tokenize Greek texts in a manner consistent with how language models process and understand input. Additionally, Pandas and Matplotlib were used for data analysis, tabulation, and visualization of results.

### 5.3 Prior Research

The starting point for our methodology was the study titled "A Classifier to Distinguish Between Cypriot Greek and Standard Modern Greek" by Sababa and Stassopoulou (2018) [3]. Their work introduced a simple yet effective method for classifying social media text into the two Greek language varieties. The authors relied on a relatively small dataset of labeled sentences and used traditional machine learning techniques to perform the classification. Their core approach involved three steps: preprocessing the text, extracting n-gram features (such as word-level and character-level, bigrams and trigrams), and training classifiers including multinomial Naive Bayes, linear Support Vector Classifier and logistic regression. Among these, Naive Bayes achieved the highest mean accuracy of 95%, outperforming the other methods and proving particularly effective for distinguishing between dialects [3].
Although we were inspired by the structure of their pipeline, our study took a different direction in several key areas. Most notably, we integrated a large language model (Meltemi) to tokenize and process the text and generate the token n-grams. Then, we used Naive Bayes' feature ranking of importance and introduced custom metrics to measure how "Cypriot" a sentence is based on its token composition.

### 5.4 Training and Evaluation Datasets

Two primary datasets were used as the training data. These datasets were carefully selected for their linguistic relevance, authenticity, and domain diversity. The Cypriot Greek dataset was generously provided by Dr. Spyros Armostis and originates from a project conducted in 2022 [9]. In this project, eleven Cypriot authors collaborated to produce a corpus of authentic Cypriot Greek text. The participants were specifically trained to write in Cypriot Greek using the orthographic system designed by Dr. Armostis, which standardizes dialectal spelling for Cypriot Greek [9]. The dataset includes dialogic and narrative styles, offering a rich and diverse representation of the dialect. Its use in this thesis provides a valuable resource that reflects real world usage of Cypriot Greek in written form.

```
",μιαν δημιουργικήν ιστορίαν,Θετική Σκέψη
1,MA,"Μόλις εκάτσασιν στ’ αυτοκίνητον αρκέψασιν.
-Παππού, άτε.
-Άτε ίντα πράμαν;, λαλώ τους.
-Ανέκδοτον.
-Ρε αχάπαροι, συνεχίζω, εθωρούσετε το όνειρον;
-Εν’ που την άλλην Τετάρτην που είπες πως εννά μας πεις σήμερα.
```

[9]

For the Standard Modern Greek dataset, the source was a GitHub repository related to the *nlp_greek_storytelling* project [7]. This open-source corpus includes well-formed SMG sentences drawn from narratives, news articles, and general purpose texts. The data is clean and they reflect a standard usage of the Greek language, making it ideal for comparison with the more dialectally marked Cypriot Greek dataset. For the purposes of this thesis, we specifically used the “paramithia.txt” file from the repository, which contains a collection of narrative texts in Standard Modern Greek [7]. This selection was made to match the narrative and dialogic nature of the Cypriot Greek dataset, ensuring a more balanced and meaningful comparison between the two language varieties. Together, these two datasets form a reliable foundation for both classification and analysis. They enable both the training of machine learning models and the testing of new metrics, ensuring that our results are based in real linguistic data.

```
Το κουνελάκι και ο παππούλης
Μια φορά και έναν καιρό ήταν ένα κουνελάκι μικρό και πονηρό. Όπως σε όλα τα
κουνελάκια έτσι και στο δικό μας άρεσαν πολύ τα καροτάκια και τα τρυφερά
μαρουλάκια. Είχε λοιπόν βρει ένα λαχανόκηπο και έσκαβε λαγούμια και έμπαινε μέσα
και έτρωγε με την ψυχή του.
```

[7]

To assess the effectiveness of the proposed metrics and the model’s handling of Cypriot Greek in unseen text, two evaluation datasets were compiled independently of the training data. The first dataset consists of a series of manually written Cypriot Greek prompts, each designed to reflect realistic and diverse linguistic constructs. The responses to these prompts were generated by the Meltemi model, allowing us to evaluate how effectively the model produces dialectal text. The second dataset includes human-written responses to Cypriot Greek prompts, offering a benchmark for authentic Cypriot Greek usage [9]. These texts span narrative, poetic, and dialogic forms, ensuring coverage of different stylistic registers and vocabulary. Together, the two datasets serve as comparative tools for evaluating the model’s generative capabilities and for validating the reliability of our metrics under real-world conditions.

```
Γράψε έναν ποιήμαν γραμμένην στα κυπριακά με τίτλο Το Βούρος.
<|assistant|>
Το βορύς, ο όμορφος και δυνατός,
Βόσκει τα ζωντανά στην πλαγιά του λόφου·
Με το χαριτωμένο κουδούνι γύρω από τον λαιμό του,
Χορεύει και πηδάει καθώς βόσκει ήσυχα την ημέρα του.
```

CG – Evaluation – Meltemi output

PI10. Το Βούρος
Ούλλοι βουρούσιν
τζ' εν θωρούν
Πού πάσιν πιον τα
τρία

CG – Evaluation – Human output

| Number of sentences from each dataset | | | |
|---|---|---|---|
| **CG - Training** | **SMG – Training** | **CG – Evaluation -Meltemi output** | **CG- Evaluation – human output** |
| 1988 | 1988 | 132 | 656 |

Table3: Number of sentences from datasets

### *5.5 Data analysis methodology*

The first step of data analysis is using code for generating our desired results. Our code performs the initial data cleaning, tokenization, and statistical analysis of the input text [3] [8].

