NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

INTERDEPARTMENTAL MASTER'S PROGRAM

"LANGUAGE TECHNOLOGY"

THESIS

Key Point Analysis in Greek: A New Dataset and Baselines

Kleopatra P. Karapanagiotou

Supervisor: **Dimitrios Galanis,** Researcher C' (ILSP)

ATHENS

JANUARY 2025

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

"ΓΛΩΣΣΙΚΗ ΤΕΧΝΟΛΟΓΙΑ"

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανάλυση Keypoints στα Ελληνικά: Ένα Νέο Σύνολο Δεδομένων και Βασικές προσεγγίσεις

Κλεοπάτρα Π. Καραπαναγιώτου

Επιβλέπων: **Δημήτριος Γαλάνης,** Ερευνητής Γ' (ΙΕΛ)

ΑΘΗΝΑ

ΙΑΝΟΥΑΡΙΟΣ 2025

THESIS

Key Point Analysis in Greek: A New Dataset and Baselines

Kleopatra P. Karapanagiotou
A.M.: LT12200010

SUPERVISOR: **Dimitrios Galanis,** Researcher C' (ILSP)

EXAMINATION COMITTEE:
Aikaterini Gkirtzou, Research Associate (ILSP)
Sokratis Sofianopoulos, Research Associate (ILSP)

January 2025

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανάλυση Keypoints στα Ελληνικά: Ένα Νέο Σύνολο Δεδομένων και Βασικές προσεγγίσεις

Κλεοπάτρα Π. Καραπαναγιώτου

Α.Μ.: LT12200010

ΕΠΙΒΛΕΠΩΝ: **Δημήτριος Γαλάνης,** Ερευνητής Γ' (ΙΕΛ)

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ:
Αικατερίνη Γκίρτζου, Επιστημονική Συνεργάτιδα (ΙΕΛ)
Σωκράτης Σοφιανόπουλος, Επιστημονικός Συνεργάτης (ΙΕΛ)

Ιανουάριος 2025

ABSTRACT

Identifying key statements in large volumes of opinionated texts that appear daily in social media, and online debates is an essential tool for informed decision making. During the 8th Workshop on Arguments Mining at EMNLP 2021, in an attempt to suggest relevant solutions to this recent problem, the Quantitative Argument Summarization - Key Point Analysis Shared Task was introduced. The task is divided into Key Point Generation (KPG), which focuses on the identification and generation of key statements from a text corpus, and Key Point Matching (KPM), that maps these statements back to arguments of the original corpus. This subtask combination contributes to a quantitative and explainable solution in the field of multi-document argument summarization, which has been extensively studied in the English language, however, the current landscape lacks research in a multilingual setting. This thesis project is an attempt to adjust the task of Key Point Analysis (KPA) in Greek, a low-resource language. We propose baseline solutions for both subtasks by leveraging available state-of-the-art Greek Language Models with a focus on the recently introduced decoder-only Greek model, Meltemi, to explore both its NLU and NLG capabilities. For both subtasks we use the official dataset of the KPA shared Task, which we adjusted for the Greek language through machine and human translation. For KPM a 4-bit quantized Meltemi-base model is finetuned for classification using PEFT methods and compared to two encoder-only baselines. For KPG we experiment with clustering based abstractive baselines in combination with encoder-decoder and decoder-only models (foundation and instruction-tuned) in zero and few-shot inference settings. The findings show the performance of Meltemi-base v1.0 in the KPM classification task **(avg mAP: 89.06)** comparatively better than Greek encoder-only based classifiers **(avg mAP: 82.01)** as well as that of Meltemi-Instruct v1.5 (**R_1: 20.2, R_2: 8.0, R_L: 19.1, BERTScore P: 74.0, R: 72.8, F1: 73.4)** that outperforms Greek T5 models (**R_1: 12.3, R_2: 3.6, R_L: 11.0, BERTScore P: 66.0, R: 67.5, F1: 66.7)** in KPG. The proposed approaches provide a promising methodology for extending the KPA task in a multilingual setting.

SUBJECT AREA: Text Summarization

KEYWORDS: multi-document, quantitative argument summarization, text classification, clustering methods, abstractive text generation

ΠΕΡΙΛΗΨΗ

Ο εντοπισμός των βασικών θέσεων μέσα σε ένα μεγάλο όγκο ιδεολογικά χρωματισμένων κειμένων, που παρουσιάζονται καθημερινά στα κοινωνικά δίκτυα και τις διαδικτυακές συζητήσεις, αποτελεί ένα απαραίτητο εργαλείο για την συνειδητή λήψη αποφάσεων. Στη διάρκεια του 8ου Εργαστηρίου πάνω στην Εξόρυξη Επιχειρημάτων στο Συνέδριο EMNLP το 2021, σε μια προσπάθεια να προταθούν συναφείς λύσεις σε αυτό το καινούριο πρόβλημα, παρουσιάστηκε ένα έργο προς κοινή επίλυση με τίτλο «Quantitative Argument Summarization - Key Point Analysis». Το έργο χωρίζεται στην Παραγωγή Keypoints (ΠΚ), που ασχολείται με τον εντοπισμό και την παραγωγή δηλώσεων-κλειδιά από ένα σώμα κειμένων, και την Αντιστοίχιση Keypoints (ΑΚ), που αντιστοιχεί αυτές τις δηλώσεις πίσω σε επιχειρήματα του αρχικού σώματος κειμένων. Αυτός ο συνδυασμός υπο-εργασιών προτείνει μια ποσοτική και επεξηγήσιμη λύση στον τομέα της πολυκειμενικής περίληψης επιχειρημάτων, η οποία έχει διερευνηθεί αρκετά στην Αγγλική γλώσσα, ωστόσο το σημερινό τοπίο στερείται έρευνας σε ένα πολυγλωσσικό περιβάλλον. Η παρούσα διπλωματική εργασία αποτελεί μια προσπάθεια προσαρμογής του έργου της Ανάλυσης Keypoints στην Ελληνική, μια γλώσσα με χαμηλό επίπεδο πόρων. Προτείνουμε βασικές λύσεις για κάθε υπο-εργασία, αξιοποιώντας τα πιο σύγχρονα διαθέσιμα ελληνικά γλωσσικά μοντέλα, εστιάζοντας στο πρόσφατο μεγάλο γλωσσικό μοντέλο με αρχιτεκτονική decoder-only, το Meltemi, σε μια προσπάθεια να εξερευνήσουμε τις δυνατότητές του στην Κατανόηση και την Παραγωγή Κειμένου. Σε κάθε υπο-εργασία χρησιμοποιούμε το επίσημο σύνολο δεδομένων του έργου, το οποίο μεταφράσαμε στα Ελληνικά με μεθόδους μηχανικής μετάφρασης και με ανθρώπινη παρέμβαση. Για την ΑΚ χρησιμοποιήθηκε το θεμελιώδες μοντέλο, κβαντοποιημένο σε 4 bits, και εκπαιδευμένο με Parameter Efficient Fine Tuning (PEFT) μεθόδους για κειμενική ταξινόμηση, ενώ το συγκρίνουμε με δύο υπάρχουσες υλοποιήσεις με encoder-only μοντέλα. Για την ΠΚ πειραματιζόμαστε με μεθόδους abstractive παραγωγής κειμένου, βασισμένες σε μεθόδους συσταδοποίησης, με μοντέλα encoder-decoder και decoder-only (θεμελιώδη και instruction-tuned) σε 0-shot και few-shot πειράματα. Τα ευρήματά μας δείχνουν την εξέχουσα απόδοση του Meltemi-base-v1.0 στην ΑΚ ως έργο κειμενικής ταξινόμησης **(avg mAP: 89.06)** σε σχέση με encoder-only μοντέλα **(avg mAP: 82.01)** που έχουν εκπαιδευτεί για τον ίδιο σκοπό, καθώς και την εξέχουσα απόδοση του Meltemi-Instruct-v1.5 (**R_1: 20.2, R_2: 8.0, R_L: 19.1, BERTScore P: 74.0, R: 72.8, F1: 73.4)**, που ξεπερνάει μοντέλα της σειράς GreekT5 στην abstractive ΠΚ (**R_1: 12.3, R_2: 3.6, R_L: 11.0, BERTScore P: 66.0, R: 67.5, F1: 66.7)**. Οι προτεινόμενες προσεγγίσεις παρέχουν μια πολλά υποσχόμενη μεθοδολογία για την επέκταση έργου της Ανάλυσης Keypoints σε ένα πολύγλωσσο περιβάλλον.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Περίληψη κειμένου

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: πολυκειμενική, ποσοτική περίληψη επιχειρημάτων, ταξινόμηση κειμένου, μέθοδοι συσταδοποίησης, αφηρημένη παραγωγή κειμένου

CONTENTS

LIST OF FIGURES

LIST OF TABLES

PREFACE

This research study is a thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in “Language Technology” in the Department of Informatics and Telecommunications of National and Kapodistrian University of Athens (NKUA).

1. INTRODUCTION

There are many controversial topics that concern individuals, companies, and governments daily. Whether it comes to disagreement with a current societal situation, a change in the legislation or any other issue, people need to be able to summarize and review the arguments that support or refute the claims/opinions of all sides. There have been significant steps towards addressing this need through the development of appropriate datasets and novel methods. For example, a few years ago, Haim et al. [1] introduced a novel argument summarization framework that maps large amounts of argumentative sentences into a short bullet-like list of key points, where each of them is ranked by its prevalence based on the number of matched arguments. The so called 'Key Point Analysis' (KPA) task introduced a new quantitative approach that was missing from the argument summarization research landscape. Also, in view of the benefits of a full automation of the whole task [2] and in an attempt to trigger the interest of the NLP research community, the KPA-2021 Shared Task [3] was introduced. The organizers split the task into two complementary, but independent subtasks. In the first, named **Key Point Matching**, participants had to create a system that would give matching scores for each (argument, key point) pair belonging to the same debatable topic and stance (positive/negative). Achieving high scores in the matching system was a prerequisite for participation on the second subtask, namely **Key Point Generation**, which required the generation of a list of key points for each topic and stance and the subsequent prediction of matching scores between arguments and new generated key points. The KPA-2021-Shared Task and relevant dataset enabled experimentation and important advances in the field.

1.1 Aim of the study

Considering KPA systems as valuable tools for informed decision making that should not be limited to English-speaking populations, the aim of this thesis is to develop a Greek version of the KPA task by providing a) appropriate datasets and b) baseline solutions for both of its subtasks. For the latter we aim to leverage existing state-of-the-art language models for Greek [4], [5], [6].

1.2 Related work

Several methods have been proposed for the task of argument extraction [7]. Each of these methods may result in thousands of arguments per topic, making it impossible for a human to digest, review and summarize them. For that reason, several approaches used clustering techniques to group similar arguments. For example, Misra et al. [8] use the notion of 'argument facets' for a set of similar sentences that discuss a particular aspect of an argument and train a regression model that predicts Argument Facet Similarity of two sentential arguments, i.e., whether they are different paraphrases of the same facet. Similarly, Ajjour et al. [9] identify 'frames' of similar arguments using clustering, where a 'frame' refers to arguments that cover the same aspect of a topic. They use the following procedure, in the first step TF-IDF or LSA [10] is used for mapping each argument into a vector space; then in the second step they remove topic-specific structural and lexical features of the arguments and cluster the resulting "topic-free" arguments. Reimers et al. [11] experiment with supervised and unsupervised

methods for argument classification (identification) and clustering and introduce the use of contextual BERT [12] and ELMo [13] embeddings for solving the tasks. They achieved significant improvement over state-of-the-art methods, e.g., 20.8 percentage points for UKP Sentential Argument Mining Corpus and 12.3 for the Argument Facet Similarity (AFS) Corpus. The common limitation of all the above methods is that they did not attempt to create the actual summary of the arguments which have to be compiled manually by analyzing the contents of the clusters.

After 2020, a significant part of research in the field focused on the KPA subtasks and datasets [2], [3]. Below we present these datasets, the respective evaluation measures, and the most important KPA approaches.

1.2.1 ArgKP Datasets

KPA was introduced with a series of benchmarks in the domain of Argumentation. The first large-scale dataset for the task, **ArgKP** [1] consists of 24,093 (argument, key point) pairs along with binary labels, indicating whether an argument is matched to the respective key point. These pairs cover a total of 28 controversial non-domain specific, general topics along with a stance attribute, positive or negative, towards a topic. The arguments in ArgKP are a subset of the IBM-ArgQ-Rank-30kArgs data set [14], consisting of 30K crowd-sourced arguments labeled for their stance and quality towards a specific topic. Argument filtering based on specific polarity and quality score thresholds helped to ensure strong argumentative content with clear argument stances. The key points for each topic were authored by an expert debater based on specific instructions, while binary labeling of each (argument, key point) pair was performed via crowdsourcing.

One year later, an extension of the ArgKP dataset was released in the context of the KPA-2021 Shared Task [3]. The **ArgKP-2021** dataset consists of the existing ArgKP dataset for training and validation sets and 3 new debatable topics as a test set, amounting to 31 topics overall. From the given arguments, 4.7% are unmatched, 67.5% belong to a single key point, and 5.0% belong to multiple key points. The remaining 22.8% of the arguments have ambiguous labels, meaning that the annotators could not agree on a correct matching to the key points. The final dataset contains 27.519 (argument, key point) pairs, of which 20.7% are labeled as matching, indicating a strong dataset imbalance. For a more thorough overview of the ArgKP-2021 dataset, see Table 1.

Table 1: ArgKP-2021 Dataset statistics

Statistics	Train	Validation	Test
Num. topics	24	4	3
Num. arguments	5583	932	723
Num. key points	207	36	33

Num. <arg., kp., label> pairs	20635	3458	3426
Num. <arg.,kp.> pairs[1]	24454	4211	3923
Num. positive pairs	17%	18%	14%
Num. negative pairs	67%	64%	73%
Num. undecided pairs	14%	18%	13%

The latest work on Argumentation datasets is an updated version of the ArgKP dataset, with 10 additional topics and a total of 7,584 (argument, key point) pairs from the IBM-ArgQ-Rank-30kArgs dataset created for evaluating the scalability of existing KPA systems [15].

KPA has also been explored in various specific domains using the relevant datasets, e.g., legal texts [16], political debates at twitter [17], surveys [2], [15], citizen consultation for COVID-19 policies [18], customer reviews [2], [19], [20], ideological debates [18], [21] and employee feedback [15].

1.2.2 Key Point Matching (KPM)

1.2.2.1 KPM solutions before, during and after the KPA Shared Task

The task of Matching an Argument to its associated key point(s) that summarize and reflect its content, was first introduced by Haim et al. [1] Their initial unsupervised approach, that used embedding similarity between (argument, key point) pairs based on TF-IDF-weighted vectors, non-contextual GloVe [22] or contextual BERT [12] embedding representations proved insufficient to produce highly accurate matching scores. The obtained F1-scores are by a small amount better compared to random predictions; TF-IDF:0.352, GloVe:0.330, BERT:0.403 vs. 0.203. Also, their experimentation with Transfer Learning from models developed for Natural Language Inference (NLI); i.e., BERT-based models trained on the respective datasets (MNLI, SNLI) achieved better results than the aforementioned approaches, but still are considered insufficient for the task. For example, BERT-LARGE tuned on MNLI reached only an F1-score of 0.526.

The first satisfactory results were obtained with supervised methods that used fine-tuning of small and large versions of BERT-family LMs [12], [23], [24], [25] or XLNet [26]. For example, RoBERTa-large [23] achieved competitive results in terms of F1

[1] Incl. undecided pairs

(0.773) and runtime efficiency trade-off [2]; it was also the prevalent model choice of top-ranked Shared Task participants [27], [28, [29].

The use of LMs was also meaningful for training sentence-transformer models that capture the semantic similarity between matching (argument, key point) pairs, as well as the dissimilarity of unmatching pairs [30]. By modelling both relationships with contrastive loss function and the use of Siamese Neural networks, Alshomary et al. [30] managed to reach the top performance in the KPM subtask.

More recent solutions introduced the use of sequence-to-sequence models to address KPM as a generative task. Samin et al. [31] was one the first works to implement prompt-based learning for the argument-to-key point mapping task. The use of prompt engineering techniques to incorporate (argument, key point) pairs to a chosen prompt template, that is further used with a fine-tuned encoder-decoder model, e.g., BART [32] and T5 [33], gave improved results compared to models that use fine-tuning and do not incorporate prompt engineering. Zhao et al. [34] used GPT-3.5 and GPT-4 as a 0-shot Key Point Relevance Evaluator to identify the mapping of each argument to all possible key points from the reference dataset within the same topic, using a 1 (not relevant) to 5 (highly relevant) scale. Although the GPT-4 model seems to significantly outperform various baselines [2] with an F1-score of 0.862 on the ArgKP-2021 dataset, it is still unclear how they translate the [1-5] ranks to matching/unmatching pairs. Eden et al. [15], in their attempt to address the practical challenges of implementing KPA systems in production with ever-growing datasets, proposed the use of FLAN-T5-XL [35] (2.85B parameters), an instruction-tuned encoder-decoder model, for the Matching task. They performed fine tuning on FLAN-T5-XL with QLoRA [36] for 1 epoch and inference with a natural language instruction that asks the model to generate a yes/no answer, after providing the relevant KPA context. The fine-tuned model performance was comparable to a deberta-v3-large model (304M parameters) on four out of five debate benchmarks, while being nearly 15 times slower (measured on a single A100 GPU), so overall it was not considered beneficial for use in production. Van de Meer et al. [18] prompts Chat-GPT (gpt-3.5-turbo-16k) to perform the whole KPM subtask by predicting matches for a batch of 10 arguments at a time, belonging to the same topic and writing the results in the required file format. The results show that this approach gave poor results compared to the winning Shared Task solution [30] and the official Debater API [37], an online KPA system (see sec. 1.2.3.1), proving that even a 0-shot approach with 175B parameter models, like the GPT-3.5 models is not enough for the KPM subtask.

1.2.2.2 KPM Evaluation measures

Due to the success of fine-tuning classification models on (argument, key point) pairs [1][2], initial KPM solutions were evaluated with typical classification measures like Accuracy, Precision, Recall and F1. In the context of KPM, Accuracy measures the fraction of all (argument, key point) pairs correctly classified, Precision measures the fraction of correctly classified matching (argument, key point pairs out of all pairs classified as matching. Recall measures the proportion of correctly classified matching pairs out of all matching pairs and F1 is the harmonic mean of precision and recall. In the case of KPM, the dataset is highly imbalanced, i.e., the majority class is “non-matching” (see Table 1 above), therefore it is highly prioritized to correctly predict as many true matching pairs as possible, i.e., the goal is to optimize Precision.