We begin by preprocessing the text to remove any unwanted noise. This includes stripping punctuation, converting all characters to lowercase, and removing English words, URLs, emojis, and special symbols [8]. One important aspect of the preprocessing phase is the removal of any sentence that contains characters from the Latin alphabet. Since both Cypriot and Standard Greek use the Greek alphabet, this step ensures that the data remains strictly monolingual. Following cleaning, we use the Meltemi tokenizer to segment each sentence into tokens. The Meltemi tokenizer breaks text into subword units that the model can process. We use this to extract n-grams. Specifically:

- **Unigrams**: single tokens
- **Bigrams**: sequences of two consecutive tokens

These n-grams are useful for capturing short, repeating patterns in the language. For example, a Cypriot Greek sentence may frequently contain bigrams like “εν να” or “πάω κάτω”, while Standard Greek might exhibit bigrams such as “θα πάμε”. This differs from traditional tokenizers that might simply split text by spaces or punctuation. The next part of our code involves analysing token frequencies. For both dialects (CG and SMG), the most frequently occurring tokens are computed. These high-frequency tokens serve as indicators of dialect-specific vocabulary and grammatical structures.

### *5.6 The Classifier*

The second part of the code, builds a machine learning classifier capable of identifying whether a sentence is written in Cypriot Greek or Standard Modern Greek [3] [8]. In this part of the code, the cleaned and preprocessed sentences are once again tokenized using the Meltemi tokenizer. These tokens are then converted into a form that a machine learning model can process. Specifically, the token sequences are joined into strings and transformed into numerical vectors using the CountVectorizer tool from Scikit-learn [8]. This tool creates a matrix in which each row represents a sentence and each column represents the frequency of a particular token. Once the data is vectorized, it is split into training and test sets. The training set is used to teach the classifier to recognize patterns, while the test set is used to evaluate how well the classifier performs on unseen data. The classifier used is the Multinomial Naive Bayes model, which is well-suited for text classification tasks involving word or token frequencies [3] [8]. The model is then trained and evaluated. Key performance metrics such as accuracy and the confusion matrix are computed. These results allow us to understand how well the model distinguishes between CG and SMG, and where it may still make mistakes.

### *5.7 Evaluation metric 1 (CyGr Score)*

Beyond simply classifying sentences as either Cypriot Greek or Standard Modern Greek, this thesis introduces a more nuanced method for evaluating how characteristic a given sentence is of the Cypriot Greek dialect. This is achieved through a custom-designed metric called the CyGr Score. The motivation behind this metric is to move beyond binary labels and instead assess how much a sentence *resembles* Cypriot Greek in its vocabulary. A sentence may not be written entirely in the dialect but could contain several Cypriot elements. We use this to measure the evaluation data success, meaning how well does the model outputs its responses in Cypriot Greek. The CyGr Score is implemented as a function that operates on a sentence-by-sentence basis. It works by comparing the tokens of a sentence to a list of the most frequently used tokens in Cypriot Greek. These frequent tokens are extracted from the

Cypriot training dataset after tokenization using the Meltemi tokenizer. Once the list of top tokens is created, the metric evaluates new sentences by checking how many of their tokens also appear in the top-n list of Cypriot tokens. The calculation is straightforward: the sentence is first passed through the same Meltemi tokenizer, and the resulting tokens are compared to the top-*n* Cypriot tokens. The metric then returns the percentage of the sentence's tokens that are also found in the top-n Cypriot token list. A higher percentage indicates that the sentence is composed largely of vocabulary that is highly characteristic of the Cypriot Greek dialect. This approach allows us to score individual sentences on a scale from 0 to 100, where 0 means that the sentence shares no tokens with the top Cypriot tokens, and 100 means that all of its tokens are found in the Cypriot token list. This score serves as a probabilistic proxy for dialectal influence and is particularly useful when analysing texts that contain a mixture of Cypriot and Standard Greek elements. Importantly, this metric does not rely on machine learning classification or predefined rules. Instead, it uses token frequency derived from actual Cypriot Greek data to assess similarity. This makes it simple to compute and easy to interpret, while still providing meaningful insights into how dialectal a sentence is. The CyGr Score complements the classification model by offering a continuous evaluation, where texts can be ranked or filtered based on their dialectal weight. In the next chapter, we will present examples of how this metric performs across a variety of sentences and discuss how well it aligns with human intuition regarding dialectal usage.

### *5.8 Evaluation metric 2 (Zipf's Law)*

Zipf's Law [5] is a fundamental statistical observation in natural language processing, originally formulated by linguist George Kingsley Zipf. It posits that the frequency of a word in a language corpus is inversely proportional to its rank in the frequency table. Formally, if we denote a word's rank by r and its frequency by f(r), Zipf's Law can be expressed as [5]:

$$f(r) \propto \frac{1}{r^s}$$

where s is a constant typically close to 1 in natural language, and the most frequent word has rank r = 1. A remarkable property of Zipf's Law is that it applies across virtually all human languages. When the frequency of words is plotted against their rank on a log-log scale, the result is a nearly straight line with a negative slope (approximately -1), demonstrating the power-law distribution of word usage. This phenomenon suggests that language is highly structured: a small number of words are used very frequently, while the majority occur rarely.

This distributional pattern reflects the statistical regularities that emerge from human language production and comprehension. Therefore, measuring how closely a body of text follows Zipf's Law provides insight into how natural and authentic it is.

### 5.8.1 Application to Cypriot Greek Evaluation

In the context of this thesis, Zipf's Law is used as a statistical tool to evaluate the linguistic structure of Cypriot Greek in a set of evaluation texts. The objective is to determine whether the evaluation data follows a frequency distribution consistent with the known patterns of Cypriot Greek, as represented by a training corpus. The core idea is that if the evaluation text exhibits a Zipfian distribution similar to that of authentic Cypriot Greek training data, it suggests that the text aligns with the expected statistical properties of natural language use in the dialect. While this does not confirm linguistic correctness, it supports the hypothesis that the evaluation data reflects a naturally distributed use of Cypriot Greek.

### 5.8.2 Methodology

Token frequencies were computed using standard tokenization techniques, and the tokens were subsequently sorted in descending order by frequency. For each dataset, we then transformed the rank and frequency values using the natural logarithm, thereby preparing the data for Zipfian visualization. The resulting log-log values were used to generate Zipf plots, where each point represents a word's frequency versus its rank. To quantify the degree to which each dataset conforms to Zipf's Law, a linear regression model was fitted to the log-log data [5]. The slope of the fitted line indicates how closely the data follows the expected power-law distribution, with a slope near -1 suggesting strong adherence to Zipfian behaviour. Additionally, the coefficient of determination ($R^2$) was computed to assess the goodness of fit; values closer to 1 indicate that the data closely follows a linear trend on the log-log plot.

This analysis serves as a structural benchmark for evaluating how natural the evaluation texts are from a statistical and linguistic perspective. Unlike simpler metrics that rely solely on lexical overlap with known Cypriot Greek tokens, Zipfian analysis captures the overall distributional characteristics of the language, offering a more holistic and theoretically grounded evaluation. Texts that deviate significantly from the Zipfian pattern may either lack linguistic structure, represent different dialects or registers, or include noise and artificial content. In the following section, we present the Zipf plots derived from both the evaluation text and the Cypriot Greek training data. We visually inspect the similarity in shape, and we also compare the slope and $R^2$ values to quantify their correspondence. These results help us assess whether the evaluation data conforms to the linguistic regularities expected of authentic Cypriot Greek.

# Chapter 6

## Data analysis results

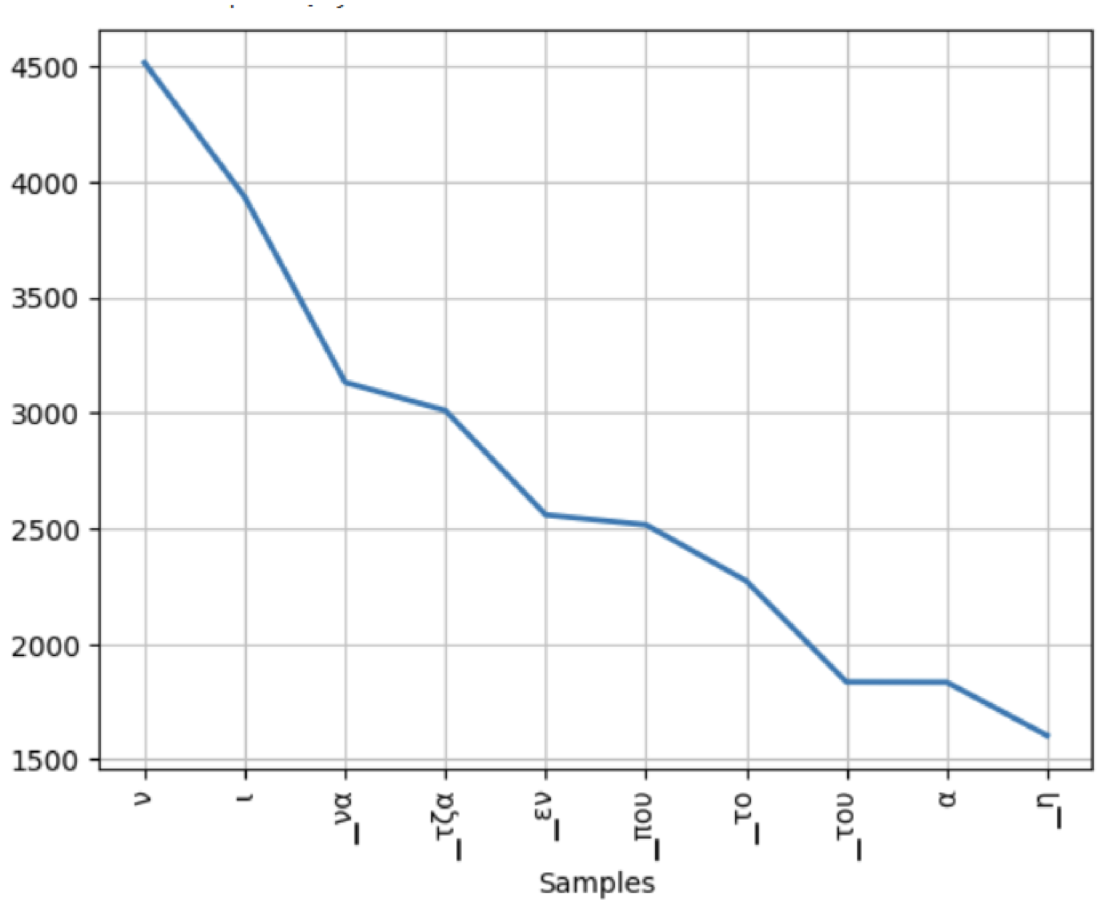


### *6.1 Introduction*

In this chapter, we present and interpret the results of the data analysis conducted on Cypriot Greek (CG) and Standard Modern Greek (SMG) texts. The purpose of this analysis is to reveal the linguistic features that distinguish the two language varieties, using both frequency based metrics and classification outputs. The visualizations in this chapter were generated from the tokenized and cleaned datasets using the Meltemi tokenizer, and they reflect the most common tokens, unigrams and bigrams found in each dialect. These visual summaries help us identify patterns in word usage and grammar that are typical of Cypriot Greek compared to Standard Greek. Each section below is dedicated to a specific type of token or n-gram, accompanied by a graph that visualizes the frequency of the most common elements in each dataset. Following these sections, a comparison is provided to highlight the key differences observed between the two dialects based on these visualizations.

### *6.2 Cypriot Greek-Training Data Most Frequent Unigrams*

The first graph presents the most frequent Meltemi tokens found in the Cypriot Greek corpus. These tokens provide insight into common vocabulary patterns that are specific to the dialect. The "_" that is included at many tokens (e.g _να) is inserted by the meltemi tokenizer when the token is the beginning of a new word.