The Shared Task organizers use Average Precision (AP) [38] as KPM evaluation metric since, as they report, “it supports evaluating the correlation between a model’s confidence and prediction success” [3]. Specifically, they pair each argument with its best matching key point (randomly chosen in case of a tie) according to the predicted matching probabilities. Within each topic-stance combination, only 50% of the arguments with the highest predicted matching score are kept for evaluation. The task organizers claim that this removal of 50% of the pairs is necessary because a significant portion of the arguments are not matched to any KP, and this would influence mean Average Precision (mAP) negatively [3]. mAP is obtained by macro averaging over all topics and stances so that each topic and stance have the same effect over the final score. The task organizers consider two evaluation settings: strict and relaxed, which are created to account for (argument, key point) pairs in the ArgKP-2021 with undecided labels (i.e., not sufficient agreement between annotators). In the strict setting, undecided pairs are considered as no-match, while in the relaxed setting as match. The score is then calculated based on the given labels and the derived labels from each setting. The evaluation score in general favors matchers that can match a single key point for each argument with high precision. It is however not important if a matcher does predict non-matches with high certainty.

1.2.3 Key Point Generation (KPG)

1.2.3.1 KPG solutions before, during and after the KPA Shared Task

Key Point Generation has been approached both with extractive as well as abstractive methods. For the first category it is common to see terms in literature such as Key Point/Argument Selection or Extraction, as these solutions are based on the idea that a single argument can be representative of a set of arguments. The methods of the second category rely on the generative capabilities of LLMs to produce an abstractive summary for a set of arguments.

Prior to the introduction of the KPA shared task, the first fully automatic KPA solution was a two-step extractive approach by Haim et al. [2], in which, first, high-quality key point candidates are extracted based on specific filtering steps, like sentence length in tokens and argument quality [14]. Then a matching model [1], [2] obtains match scores between each argument and key point candidate as well as between each candidate pair. In the former scores, arguments are matched to their best-matching candidate, provided their score exceeds a given threshold and are ranked based on coverage, i.e., the number of matched arguments. The latter scores are used to avoid redundancy. Each candidate whose matching score with a higher-ranked candidate is above a given threshold, is dropped and its matched arguments are rematched to the remaining candidates. The key points are again sorted based on coverage scores, to reach the final list. This algorithm has been used for IBM’s official KPA service as part of the IBM Project Debater content summarization API and has served as a baseline for many proposed KPA solutions that followed. An extractive approach was also adopted by the best performing Shared task submission [30], that used a graph-based approach based on PageRank, where the nodes of the graph are arguments, and the edges are matching scores between argument pairs. Again, arguments were filtered based on quality and top nodes were selected as predicted key points according to their importance score, given that their similarity score with already selected nodes is below

a threshold, to ensure diversity. One important limitation of the above extractive solutions is that they prioritize the most popular, frequently seen arguments, leading to generated summaries that often over-represent some key points, while not mentioning others. This phenomenon occurs because there is often an imbalance in the number of arguments discussing different aspects of a topic. Additionally, filtering based on argument quality and length in tokens limits the candidate arguments but also affects the potential coverage of most or all key points.

Abstractive Key Point Generation approaches have received less attention in literature than extractive ones. During the shared task, as per Friedman et al. [3], “only one solution provided semantically meaningful results with abstractive methods”. Kapadnis et al. [28] used each argument along with its topic as input, to generate argument paraphrases with a Pegasus language model [39] fine-tuned for that purpose. Candidate key points were compared with expert annotated key points using the ROUGE-1 [40] evaluation metric, and only the top five highest in rank were retained as final key points, for each topic-stance combination. An important limitation of their approach is that it assumes the availability of reference key points, which is not always granted in real-life applications.

Several post Shared task solutions introduced argument clustering as an intermediate step prior to key point generation motivated by the fact that semantically similar arguments should be sharing the same key point. Such approaches address many limitations of past extractive and abstractive solutions that have to do with reference key point availability [28] and over-representation of frequently seen/recurrent arguments [2][30]. For this intermediate step, most solutions employed sentence-transformers to identify similar arguments. This semantic similarity-based clustering approach has been explored both with unsupervised [41][17][21] and supervised methods [41][21]. In the former, no prior fine-tuning was performed on the chosen pretrained sentence embedding model, while in the latter, the sentence embedding model was first fine-tuned on the train set of the ArgKP-2021 dataset for learning to map arguments to their matching key point. This significantly improved clustering performance in terms of Rand Index [42]. Li et al. [41] using argument clustering as a solution to the KPM subtask, further explored the mapping of multiple arguments to multiple key points using the probability scores of each argument belonging to each cluster. In BERTopic these probabilities are calculated based on the distances between the document embeddings, obtained from the BERT language model, and the centroid of each topic cluster. If the probability of each argument with each of the topic/cluster is above a learned threshold, then this argument is matched to this cluster. Another contribution of their work was the idea of iteratively going through the unclustered arguments and mapping them to existing clusters, by computing the cosine similarity between each unclustered argument and each cluster centroid. An argument is assigned to an existing cluster, if the similarity is higher than a given threshold, otherwise a new cluster is created. In both supervised and unsupervised clustering experiments, iterative clustering demonstrated significant improvement in automatic evaluation metrics, validating its importance to the whole KPA task.

Khosravani et al. [21] propose extracting representative arguments as key points, using a model that predicts matches between all possible argument pairs within each cluster and then extract the argument with the highest number of matches. In a different direction, Li et al. [41] and Ehnert et al. [17], treat the task as abstractive text summarization, where the arguments of each cluster are concatenated and fed to an

LM that generates a key point. For that purpose, Li et al. [41] fine-tuned a FLAN-T5-base, an instruction-tuned version of the encoder-decoder T5 model, while Ehnert et al. [17], use PEGASUS-XSUM, a model pre-trained on BBC articles and their respective headline summaries.

The mentioned clustering-based approaches share limitations that later work came to address. As noted by Li et al. [41], clustering arguments based on their semantic similarity, does not fully align with the objectives of the KPA task, as it is possible for non-semantically similar arguments to share a key point, because an argument can correspond to more than one key points. Furthermore, these approaches focus only on the relationships between arguments that share a common key point (intra-cluster arguments) and neglect those that do not (inter-cluster-arguments). Li et al. [43] fine-tune a generative model (FLAN-T5-large) that can simultaneously provide a probability score indicating the presence of a shared key point between a pair of arguments and generate the shared key point. They then present an iterative graph partitioning algorithm, which generates subgraphs, each representing a collection of arguments that share the same or similar key point, out of which a representative key point is selected based on the highest score of shared key point probability. Their approach has outperformed previous state of the art methods in abstractive KPG [41], proving that both intra and inter-cluster relationships between arguments are important for the task.

Coming to most recent abstractive works, Van de Meer et al. [18] use prompting with ChatGPT to perform KPG with open and closed book prompts. In the former, a list of reference arguments, limited by considering the maximum context window, are provided while in the latter, the model is prompted to generate key points for and against a debatable topic based on its parametric memory. The open book approach showcased significant improvements in several metrics, i.e., ROUGE [40], BARTScore [44] and BLEURT [46] for the ArgKP dataset compared to existing extractive baselines [2][30]. However, experimentation with diverse datasets resulted in a different top performer model, therefore implying that relying only on one dataset for exploring new KPG methods might be misleading.

1.2.3.2 KPG Evaluation measures

In the KPG subtask, an ideal summary should be “concise, non-redundant, and cover different aspects of the topic.” To this end, task organizers [3] used human annotators to evaluate the generated summaries (list of key points) based on how redundant a summary is, and to what extent it captures central points of the topic. The evaluation included questions regarding 1) the quality of the set of KPs generated by the submitted model in terms of clarity of stance (the stance towards the topic), coverage (the number of key points that cover points central to the topic) and redundancy (the number of duplicate key points) as well as 2) the model’s ability to correctly match arguments to the generated KPs. A model’s rank in the Generation Track was the average of these two ranks.

Although human evaluation is preferable, it is neither easily scalable nor easily reproducible. For that reason, most of the post Shared Task approaches experimented with various automatic measures, starting from the common n-gram-based ROUGE [40] measure. Li et al. [41] compute for each unique topic-stance combination the ROUGE F1-score for unigrams (R-1), bigrams (R-2) and longest common substrings (R-L) of

each generated key point with each reference, then take the average of them and finally compute the average for all topic-stance combinations to create the final score. Additionally, Li et al. [41] proposed another approach that was further adopted by later works [18] [21] [44]. They relied on semantic similarity measures to identify the best match between generated and reference key points. They call these metrics Soft-Precision (sP) and Soft-Recall (sR). The former finds the reference key point with the highest similarity score against all generated key points and the latter finds the generated key point with the highest similarity score against all reference key points. They have chosen state-of-the-art semantic similarity evaluation methods such as BLEURT [45] and BARTScore [44] as similarity functions to be maximized (fmax). Below we see the formal representation of the chosen metrics, where A, B are the set of candidates and references and n = |A| and m = |B|, respectively. When i iterates over each candidate, j iterates over each reference and selects the pair with the highest score as the reference for that candidate:

$$sP = \frac{1}{n} \times \sum_{\alpha_i \in \mathcal{A}} \max_{\beta_j \in \mathcal{B}} f(\alpha_i, \beta_j)$$

$$sR = \frac{1}{m} \times \sum_{\beta_i \in \mathcal{B}} \max_{\alpha_j \in \mathcal{A}} f(\alpha_i, \beta_j)$$

Figure 1: soft-Precision and soft-Recall formulas

Ehnert et al. [17] compute ROUGE F-1 for unigrams and bigrams along with BERTScore-based Precision, Recall and F-1 between concatenated strings of generated and reference key points. BERTScore [46] is an automatic evaluation metric for text generation that computes a cosine similarity between the corresponding contextual word embeddings for each pairwise token of the generated and reference summaries and is thus able to detect semantic overlap between the reference and the generated summary even when there is no lexical overlap.