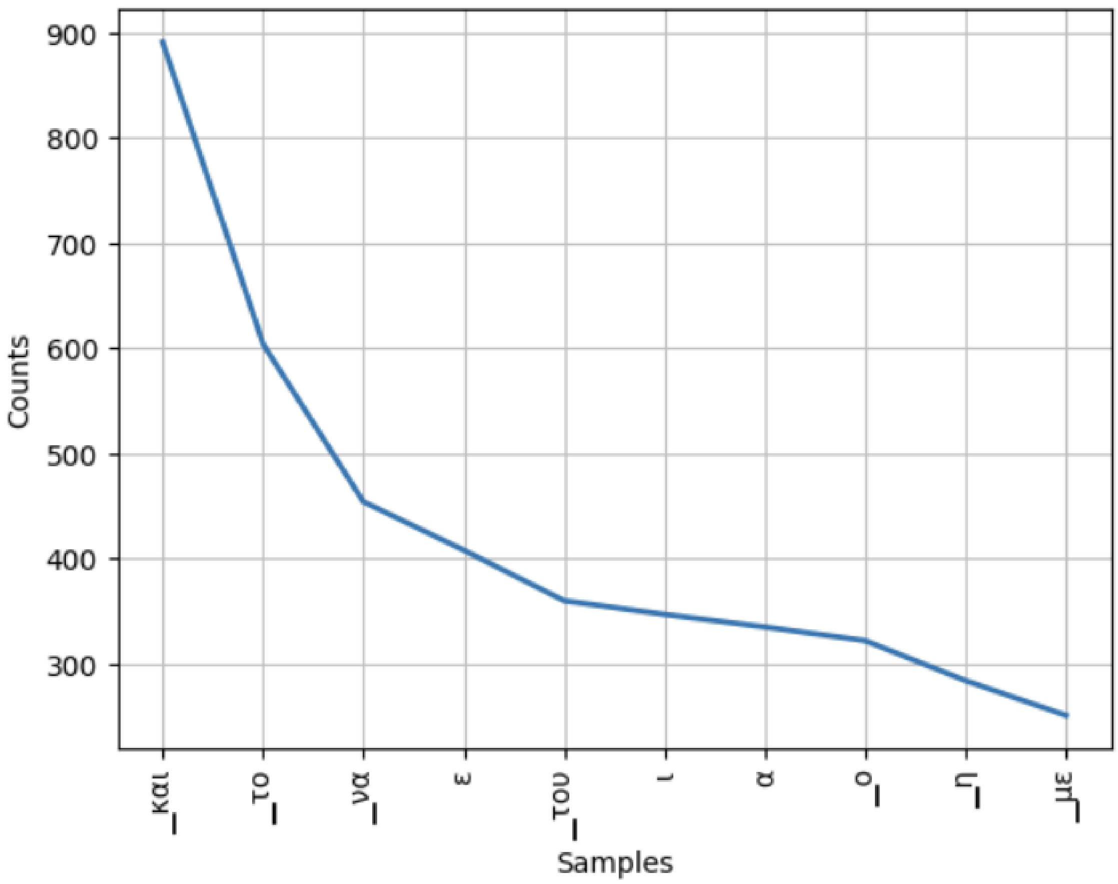



Graph1: Cypriot Greek most frequent Unigrams

The data shows that tokens such as "ν", "να", "τζα", and "εν" dominate the frequency distribution. These appear in a significantly higher frequency in the Cypriot Greek training data.

### *6.3 Standard Greek-Training Data Most Frequent Unigrams*

This graph displays the most commonly used tokens in the Standard Greek dataset, again using the Meltemi tokenizer.

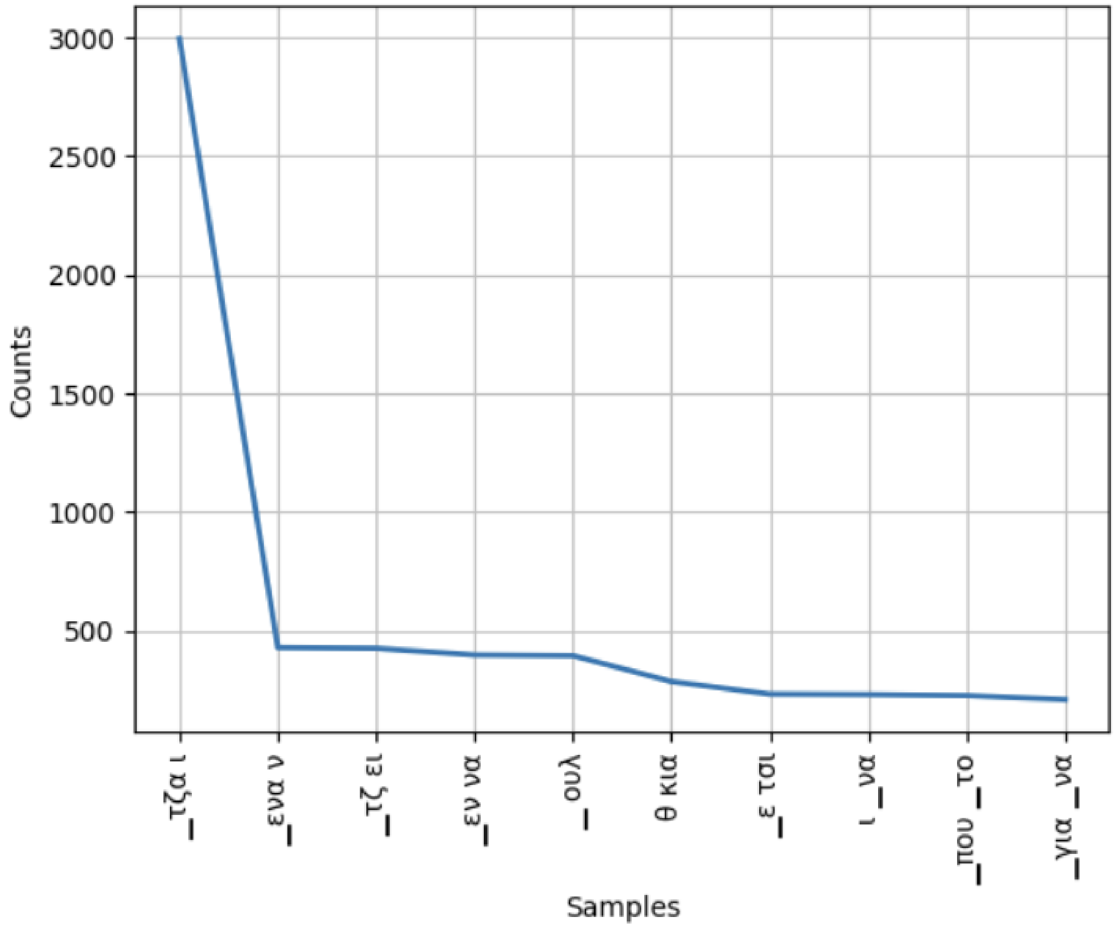



Graph2: Standard Greek most frequent Unigrams

In this case, the most frequent tokens include “και”, “το”, “του”, and “να”. These reflect standardized grammar and usage, aligning closely with what is expected in formal or narrative Standard Greek. We can observe that most of the frequent tokens, appears to be either conjunctions (e.g και) or pronouns (e.g να, του, με).