Khosravani et al., [21] show that the standard evaluation measures such as ROUGE “are incapable of differentiating between generated key points of different qualities”. For that reason, they contribute two new measures that assess a) the extent to which different aspects of the debatable topics are represented by the predicted key points and b) the redundancy of the generated key points. For the first objective they define coverage by comparing each generated key point with each reference with a simple fine-tuned classifier to produce matching/unmatching pairs. The number of matched pairs out of all compared pairs is calculated as the final coverage score. Redundancy considers generated key points that are equivalent to the same reference key point as duplicates and measures the percentage of duplicates, i.e., number of duplicate key point pairs divided by the number of all possible pairs. Their proposed KPG solutions seem to outperform existing extractive baselines in the two new measures, achieving a wider representation of arguments, since key points are extracted from smaller clusters as well.

1.2.4 Key Point Analysis in Greek

KPA is a task with strong connections to multi-document argument summarization and argument mining, as the main objective of a fully automatic KPA system is to produce a set of unique key points, where each one will summarize a cluster or set of similar arguments. Focusing on the Greek language and the two mentioned related tasks, multi-document summarization was introduced in Greek through the MultiLing2013 Workshop at ACL 2013 and focused on multilingual summarization. The participants of the respective pilot tasks had to develop systems that generate summaries out of document sets. As reported by Giannakopoulos [47], three out of seven overall submitted systems were for Greek, which indicates a strong interest in the specific task and language. In addition, various argument mining related tasks has been explored for the Greek language; identification of argumentative sentences [48], the effect of multitask learning on Argument Mining [49], the use of graph neural networks [50] and the extraction of arguments from online news articles [51] A recent work in Greek that approaches text summarization in the survey domain and is similar to the quantitative argument summarization task that we focus on, is the work of Karousos et al. [52]. Their pipeline includes response collection from online education surveys, sentence preprocessing and cleaning, clustering and ranking, extraction of the most representative (highest in rank) sentence out of each cluster and finally the generation of an abstractive summary report with GPT-4o, based on the representative sentences. Although this recent work has significant similarities with the KPA task, the main difference is the target output in KPA is the generation and demonstration of a set of keypoint-like sentences in a bullet-style format, while in Karousos et al. [52] case the final output is a paragraph-level report.

1.2.5 Greek Language Models

Our aim is to leverage existing Greek open-source LMs for developing strong baselines for the two KPA subtasks. If compared to widely spoken languages such as English, there exist much fewer NLP resources and LMs for the Greek language as noted by Evdaimon et al. [53] and Papantoniou and Tzitzikas [54]. Below we present the most well-known Greek LMs.

The most significant breakthrough, that brought the Greek language in the new era of AI, was the development of GreekBERT [4]; a model of 110 million parameters, pretrained on two tasks, Masked Language modelling and Next Sentence prediction. GreekBERT uses the BERT-BASE-UNCASED architecture [12] and was pre-trained as a monolingual model on 29 GB of Greek text from the Greek Wikipedia, the Greek part of the European Parliament Proceedings Parallel Corpus (Europarl) [55], and the Greek part of OSCAR [56], a clean version of Common Crawl. More recently, GreekBART was introduced by Evdaimon et al. [54]. It is the first pretrained encoder-decoder Greek model based on BART architecture [32] and its size is roughly 181M parameters. The model is pre-trained on a large Greek corpus (87.6 GB), with a vocabulary of 50,000 sub-words. This corpus comprises the same datasets as GreekBERT plus the Greek web corpus dataset [57], that includes diverse Greek text types, as well as formal and informal text, to enhance robustness [58]. Giarelis, Mastrokostas, and Karacapilidis [5] introduced a series of models for abstractive news summarization, trained on the

GreekSum dataset, using the multilingual T5 LMs, which include google/mt5-small [59] as well as google/umt5-small and google/umt5-base [60]. The so-called 'GreekT5' models consist of a mt5-small and a umt5-small version with 300M parameters each as well as a umt5-base with 580M parameters.

Many language models today (GPT-4, Mistral, Llama etc.) are decoder-only since they are designed to perform text generation tasks. Such models autocomplete a given sequence by iteratively predicting the most probable next word and the representation computed for a given token in this architecture depends only on the left context. This is often called causal or autoregressive attention. Meltemi is the first open-source decoder-only LLM for the Greek language [6]. It is built on top of Mistral-7B [61] and has been trained on a corpus of 43.3 billion monolingual Greek tokens, constructed from publicly available resources. As noted by Voukoutis [62] "Meltemi is developed as a bilingual model using state-of-the-art techniques, maintaining its capabilities for the English language, while being extended to understand and generate fluent text in Modern Greek"[2].

1.3 Contribution of the study

While there have been attempts to approach KPA-related tasks like Argument Mining in a multilingual setting [63] [64], to the best of our knowledge, KPA itself has not been approached in a non-English language. Furthermore, by reviewing the existing KPM and KPG methods, we reached to the conclusion that the use of large decoder-only models (GPT-3.5, -4) has been limited to 0-shot prompting [18], [34], [53], while no experimentation with finetuning or few-shot learning has been reported. With these in mind, our main contributions are highlighted below:

1. We release ArgKP-2021-GR[3] for Greek the first non-English version of the ArgKP-2021 dataset, the official dataset for the KPA task. It was developed with machine and human translation.
2. We propose three KPM baselines with existing Greek state-of-the-art models [4], [6] that reach results comparable to the English equivalent models.
3. We propose four abstractive KPG baselines with existing encoder-decoder and decoder only models [5], [6], developed with zero- and few-shot inference.
4. We identify and point out deficits in Greek resources that will benefit future KPA implementations as well as future Greek NLP applications.

The developed code was made available on GitHub[4].

---

[2] https://medium.com/institute-for-language-and-speech-processing/meltemi-a-large-language-model-for-greek-9f5ef1d4a10f

[3] https://huggingface.co/datasets/Kleo/ArgKP_2021_GR

[4] https://github.com/Kleo-Karap/KPA_thesis

2. PROPOSED KPM METHODS AND EXPERIMENTS

Considering the abundance of proposed KPM solutions submitted for the ArgKP-2021 shared task [3] a meaningful starting point for developing baselines was to replicate two of the top-ranked KPM solutions and adjust them for the Greek language based on available deep learning models. Towards this direction and in view of the recent advancements in Greek NLP, we introduce our baseline method based on Meltemi-base, a decoder-only model for Greek. In the following sections we describe challenges in terms of dataset translation, baseline re-implementation and decoder-based finetuning experiments. All KPM-related experiments were conducted on the GPU infrastructure provided by Kaggle [65].

2.1 Dataset translation

To align with previous work, the dataset used for finetuning, validation and inference was the official Shared task dataset, ArgKP-2021, which was adjusted in the Greek language via machine and human translation. Due to the high number of (argument, key point) pairs (20635), the train set was translated through automatic zero-shot translation with MADLAD400-3B-mt [66], a multilingual, 32-layer, 3 billion parameter model that excels in machine translation and multilingual NLP tasks. It was trained on 1 trillion tokens of publicly available data, making it able to handle over 400 languages. Specifically for the Greek-English language pair, it has been shown competitive with models of the same family that are significantly larger (MT-7.2B, MT-10.7B). We opted for the smaller version due to the limited available computational resources. We manually assessed the quality of the resulting train set by sampling a subset of it and in most cases satisfying translations have been generated. The validation and test sets, consisting of 3458 and 3426 respectively, were human translated by the author of this thesis.

2.2 Adjustment of existing KPM baselines

The first KPM solution that we used as a basis was the one from Alshomay et al. [30] named ‘SMatchToPR’. It trains a sentence-transformer model for bringing closer in an embedding space matching (argument, key point) pair while setting apart the non-matching ones. The other solution that we used is the one from Kapadnis et al. [28] named ‘Enigma’. It is a typical classifier, giving similarity/matching scores for (argument, key point) pairs. Both solutions leverage encoder-only models for their implementation. More specifically, in SMatchToPR they embed each argument and key point-topic concatenation separately with a RoBERTa-large model and then feed them to a Siamese Neural network architecture for training with a contrastive loss function, achieving a strict mAP score of 0.789 and relaxed mAP 0.927 on the English test set of ArgKP-2021. In Enigma they finetune a DeBERTa-large model on concatenations of argument-key point-topic triplets and then concatenate their outputs with their encoded POS tag[5] [67] representations to further feed them to two dense layers to get the final

[5] https://spacy.io/usage/linguistic-features#pos-tagging

matching score out of a sigmoid activation function, achieving a strict mAP score of 0.739 and relaxed mAP 0.928 on the English test set of ArgKP-2021.

2.2.1 Re-implementation Experiments and Evaluation

To be able to work with available Greek models and resources and make more direct comparisons, we had to adjust the existing implementations[6],[7] with models that are available both in English and Greek. We therefore considered BERT (bert-base-uncased) and its Greek version GreekBERT (bert-base-greek-uncased-v1) for replicating SMatchToPR and Enigma, as it has reported exceptional performance in various NLU tasks, including text classification and semantic similarity. For both solutions we use the existing publicly available implementations with the same hyperparameters (which can be found in Appendix I: Table 12 and Table 13) and change only the model and dataset. For replicating Enigma, no additional POS tag features are used to align with the input in the SMatchToPR implementation. An additional significant difference is that for SMatchToPR Alshomary et al. [30] use the ~20k argument-key point pairs of the ArgKP-2021 train set for fine-tuning, while for Engima, Kapadnis et al. [28] use both the train and the validation set (~24k argument-key point pairs) during fine-tuning. We kept the same setting to make comparisons across the two languages (see Table 2 below). We present the evaluation results of an initial comparative view of the KPM task in the two languages on the original (for English) and the human translated (for Greek) test set of the ArgKP-2021 dataset, including the undecided pairs, with the official Shared task evaluation metrics and scripts. For English we do not obtain the same results with the ones reported in the respective papers. This is expected because different models are used. However, that allows us to compare the approaches on an equal footing, to the extent possible.

Table 2: Re-implementation Results on ArgKP-2021-GR test set

Model	Measures					
	EN (BERT)			GR (GreekBERT)		
	mAP strict	mAP relaxed	Avg mAP	mAP strict	mAP relaxed	Avg mAP
Enigma replication	79.92	91.70	85.81	78.96	85.07	82.01
SMatchtoPR replication	81.83	88.16	85.00	80.15	90.10	85.12

[6] https://github.com/webis-de/argmining-21-keypoint-analysis-sharedtask-code

[7] https://github.com/manavkapadnis/Enigma_ArgMining/tree/main

Reading each re-implementation separately, we observe that:

1. The Greek version of Enigma is giving less correctly classified matching argument-key point pairs, compared to its English equivalent. We conjecture that this is due to the machine translated data which were used for the model's finetuning, as well as the small number of training epochs (3).
2. Interestingly, the Greek version of SMatchToPR reimplementation is not showing the same decrease on identifying matching pairs. The model gives on average equal performance (avg. mAP) in both languages, which is an indication that fine tuning a GreekBERT-based sentence representation model on machine translated data is more efficient and robust in capturing semantic similarity between argument key point pairs, than training a custom classifier model on top of BERT. Furthermore, the number of training epochs (10) seem to have played a significant role towards stabilizing the embedding model's performance.

The main takeaway of the above preliminary comparison is that a) the KPM subtask is of equal difficulty in both English and Greek language b) using the machine translated data for fine-tuning is a viable approach that achieves competitive scores. Both aforementioned facts show that it is worth exploring new baselines with the existing machine translated train set and draw conclusions from our human translated development and test set.

2.3 Proposed decoder-based KPM classifier

Taking into consideration that a) initial implementations [28] [30] have mainly approached KPM as NLU tasks (text classification , sentence-level semantic-similarity), where encoder-only models traditionally prevail, b) the recently reported NLU capabilities of decoder-only models like Llama and Mistral in text classification tasks [68] and c) the recent advancements in Greek NLP, with the introduction of the first Greek decoder-only model, Meltemi-7B [6], our initial approach was to replicate the two baselines mentioned in the previous section and re-implement them with Meltemi-base (v1.0). The next version of Meltemi (v1.5) was made available after the completion of our KPM experiments and was used only for KPG; see section 3. A re-implementation of SMatchToPR (the best-performing method), would require obtaining embeddings from the input text; to the best of knowledge such models, dedicated for Greek are not available. Although recent research has shown that decoder-based models (like Meltemi) can be used for efficiently obtaining text embeddings [69], that has not yet been done and evaluated in Greek. Our implementation is conceptually closer to the 'Enigma' solution, as we train our model on concatenations of <argument+key point> pairs or <argument+key point+topic> triplets and output a matching score for each one of them, which is then used for evaluation with the official KPM evaluation framework.

2.3.1 Meltemi Classification Finetuning Experiments and Evaluation

To develop our approach, we use the split as provided by the task organizers with 24 topics for training, 4 topics for validation, and 3 topics for testing. We take into consideration the dataset's high class imbalance and add weights to the classes during training. The model is loaded with its classification head through the Transformers

library, meaning it outputs two class labels, out of which the matching score is extracted. We consider as scores the probabilities of the class_1, meaning the probability of a key point to match the argument. To load a 7-billion-parameter model, like Meltemi, in a machine with limited computational power we use Quantization techniques, for reducing the memory usage and speeding up model execution. The model was loaded with 4-bit quantization, where each weight of the model was represented using only 4 bits, as opposed to the typical 32 bits used in single-precision floating-point format (float32). To fine-tune Meltemi in our specific classification task, we apply a technique named LoRa [70] that freezes the pretrained model weights and further adds lightweight trainable rank decomposition matrices into each layer of the Transformer architecture. LoRa belongs to the family of PEFT (Parameter-Efficient Fine Tuning) methods that are designed for efficient adaptation of large pretrained models to various downstream applications, by reducing the number of training parameters, making the model more efficient in terms of memory and storage usage, while still achieving (in many cases) performance comparable to full fine tuning. To fine-tune a model using LoRa, the task type, the dimension of the low-rank matrices (LoRA r), the scaling factor for the weight matrices (LoRA alpha), and the dropout probability of the LoRA layers (LoRA dropout) as well as the LoRA bias to train all bias parameters had to be defined. For a thorough inspection of the model's hyperparameters refer to Appendix I-Table 14.

We plot the learning curves of the fine-tuning experiments (Image_1, Image_2) on the Greek ArgKP-2021 validation set for every 125 training steps and report the results after one epoch finetuning in Table_3.

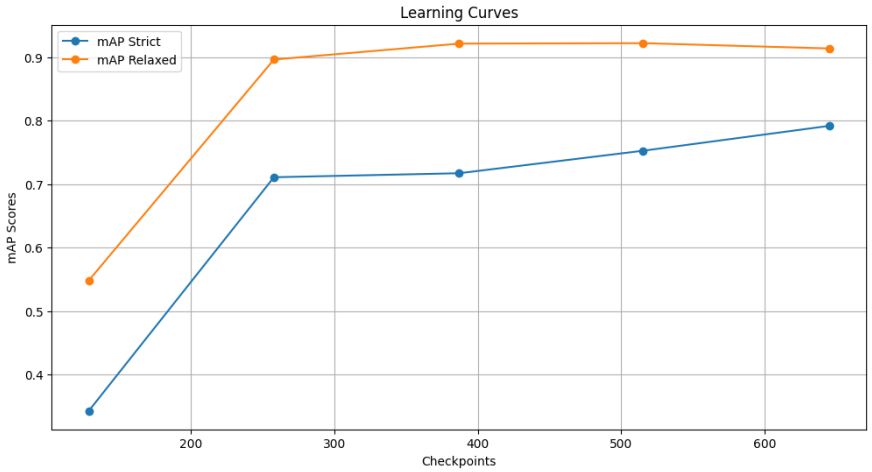

Figure 2: Meltemi KPM Experiment_1

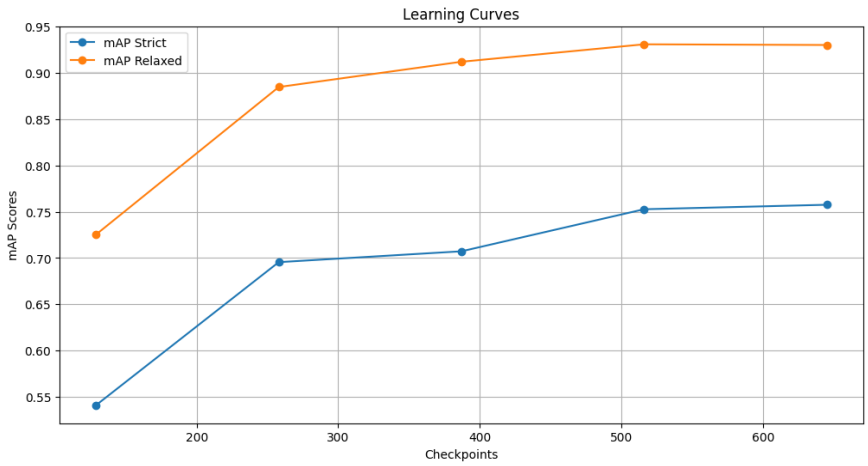

Figure 3: Meltemi KPM Experiment_2

Table 3: Meltemi KPM Experiments on ArgKP-2021-GR dev set

	mAP(strict)	mAP(relaxed)	Avg mAP
Experiment_1	79.18	91.37	85.27
Experiment_2	75.74	93.00	84.37

Experiment_1: The model taking as input concatenated <argument+key point> pairs seem to stabilize its mAP relaxed performance after 300 training steps above 90.00, while its strict performance seems to steadily increase above 70.00 reaching a mAP of around 80.00 at the end of one epoch. The increasing tendency of strict mAP score is a positive indication that further finetuning will increase the model's avgmAP score.

Experiment_2: We experiment with the addition of topic as additional context to the model's input. After 500 training steps, the model's performance seems to be stabilizing around 93.00 mAP relaxed and 75.00 mAP strict, showing that the addition of topic to the model's input does not contribute to the model's performance, i.e., mAP strict has been stabilized and decreased at around 5 points less than the model in Experiment_1.

We recognize the model trained on pairs of arguments and key points (Experiment_1) as the best performing model, but we keep both versions for inference on the test set to make the required comparisons with previously explored baselines.

2.4 KPM Results

Below we present the results of our baselines on the test set of the Greek version of ArgKP-2021**.**

Table 4: Final KPM results on ArgKP-2021-GR test set

Experiments	Num. training instances	mAP (strict)	mAP (relaxed)	Avg mAP
SmatchtoPR (arg,topic+kp)-GreekBERT	20.635	80.15	90.10	85.12
Enigma (kp,arg,topic)-GreekBERT	24.093	78.96	85.07	82.01
Meltemi-base (kp, arg)	20.635	83.86	94.27	89.06
Meltemi-base (kp,arg,topic)	20.635	81.78	93.68	87.73

The main observations from this comparison are:

1. Shared task submission results were verified in our experimentation, as 'SMatchToPR' solution continues to achieve higher mAP scores compared to the 'Enigma' solution in the Greek language.
2. Our findings from experimentation on the Greek ArgKP-2021 development set were verified also on the test set, with the Meltemi model fine-tuned only on (argument, key point) pairs, without the addition of 'topic' context giving higher mAP scores.
3. Comparing our decoder-based classifier with <kp,arg,topic> input with the Enigma classifier, that has the same input, we observe that a decoder-based 7B model, fine-tuned on 20k train set with QLoRa for classification for 1 epoch, correctly identifies more matching argument-key point pairs, than a fully fine-tuned 110M parameter encoder-only model trained for 3 epochs and with 24k train set. This behavior is somewhat expected due to the significant difference in the number of model parameters. In terms of training time Meltemi-base fine-tuned with 5% trainable parameters of its original size (which is about 3.5 million parameters) took ~20 hours, on the other hand Enigma required less than an hour.