### *6.4 Cypriot Greek-Training Data Most Frequent Bigrams*

This chart visualizes the top 10 most frequent bigrams (sequences of two tokens) found in the Cypriot Greek dataset.

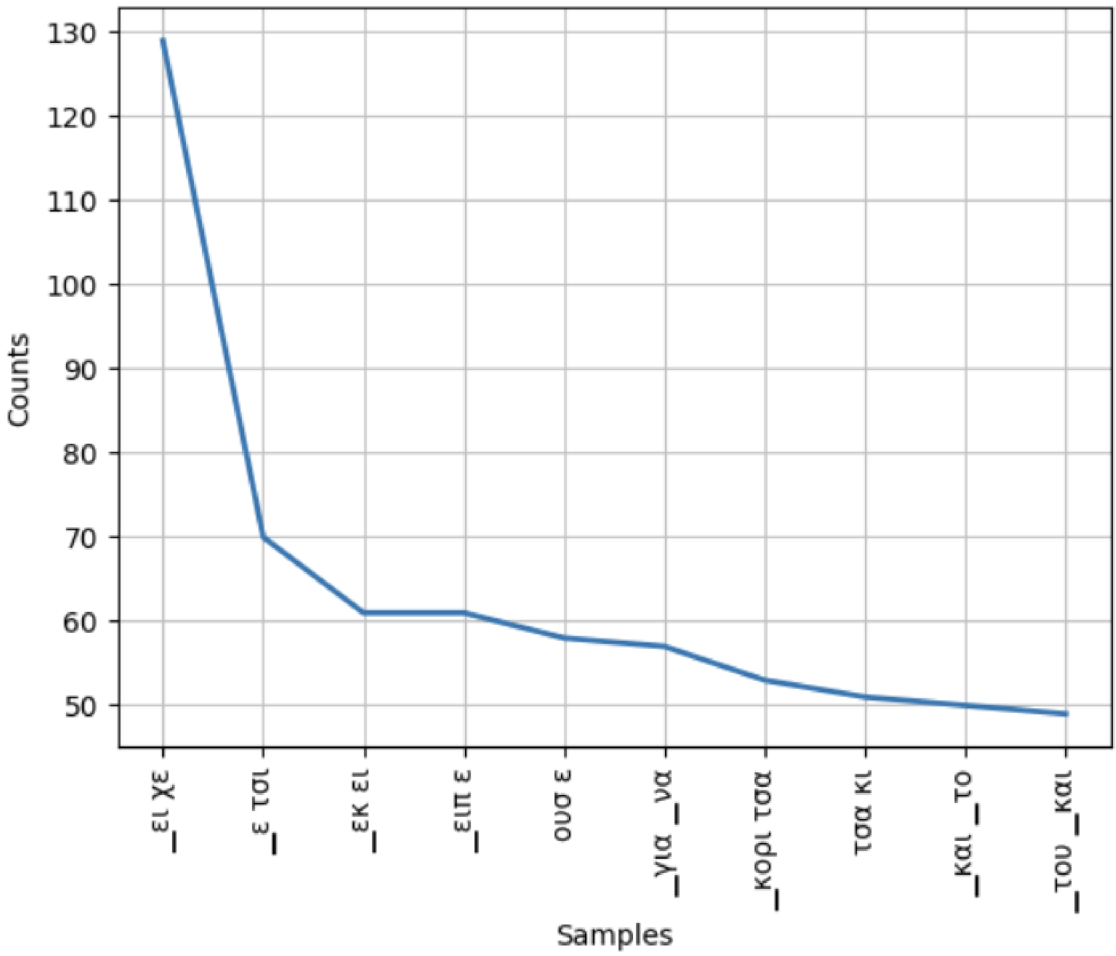



Graph3: Cypriot Greek most frequent Bigrams

The prevalence of bigrams such as “τζα ι”, “ενα ν”, or “τζ ει” highlights typical Cypriot structures and expressions. These sequences are commonly used in informal speech and demonstrate dialect specific grammatical constructions that are entirely absent in Standard Greek.

### *6.5 Standard Greek-Training Data Most Frequent Bigrams*

The following graph shows the most common bigrams in the Standard Greek dataset.




Graph4: Standard Greek most frequent Bigrams

Here, we observe more formal and standardized constructions such as "ει χε", "ε τσι", and "και το". These structures reflect written norms and grammatical consistency aligned with Standard Greek usage. The use of the token "και" is especially frequent and contrasts with "τζαι" in the Cypriot data.

An interesting pattern emerges when comparing the token frequency distributions between Cypriot Greek and Standard Modern Greek at the unigram and bigram level. In the unigram plots, the Cypriot Greek frequency curve is notably flatter than that of Standard Greek. This suggests a more even distribution of token usage in the Cypriot data, potentially due to greater lexical variation or less standardized phrasing across the corpus.
However, when we examine the bigram frequency plots, the pattern reverses. Cypriot Greek shows a steep initial drop, whereas Standard Greek exhibits a more gradual decline. This indicates that in the CG dataset, only a few bigram token combinations dominate, and the

frequency drops sharply after them. In contrast, the SMG data maintains a broader set of commonly recurring bigram pairs.

This flip in distributional behaviour may reflect how the Meltemi tokenizer, which was pre-trained on mostly Standard Greek, generalizes less effectively to Cypriot word combinations. While it may handle individual Cypriot tokens reasonably well (as seen in the flatter unigram curve), it appears to struggle with capturing consistent multi-token patterns in the dialect possibly due to limited exposure to Cypriot syntax during training.

This observation highlights a subtle but important limitation in tokenization transferability: a model trained on a standard variety might tokenize individual dialect words well, but still miss out on common patterns that are crucial for higher-level understanding or generation in different dialects.

### *6.6 Naive Bayes Classifier: Most Informative Features*

As part of the supervised classification task, we trained a Multinomial Naive Bayes model using Meltemi-tokenized input data to distinguish between Cypriot Greek (CG) and Standard Modern Greek (SMG) sentences [3] [8]. Once trained, one of the most valuable outputs of the model is its list of most informative features, that is, the tokens that contributed most strongly to the classification decision. These features are based on the likelihood ratios the model computes internally. In simpler terms, the model identifies which tokens are much more likely to appear in one category than the other. A high-scoring token for CG, for example, is one that appears frequently in CG texts but rarely or never in SMG texts.

| SMG | | CG | |
|---|---|---|---|
| -11.3645 | word(αιvο) | -4.7894 | word(ν) |
| -11.3645 | word(ακου) | -4.9609 | word(ι) |
| -11.3645 | word(ανε) | -5.1281 | word(_τζα) |
| -11.3645 | word(ανο) | -5.1340 | word_bigram(_τζα ι) |
| -11.3645 | word(βαι) | -5.2464 | word(_να) |
| -11.3645 | word(βαλε) | -5.2984 | word(_εν) |
| -11.3645 | word(βατα) | -5.3983 | word(_που) |
| -11.3645 | word(βελ) | -5.4924 | word(_το) |
| -11.3645 | word(βηκαν) | -5.5474 | word(_του) |
| -11.3645 | word(βηκε) | -5.6344 | word(α) |
| -11.3645 | word(βολο) | -5.7550 | word(_η) |
| -11.3645 | word(βου) | -5.8000 | word(_) |
| -11.3645 | word(βρου) | -5.8115 | word(_τζ) |
| -11.3645 | word(γαμε) | -5.8551 | word(_την) |
| -11.3645 | word(γαν) | -5.9708 | word(_ε) |
| -11.3645 | word(γεις) | -5.9938 | word(εν) |
| -11.3645 | word(γεται) | -6.0174 | word(_τα) |
| -11.3645 | word(γιο) | -6.0221 | word(ε) |
| -11.3645 | word(γκα) | -6.0366 | word(_με) |
| -11.3645 | word(γκας) | -6.0562 | word(_μου) |

Table1: Naïve bayes results

On the right, we see the top features for Cypriot Greek. These include many tokens that are either:

- Dialect-specific function words (e.g., ▁εν, ν, ▁τζαι)
- Subword particles commonly used in Cypriot syntax (e.g., ▁τζ, ▁να, ▁που)

For example:

word(▁τζαι) and word_bigram(▁τζα ι) are clear dialectal markers, as "τζαι" is a conjunction commonly used in CG in place of the Standard Greek "και". Tokens like ▁εν and ▁να are very frequent in CG due to their use in verbal constructions (e.g., "εν να πάω").

These values are the log probabilities assigned by the Multinomial Naive Bayes model to each token given the class.

For example:

- The number -4.7894 next to word(ν) under the CG (Cypriot Greek) column means:

  log(P(word = "ν" | class = CG)) = -4.7894

During training, the Multinomial Naive Bayes classifier estimates the probability of each token appearing in documents of a given class (CG or SMG). These are calculated using the formula:

$$P(\text{word} \mid \text{class}) = \frac{\text{count of word in class} + \alpha}{\text{total word count in class} + \alpha \cdot V}$$

Where:

- α is the smoothing factor (usually 1).
- V is the size of the vocabulary.

The classifier then stores logarithms of these probabilities for numerical stability and performance.

The CG column has more varied and less negative values (e.g. -4.78, -5.13, -6.05), indicating that those tokens appear more often and more informatively in the Cypriot Greek training data.

### 6.7 The GyGr score metric

Beyond classification, a key objective of this study was to develop a metric capable of quantifying how strongly a sentence reflects the Cypriot Greek dialect. This goal led to the implementation of the CyGr Score, a numeric indicator designed to evaluate individual sentences or texts based on their token-level similarity to Cypriot Greek vocabulary. As described in Chapter 5, the CyGr Score works by calculating the percentage of tokens in a sentence that match the top-*n* most frequently used tokens in the Cypriot Greek corpus. The score is expressed as a percentage between 0 and 100, where higher scores indicate stronger similarity to the Cypriot Greek dataset.

```
def CyGr_score(eval_sents_tokens, cg_Text, n):

    # Get the top-n most frequent Cypriot tokens from training data
    top_cy_tokens = set(token for token, _ in cg_Text.vocab().most_common(n))

    # Flatten all tokenized sentences into a single list of tokens
    eval_tokens = [token.lower() for sent in eval_sents_tokens for token in sent]

    if len(eval_tokens) == 0:
        return 0.0

    # Count how many tokens are in the top-n Cy tokens
    cy_token_matches = sum(1 for token in eval_tokens if token in top_cy_tokens)

    # Return percentage
    return (cy_token_matches / len(eval_tokens)) * 100
```

To evaluate the usefulness of the CyGr Score, we applied the metric in our evaluation datasets. Sentences from the Cypriot-Greek Meltemi output dataset and Cypriot-Greek human responses dataset were analyzed using the same list of top-*n* Cypriot Greek tokens, but changing the depending n-tokens each time to observe differences.

| Cypriot-Greek Meltemi output evaluation data | |
|---|---|
| Top-Tokens Considered (n) | GyGr score |
| 10 | 6.93% |
| 20 | 12.28% |
| 100 | 30.40% |
| 1000 | 73.83% |
| 10000 | 96.18% |

Table2: CyGr score results- Meltemi Output

As the results indicate, there is a clear upward trend in the CyGr Score as the number of considered tokens increases. For the Meltemi output dataset, when only the **top 10 tokens** are used, the average score is just **6.93%**, suggesting that these tokens, while highly frequent, appear in only a small fraction of the text. This makes sense, as such a small token list can only match a limited range of vocabulary. With **20 tokens**, the score rises to **12.28%**, and by **100 tokens**, the metric captures a broader lexical base, resulting in a score of **30.40%**. This shows that dialectal influence becomes more apparent when a slightly larger portion of the core vocabulary is taken into account. A more substantial jump occurs at **1,000 tokens**, where the CyGr Score climbs to **73.83%**. This suggests that a significant portion of the Cypriot dataset's vocabulary is concentrated within this token range, indicating both lexical consistency and stylistic repetition within the dialect sample. Finally, with **10,000 tokens**, the score reaches **96.18%**, which effectively confirms that nearly all of the tokens in the evaluation dataset appear in the top-10,000 list.