Based on the above we conclude that decoder-only models are very competitive in KPA-related classification tasks compared to traditional encoder-only models for the Greek language.

3. PROPOSED KPG METHODS AND EXPERIMENTS

To create our KPG baselines, we adopt a two-step clustering-based abstractive KPG solution that is similar to Ehnert et al. [17] method; see Section 1.2.2.1 for more details. Our approach uses in the first step a customized unsupervised version of BERTopic, along with hyperparameter tuning for argument clustering. In the second step a Greek LM is used for generating a key point from each cluster. Our approach differs though, in that instead of performing the argument clustering and the key point generation completely separately, we use the representation tuning module of BERTopic, which enables us to cluster the arguments and extract a topic representation that we return as a key point for each cluster. For generating key points, we explored the capabilities of recent Greek encoder-decoder and decoder-only LLMs in zero and few shot settings with prompt engineering. More details on BERTopic hyperparameter tuning, representation tuning and prompt engineering are provided in the following sections. All KPG-related experiments were conducted on the GPU infrastructure provided by Kaggle [65].

3.1 BERTopic hyperparameter tuning

BERTopic is a modular topic modeling technique that leverages the bidirectionality of Transformers [71] for document embeddings, clustering techniques and the concept of c-TF-IDF (class-based TF-IDF) for selection of the most representative documents to create easily interpretable topics. In our pipeline we generate sentence embeddings of arguments using a pretrained sentence transformer model [72]. Following previous work for the English language [41] [17] [21], we used the paraphrase-multilingual-mpnet-base-v2 for the embeddings, a multilingual model trained with knowledge distillation [73], a machine learning technique that aims to transfer the learnings of a large, trained model, the “teacher model”, to a smaller “student” model. In our case paraphrase-mpnet-base-v2 was used as a teacher and xlm-roberta-base as a student model. Simple text preprocessing steps were applied to avoid harming the quality of embeddings, e.g., removal of common social media text, retweets, usernames @, emojis, URLs, extra whitespaces. We use UMAP for dimensionality reduction of the embeddings, and HDBSCAN for argument clustering as suggested by Grootendorst [74]. UMAP allows greater control over the distance between the generated low-dimensional representatives than alternative dimension reduction methods such as t-SNE [75] and has a positive effect on the computation time [76] and the quality [77] of the HDBSCAN clustering procedure. Furthermore, it tends to keep the dataset’s global and local structure even when reducing dimensionality. HDBSCAN groups together points that are closely packed together while marking points in low-density regions as outliers/noise [78]. As noted by Ehnert et al. [17], “this has the advantage that only similar statements with a sufficient density concentration are considered for further key point summarization and it reduces the influence of the data noise, by excluding arguments that cannot be easily classified”.

3.1.1 Clustering Experiments and Evaluation

In our experiments, a total of four hyperparameters were optimized, two in the chosen dimensionality reduction technique (number of neighbors and number of target dimensions) and two in the clustering algorithm (minimum samples and cluster selection method). To map relationships between data points in the original UMAP data space in a likelihood graph, we determined the optimal number of neighbors between the following 2 sets of 4 values [3, 5, 10, 15] (Experiment_2) and [10, 20, 30, 50] (Experiment_3), with smaller-set values giving a local view of the data and bigger-set values a more global view. The optimal number of target dimensions for reducing the number of features for each embedded argument was searched between the values [2,5,7]. To optimize the HDBSCAN clustering hyperparameters we start by setting the minimum cluster size to 3, after manual inspection of the train and dev set and control the number of outliers generated, by tuning the minimum samples value. Setting this value significantly lower than minimum cluster size helps to reduce the amount of noise. However, forcing the model to not have outliers may not properly represent the data. We find the min_samples value through the product of the minimum cluster size and a fraction by choosing between the values [0.5 and 1.0], meaning we consider half of the minimum cluster size or set the minimum cluster size and min samples to equal value. Although previous work performs automatic topic reduction, to ensure a fully unsupervised topic modelling process, we specify the topic_reduction parameter to 10, so that a maximum of 10 key points are generated for each topic and stance combination. The choice of this number was made based on the Shared Task's instructions that required a minimum of 5 and a maximum of 10 generated key points for each topic. Lastly, we optimized the cluster selection method by choosing between the default 'eom' (Excess of Mass) and 'leaf', as the latter has been recommended for producing more "homogeneous" clusters[8] [79].

The hyperparameter tuning was performed through the Optuna optimization framework [80], which enables to iteratively model the behavior of an objective function and guide the search for optimal hyperparameter values. To align with the Bayesian optimization-based sampling method used in previous works [17], we chose the TPE (Tree-structured Parzen Estimator) sampler. During Optuna optimization, the quality of the topic model has to be determined by an evaluation metric. Density-based clustering algorithms such as HDBSCAN, pose specific requirements for a target metric [81]. Evaluation metrics for unsupervised clustering such as the silhouette coefficient [82] the Calinski-Harabasz index [83] or the Davies-Bouldin Index [84] share the limitation that they do not consider the density of clusters and are sensitive to noise and outliers. Given these characteristics, we opted for the density-based cluster validity (DBCV) index [85], which lies in the interval of [−1, 1], with higher values indicating better clustering, and considers the influence of noise by considering all data points in the evaluation of the global cluster validity, which is intrinsic to the definition of the density-based clustering.

In each experiment the hyperparameter optimization is performed for each topic and stance combination, meaning that we receive multiple clustering results, out of which an

[8] Parameter Selection for HDBSCAN* — hdbscan 0.8.1 documentation

average score is calculated. A total of 100 trials were executed to find the optimal hyperparameter set for each topic and stance combination. In Table 2 we show the best clustering setting based on the chosen internal clustering validation metric. For a more thorough inspection of the values chosen for each topic-stance combination refer to Appendix II- Table 15.

Table 5: Clustering experiments on ArgKP-2021-GR dev set

DBCV									
Cluster setting	Topic1 Stance 1	Topic1 Stance -1	Topic2 Stance 1	Topic2 Stance -1	Topic3 Stance 1	Topic3 Stance -1	Topic4 Stance 1	Topic4 Stance -1	Avg scores
Experiment 1	-0.357	-0.105	-0.347	-0.436	-0.390	-0.176	-0.191	-0.219	-0.278
Experiment 2	0.210	-0.072	-0.005	-0.193	-0.057	0.060	0.053	-0.036	-0.005
Experiment 3	0.143	0.005	-0.008	-0.054	0.055	0.070	0.064	0.053	0.041

Experiment 1: As expected, using the default BERTopic hyperparameters[9] [86] gave on average the lowest results, with all the scores being below 0. In the plot below we demonstrate an example of generated non-representative clusters.

[9] https://maartengr.github.io/BERTopic/getting_started/parameter%20tuning/parametertuning.html

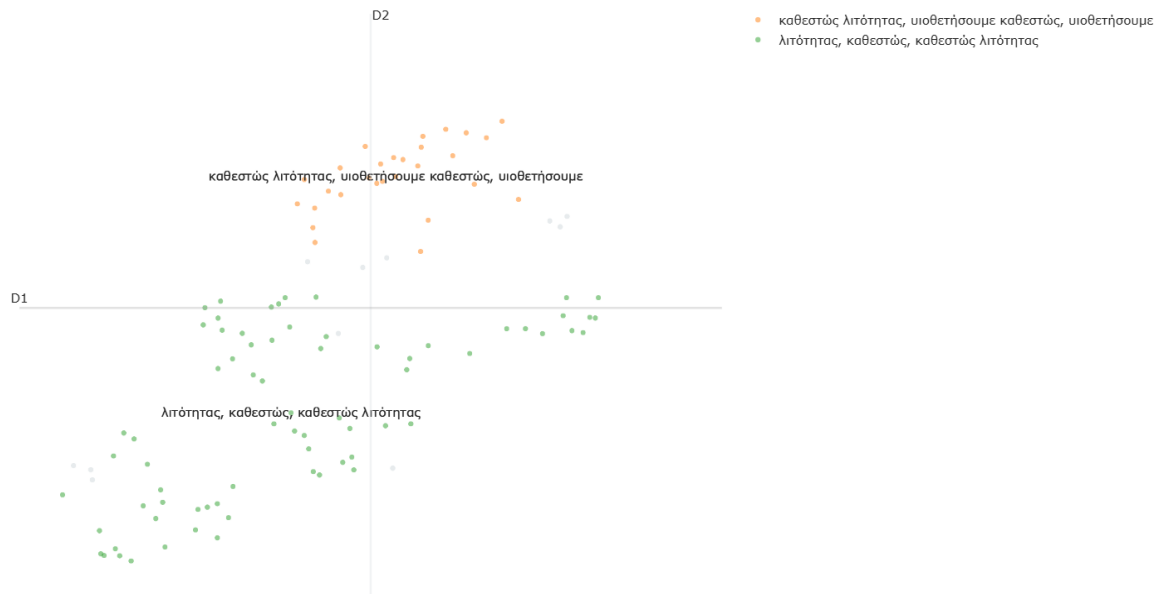

Figure 4: Clustering visualization of default parameters

Experiments 2 & 3: In experiments 2 & 3 we tune the four hyperparameters mentioned above (**number of neighbors, number of target dimensions, minimum samples ratio, cluster selection method)** with the chosen sets of values on the chosen hyperparameter optimization framework. Their difference is in the 'n_neighbors' hyperparameter of UMAP dimensionality reduction technique. Experiment 2 searches the best value among the smaller-set values [3,5,10,15], while Experiment 3 searches among larger sets [10,20,30,50]. As observed from Table_2, Experiment 3 gave on average the highest index scores (0.041), indicating that the n_neignbors parameter with higher than the default value ('15') gives better clustering results for the given hyperparameter setting. We therefore chose our best hyperparameter values from Experiment 3. For a thorough inspection of the final hyperparameters of our BERTopic models refer to Appendix II – Table 16. With this setting, we managed to capture clearer argument clusters, compared to our first plot with the default settings, that also capture semantically different aspects of the arguments, at least with the initial Bag of Words representation.

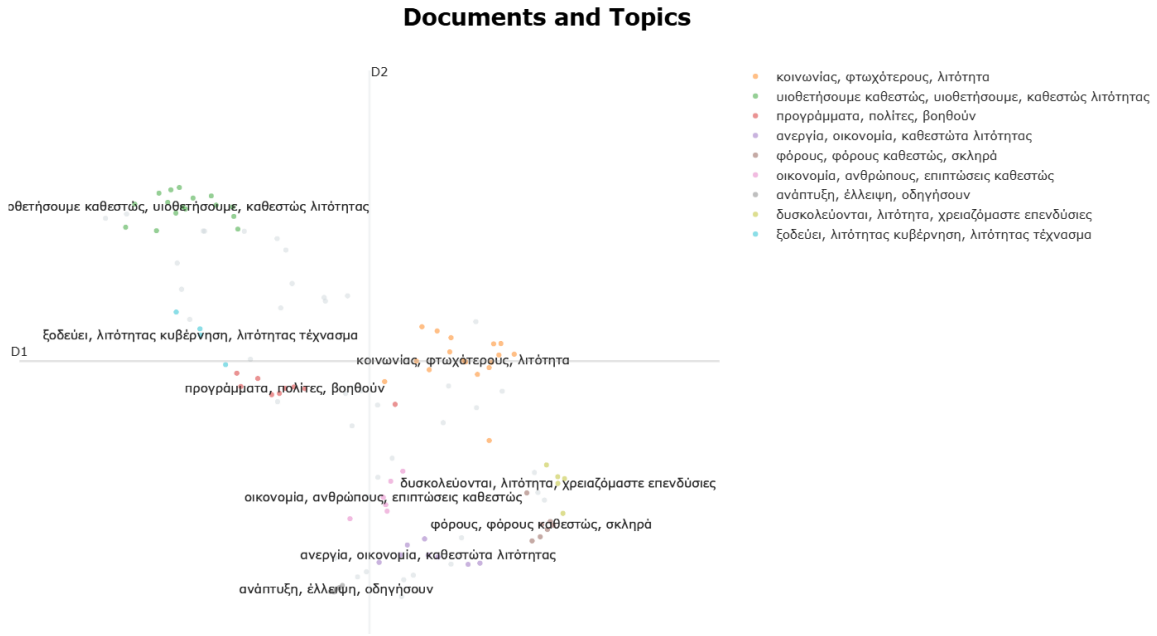

Figure 5: Clustering visualization Experiment_3

3.2 Topic Representation fine-tuning

Representing topics through Bag-of-Word representations and weighting with c-TF-IDF, as in the above plots, can be further fine-tuned and improved through several techniques. Recently, the BERTopic community has realized the value that generative AI can bring, so they enriched the 'Topic Representation' module of BERTopic with the capability to use LLMs for the generation of representative sentences for each cluster[10] [87]. First a set of keywords and documents (in our case arguments) that describe a topic best has to be generated using BERTopic's c-TF-IDF weighting scheme. c-TF-IDF treats all documents in a cluster as a single document and computes TF-IDF, to obtain the importance scores of words within a cluster. By default, the four most representative arguments are passed to the text generation model which is prompted to generate output that fits the topic best. We treated the task of KPG as an abstractive summarization task, to align with previous work [41][17]. For that reason, we explored two existing summarization models, umt5-base (a.k.a. GreekWiki) which is fine tuned for encyclopedic article summarization and umt5-base-greeksum (a.k.a GreekT5) which is fine tuned on Greek news summarization data. See also section 1.2.4 for more information on the models. Based on preliminary 0-shot examples we found umt5-base more suitable for our use case, since the output had a simpler sentence structure (SVO) which is compatible with the succinct and informative structure of a key point. To align with the recent advancements in Greek NLP and Generative AI, we also used Meltemi base and Meltemi Instruct, to explore their generative and instruction-following capabilities in a new, for them, task.

[10] https://maartengr.github.io/BERTopic/getting_started/representation/llm.html

3.2.1 Prompt Engineering Experiments and Evaluation

Prompt engineering was necessary, to give our models a more concrete direction for the target output of each cluster representation. We experimented with zero-shot learning, where only the instruction of the desired task is provided to the model and continued with few-shot learning, providing a few examples of the desired output. Evaluating the quality of generated key points has been a crucial challenge for a fully automatic KPA system that brought a lot of interest in research as we saw in section 1.2.2.2. The lack of a common established KPG evaluation framework, not only placed a significant burden in the comparison of existing solutions in literature but is also a challenge for a non-English language. Considering the resource availability for the Greek language, we opted for the ROUGE implementation of Li et al. [41] through HugginFace's Transformer library[11]. ROUGE was calculated on stemmed tokens based on the Greek stemmer of Ntais [88], after normalization, social media text removal and punctuation removal. Additionally, to overcome the limitations of exact match metrics like ROUGE, that penalize highly abstractive summaries and do not capture semantic similarity between generated and reference texts, we further compute BERTscore[12] [47] Precision, Recall and F1 scores between each pair of reference and predicted key point within the same topic-stance combination, average the pairwise scores to obtain average P, R, F1 scores for each topic-stance combination and then take the average value of all topics to reach the final scores. For BERTscore no text preprocessing was applied on the tokens. In a non-English evaluation setting, the authors of BERTScore [47] recommend the use of mBERT [12]. Finally, to avoid having too semantically similar or near identical key points, we perform deduplication. Considering that the argument clusters are ranked in descending order, based on the number of arguments belonging to a cluster, we drop the produced key points that indicate sentence similarity [47] higher than a specified threshold with a higher in rank key point. All experiments were performed on the ArgKP-2021 dataset, from which the machine translated train set was used to construct the demonstrations of the few-shot experiments and the human translated validation and test sets were used for evaluation.

3.2.1.1 Zero-shot Experiments

We start by evaluating our chosen models; GreekWiki, Meltemi base and Meltemi Instruct in zero-shot experiments by providing a single instruction of the task to perform. Since GreekWiki was fine-tuned for summarization with a task specific prefix ("summarize: <input text>") we started our experimentation with this prefix as a prompt for all 3 models. Additionally, we experimented with two deterministic decoding strategies, greedy and beam search [89], to optimize the quality of the generated text. Greedy decoding takes at each step the token with the highest conditional probability from the vocabulary. In Beam Search, as opposed to greedy decoding, a number of

[11] https://huggingface.co/spaces/evaluate-metric/rouge

[12] https://huggingface.co/spaces/evaluate-metric/bertscore

best candidates, known as beam width (k), are selected, and kept based on some score. The k-best generated sequences continue to expand, until the “end” token (EOS) is reached. Considering the available GPU resources, to ensure a balance of computational time and optimal number of beams we chose for our experiment a beam width of 3[13] [90]. In Table 6 we see the results of the initial comparison between the two decoding strategies on all three models on the human translated ArgKP-2021 dev set.

Table 6: Zero-shot decoding experiments on ArgKP-2021-GR dev set

	GreekWiki			Meltemi-base (v1.5)			Meltemi-Instruct (v1.5)		
0-shot model-specific prompt (greedy)	ROUGE	BERT Score	Avg token count	ROUGE	BERT Score	Avg token count	ROUGE	BERT Score	Avg token count
	1: 14.7 2: 5.3 L:13.9	P: 67.1 R :70.7 F1:68.8	20.1	1: 15.4 2: 5.9 L:14.6	P: 66.3 R: 72.5 F1:69.2	19.41	1: 15.4 2: 6.0 L: 14.6	P: 66.9 R: 73.6 F1:70.0	24.16
0-shot model-specific prompt (beam_3)	1: 12.8 2: 4.8 L:12.5	P: 66.5 R: 70.1 F1:68.2	19.72	1: 15.0 2: 5.0 L:14.3	P: 66.9 R: 72.7 F1:69.6	18.61	1: 15.8 2: 5.3 L: 14.8	P: 66.3 R: 73.6 F1:69.7	24.45

Our initial experiments with Greedy/Beam search decoding gave us the following observations:

1. Overall, we confirmed our initial intuition about the generative capabilities of Meltemi in an unknown task. Both base and instruction-tuned Meltemi versions outperform a significantly smaller summarization fine-tuned model like GreekWiki in ROUGE and BERTScore metrics for both decoding settings. Meltemi-base and Instruct do not have significant differences in their automatic evaluation scores, indicating that a manual evaluation of their outputs is necessary.
2. For GreekWiki, despite the relatively higher scores of Greedy decoding over Beam Search, after a manual comparison of the generated key points, it was concluded that greedy outputs are somewhat fluent, but in many cases semantically incoherent. More specifically, there are a lot of repetitions of words and phrases, numbers, dates or verb alternations and a lot of hallucinations, by producing non-existent terms and entities for the given context. Beam search on

[13] https://datascience.stackexchange.com/questions/126904/how-to-select-the-optimal-beam-size-for-beam-search

the contrary, produced more plausible sentences that retain the definition-style that is commonly found in encyclopedic texts on which GreekWiki was pre trained and overall, its outputs were more comprehensible.