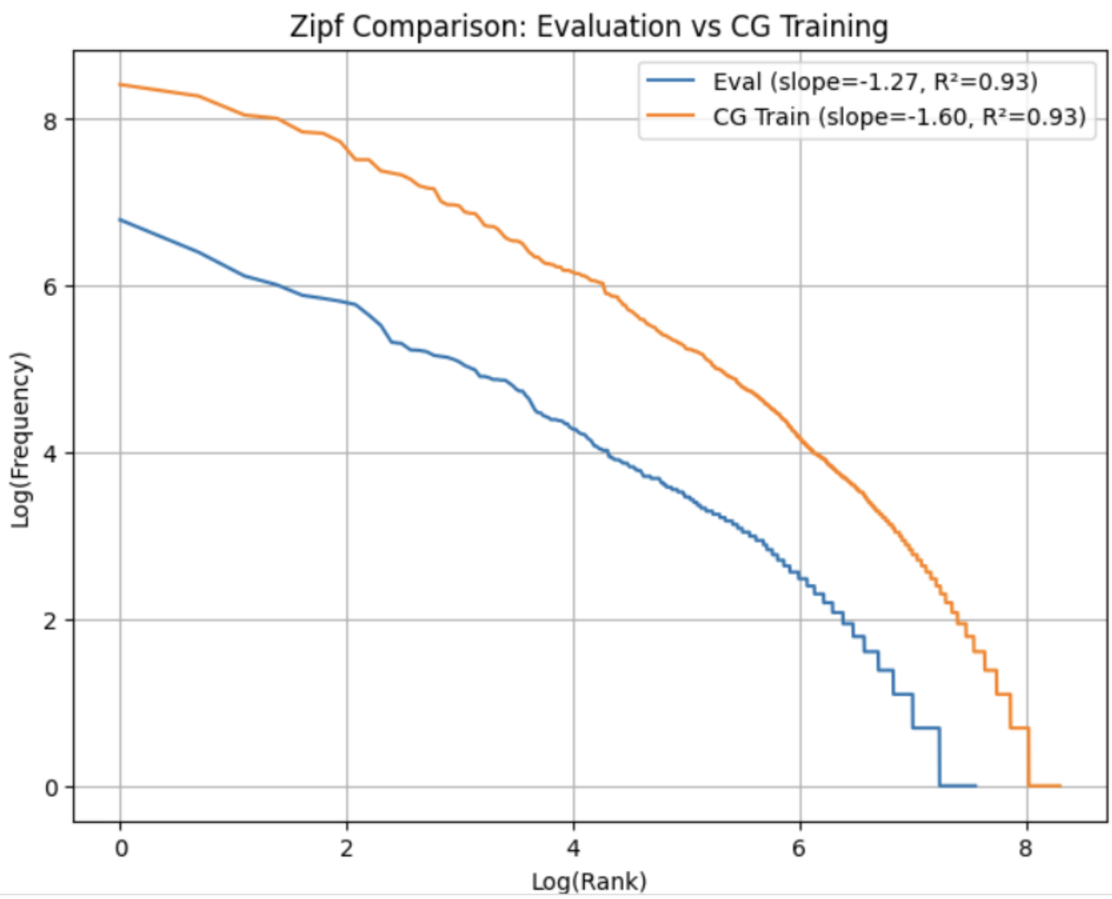
| Cypriot-Greek Human output evaluation data | |
|---|---|
| Top-Tokens Considered (n) | GyGr score |
| 10 | 17.74% |
| 20 | 26.80% |
| 100 | 52.03% |
| 1000 | 91.55% |
| 10000 | 99.84% |

Table4: CyGr score results- Human Output

This table presents the **CyGr Score results** for a dataset of human-written responses to Cypriot Greek prompts. As expected, the scores increase consistently with larger values of *n*. With only the top **10 tokens**, the CyGr Score is **17.74%**, already suggesting a notable dialectal signal. As *n* increases, the score rises sharply, reaching over **91% at 1000 tokens** and nearly full coverage (**99.84%**) at **10,000 tokens**. These values confirm that the human responses are strongly dialectal in nature, rich in vocabulary and constructions that match the most common forms in the training dataset. This behavior demonstrates the effectiveness of the CyGr metric in identifying authentic dialectal usage and provides a useful benchmark for evaluating other text sources.

### *6.8 Zipf's Law metric*

The figure below presents the Zipf plot comparing the evaluation dataset against the Cypriot Greek training corpus. Both datasets exhibit the characteristic downward-sloping trend when plotted on a log-log scale, indicating that they each approximately follow a Zipfian distribution. This confirms that the token frequencies in both corpora are governed by a power-law relationship, which is a hallmark of natural language structure.



Graph5: Zipf's Law results

A linear regression model was fitted to the log-transformed rank-frequency data of each dataset. The evaluation dataset yielded a slope of -1.27 with an $R^2$ value of 0.93, while the Cypriot Greek training corpus produced a steeper slope of -1.60, also with an $R^2$ of 0.93. The $R^2$ values in both cases are relatively high, indicating that the log-log data is well approximated by a linear model, thus validating the application of Zipf's Law in both corpora.

The difference in slope between the two curves is informative. The training data's steeper slope suggests a stronger concentration of word usage among the most frequent tokens — in other words, a small subset of words appears very frequently, while the rest are used sparingly. This is typical of large, coherent, and natural language corpora. In contrast, the evaluation data's slope is less steep, implying a slightly more even distribution of token usage, with less disparity between high-frequency and low-frequency words. This observation can be interpreted in several ways. Firstly, the similarity in $R^2$ values confirms that the evaluation data does follow the expected statistical behavior of natural language, supporting its linguistic validity. Secondly, the deviation in slope may reflect differences in register, domain, or stylistic variety within the evaluation text, especially if the content is more diverse, poetic, or informal than the training data. Alternatively, the flatter slope may indicate a smaller or more thematically varied corpus, which naturally leads to less repetition and thus weaker frequency dominance by a limited vocabulary.

Overall, the Zipfian analysis supports the hypothesis that the evaluation text exhibits a distributional structure consistent with natural language. While the slope is not identical to that of the Cypriot Greek training corpus, the strong linearity and high $R^2$ value suggest that the evaluation text aligns well with the statistical patterns observed in Cypriot Greek. This strengthens the case for its classification as Cypriot Greek, at least from a structural and statistical standpoint.

# Chapter 7

## Conclusions



### *7.1 Overview*

This thesis set out to explore how effectively Meltemi can process and interpret Cypriot Greek. The study combined traditional machine learning techniques with modern token-based methods derived from transformer models, aiming to identify meaningful patterns, frequencies, and distinguishing features in each dialect. Throughout the course of this work, two main analytical tools were developed and applied: the CyGr Score and Zipf's Law analysis. The former was used to quantify how "Cypriot" a sentence is based on token frequency, while the latter served as a statistical validation of the naturalness and structure of the dataset via frequency distribution.

### *7.2 Methods conclusions*

The CyGr Score, a metric specifically designed for this study, proved to be highly effective in evaluating how closely a sentence resembles Cypriot Greek. As demonstrated in the results, the percentage of recognition increased steadily as the number of top Cypriot tokens (*n*) used in the metric grew. For example, using just the top 10 tokens produced an average score of around 10%, while using 1,000 tokens yielded over 80%, and with 10,000 tokens, the score reached nearly 97%. Its simplicity and interpretability make it ideal for future applications in dialect research.

The Zipf Law analysis offered an alternative, statistical way to validate the structure and naturalness of the language data. The evaluation file's word frequency distribution followed a clear Zipfian pattern, with a high $R^2$ value of 0.93—comparable to the Cypriot training data. This confirms that the evaluation texts exhibit the typical word frequency behavior of natural human language. That shows that the output of the Meltemi model, follows a close to human writing in frequency dynamics. A slightly gentler slope in the evaluation set may reflect stylistic or thematic variation but does not compromise its linguistic authenticity.

Additionally, the use of a Naive Bayes classifier, trained using Meltemi tokenizer outputs, enabled automated classification between Cypriot and Standard Greek. The results showed strong accuracy and allowed for transparent interpretation through the most informative tokens. When combined with the CyGr Score and Zipf analysis, this approach offered a well-rounded toolkit for evaluating dialectal features.

## *7.3 Reflections on Meltemi's Tokenization*

A direct comparison between the CyGr Score results for human-written responses and Meltemi-generated responses reveals key insights into the model's handling of Cypriot Greek. While human texts consistently achieved high CyGr Scores, Meltemi's outputs scored significantly lower across all *n*-token thresholds. For instance, at $n = 10$, human responses scored **17.74%**, compared to only **6.93%** for Meltemi. Even at $n = 1000$, where human responses exceeded **91%**, Meltemi's output remained notably lower at **73.83%**. This performance gap suggests that although the model is partially capable of producing dialectal elements, it tends to default to more neutral or Standard Greek constructions and lacks consistent usage of highly frequent dialectal tokens.

The differences observed across these tables reinforce the idea that LLMs like Meltemi, even when exposed to Greek text during pretraining, do not fully understand dialectal patterns without targeted adaptation. The lower CyGr Scores in Meltemi's outputs point to a lack of strong representation of Cypriot Greek within the model's training distribution. These results highlight the importance of dialect-specific fine-tuning and confirm the CyGr Score's value in diagnosing such performance gaps between human and machine-generated language.

## *7.4 Future Research*

This thesis opens several paths for future research:

- Expanding the training and evaluation datasets with more thematically diverse and larger-scale corpora.
- Creating a custom tokenizer trained specifically on Cypriot Greek, which may better capture dialectal subword patterns.
- Applying neural network models, such as transformers or recurrent neural networks, for more context-aware classification.

# References


[1] Leon Voukoutis, Dimitris Roussis, Georgios Paraskevopoulos, Sokratis Sofianopoulos, Prokopis Prokopidis, Vassilis Papavasileiou, Athanasios Katsamanis, Stelios Piperidis and Vassilis Katsouros, “Meltemi: The first open Large Language Model for Greek”, Athena Research Center, Jul 2024

[2] Daniel Jurafsky & James H. Martin, “Speech and Language Processing”, Draft of January 12, 2025

[3] Hanna Sababa and Athena Stassopoulou, “A Classifier to Distinguish Between Cypriot Greek and Standard Modern Greek”, University of Nicosia, 2018

[4] Erofili Psaltaki & Dana Roemling, “Drawing on Research on Explainability of Dialect Classifiers to Extract Greek Dialect Features”, University Of Birmingham, June 2024

[5] Steven T. Piantadosi, Zipf’s word frequency law in natural language: A critical review and future directions, Psychon Bull Rev, Mar 2014

[6] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig, “Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing”, ACM Comput. Surv. 55, 9, Article 195 (January 2023)

[7] SMG dataset from: https://github.com/mcmaniou/NLP-Greek-Storytelling

[8] H. Z. Sababa, Greek-dialect-classifier, from: https://github.com/hb20007/greek-dialect-classifier

[9] Achilleos, Alexia, Armostis, Spyros, & Socratous, Eleftheria (Eds) (2022), ΑΠΟαποικιοΠΟΙΗΣΗ: Γλωσσοπλάσματα που μηχανές τζαι πλάσματα [Decolonisation: Linguistic creations by machines and humans], Limassol: Ypogeia Skini.

[10] Benyamin Ghojogh, Ali Ghodsi, Attention Mechanism, Transformers, BERT, and GPT: Tutorial and Survey, HAL open science, Jul 2024