3. Meltemi base seems to perform equally in both decoding settings. Though it seems to generate fluent and comprehensible sentences, by inspecting the generations in more detail, we observe that the model is simply copying verbatim arguments from the given prompt, which results in generating as key point an argument that already exists in the input cluster, as an extractive method would do.
4. Meltemi Instruct does not indicate substantial differences between the 2 decoding strategies. With greedy decoding we see the model repeating some existing arguments as generated key points, while in beam search it generates completely new abstractive sentences. The main drawback in both settings is that the sentences resemble summaries, consisting of many subordinate sentences, whereas a key point should convey the important information in a simple SVO structure. This is also confirmed by the average length of the generated sentences, mentioned in Table 6, compared to the average length of the reference key points (~7.8 tokens). Nevertheless, it was observed that Meltemi-Instruct was the only model of the 3 that produced promising results for our KPG use case and therefore was chosen for additional experimentation with three more elaborate prompts; Prompt_1, Prompt_2 and Prompt_3.

Prompt_1 (see Table 7) was inspired from the simplistic “summarize:” prefix that was used for fine-tuning the GreekWiki model. However, it was designed for the specific KPA task. The second prompt tested (Prompt_2), was more detailed and more specific to the task of Key Point Generation, in order to steer the given model to generate a keypoint-like sentence. The main reason for conditioning the prompt as close to the KPG task as possible was to test the instruction following capabilities of the used model. Prompt_3 follows the logic of Prompt_2 but is even more task-specific and detailed, as we instruct the model to output its answer by following a specific format. We experimented with all 3 prompts (with beam_3) on the Meltemi Instruct model, the best performing model of the previous experiments. Our decision for the most appropriate prompt was based on automatic measures as well as manual inspection of produced outputs.

Table 7: Custom Prompts

	Custom Prompts (GR)	Custom Prompts (EN)
Prompt_1	Γράψε μια σύντομη πρόταση ως περίληψη για το παρακάτω κείμενο: [ARGUMENTS] Περίληψη:	Write a short sentence as a summary of the following text: [ARGUMENTS] Summary:

Prompt_2	Τα παρακάτω επιχειρήματα υποστηρίζουν ή αντικρούουν το θέμα. Συμπλήρωσε ένα συνοπτικό keypoint, καταγράφοντας την κεντρική ιδέα των επιχειρημάτων σε μία πρόταση. [ARGUMENTS]	The following arguments support or refute the topic. Complete a succinct key point, capturing the main idea of the arguments in one sentence. [ARGUMENTS]
Prompt_3	Παρακάτω θα δεις μερικά επιχειρήματα υπέρ ή κατά για ένα συγκεκριμένο θέμα: [ARGUMENTS] Με βάση τα παραπάνω, γράψε μια σύντομη πρόταση που να συνοψίζει αυτά τα επιχειρήματα σε ένα keypoint, ακολουθώντας το μοτίβο: θέμα: <keypoint>	Below you will see some arguments for or against a particular topic: [ARGUMENTS] Based on the above, write a short sentence summarizing these arguments in a key point, following the pattern: topic: <keypoint>.

Table 8: Prompt Engineering results on ArgKP-2021-GR dev set

Custom Prompts	Automatic evaluation measures						
	ROUGE			BERTscore			Avg token count
	1	2	L	P	R	F1	
Prompt_1	14.7	4.9	13.7	65.6	73.4	69.3	26.9
Prompt_2	14.8	5.0	13.8	65.8	73.6	69.4	30.0
Prompt_3	16.9	6.0	15.9	67.6	73.5	70.4	20.96

Prompt_3 (see Table 8) produced the best results on the dev set in terms of automatic measures (ROUGE, BERTScore), while these results were confirmed by a manual inspection of a sample of the outputs. Instructing the model to follow a specific pattern in its answer seems to have helped the model produce significantly shorter, in tokens, outputs, with a lot of nominalizations and succinct format. On the contrary the first two prompts contained subordinate sentences and more complex syntactic structures, making them unsuitable for our KPA use case. Therefore, the third prompt was chosen for our next experiments. For a thorough inspection of prompt templates and their produced outputs see examples in Appendix II-Table 18.

In Table 9 we compare GreekWiki, Meltemi-base and Meltemi-Instruct for the “summarize:” and Prompt_3 templates with the beam search decoding strategy on the dev. set of ArgKP-2021 dataset.

Table 9: Zero-shot prompt experiments on ArgKP-2021-GR dev set

	GreekWiki			Meltemi-base (v1.5)			Meltemi-Instruct (v1.5)		
	ROUGE	BERT Score	Avg token count	ROUGE	BERT Score	Avg token count	ROUGE	BERT Score	Avg token count
0-shot model-specific prompt (beam_3)	1: 12.8 2: 4.8 L:12.5	P: 66.5 R: 70.1 F1: 68.2	19.72	1: 15.0 2: 5.0 L:14.3	P: 66.9 R:72.7 F1:69.6	18.61	1: 15.8 2: 5.3 L:14.8	P: 66.3 R:73.6 F1:69.7	24.45
0-shot Prompt_3 (beam_3)	1: 9.2 2: 2.3 L: 8.5	P: 65.0 R:68.9 F1:66.8	25.78	1: 15.9 2: 5.0 L:14.7	P: 66.5 R: 72.5 F1:69.3	18.68	1: 16.9 2: 6.0 L:15.9	P: 67.6 R: 73.5 F1:70.4	20.96

Our experiments gave us the following observations:

1. It was confirmed that using the GreekWiki model with the “summarize:” prefix it was pre trained with, is crucial for the model to produce comprehensible output. When tested with Prompt_3, it copies a lot of words from the prompt, leading to unmeaningful sentences. ROUGE and BERTScore measures confirm this observation.
2. The outputs of Meltemi base continue having repetitions of arguments from the prompt as extracted key points, leading to non-significant changes in the evaluation measures if compared to GreekWiki. We also observe that Meltemi base is not following the given instructions, which is an expected behavior, since the base version has not been trained on instruction data.
3. Meltemi Instruct achieves a significant increase when compared to Meltemi base and GreekWiki both in terms of automatic and manual evaluation, when tested with Prompt_3, showing the most optimization potential among all three tested models.

For more information on the manual evaluation and comparison of the models see examples in Appendix II-Table 17.

3.2.1.2 Few-shot Experiments

We continued our experiments by providing a few demonstrations on the best performing model of the 0-shot setting, the Meltemi Instruct model with beam search and Prompt_3, since it generated results that resemble in many cases human key points both in style and length. As mentioned above, BERTopic extracts the four most representative arguments of each cluster, to be used as input for the representation fine-tuning process and the subsequent key point generation. Although the number of

representative arguments is a tunable parameter, we decided to keep the default value and generate our key points based on a maximum of 4 arguments per cluster. Knowing that each key point in our dataset contains a different number of matched arguments, we had to adjust the number of arguments in the few-shot demonstrations, so that only 4 arguments are kept for each key point. We experimented with 4-, 8- and 16-shots, where from each debatable topic one key point per stance (positive/negative) is taken into consideration, to ensure that the model sees arguments both for and against a topic and can condition its answer based on the stance.

Table 10: Few-shot results on ArgKP-2021-GR dev set

Meltemi-Instruct (1.5)	ROUGE			BERTScore			Avg token count
	1	2	L	P	R	F1	
0-shot Prompt_3 (beam_3)	16.9	6.0	15.9	67.6	73.5	70.4	20.96
4-shot Prompt_3 (beam_3)	23.5	10.1	22.6	72.8	75.7	74.2	10.68
8-shot Prompt_3 (beam_3)	24.8	11.0	24.3	74.7	75.4	75.0	8.23
16-shot Prompt_3 (beam_3)	**25.3**	**11.9**	**24.9**	**75.1**	**75.1**	**75.0**	**7.92**

Overall, all few-shot experiments proved effective in terms of generating 1-sentence key points. Although the automatic evaluation shows the 16-shot setting as the best performing, our choice was determined by the manual evaluation of the key points and the level of abstraction and the granularity that is needed for the matching task. We opted for the 4-shot setting for the final comparison, since it achieves competitive results if compared to 8- and 16-shot and requires less computational resources. For a manual comparison on the quality of the generated key points, refer to Appendix II-Table 19.

3.3 KPG Results

We generate key points on the human translated test set of ArgKP-2021 with the best models chosen from our zero- and few-shot experiments. Model settings were chosen based on those that gave the best performance in terms of automatic and manual evaluation on previous experiments. GreekWiki has been used with its summarization-specific prompt, while the rest three Meltemi-based approaches have been tested with Prompt_3, which previous experiments have shown as the most appropriate. For all 4 cases beam search decoding with beam width 3 is used.

Table 11: Final KPG results on ArgKP-2021-GR test set

Experiment setting	Rouge	Bertscore	Avg token count

	1	2	L	P	R	F1	
GreekWiki (0-shot)	12.3	3.6	11.0	66.0	67.5	66.7	24.08
Meltemi-base (0-shot)	13.2	2.3	11.5	66.9	69.1	68.0	19.5
Meltemi-Instruct (0-shot)	15.8	4.6	14.1	68.0	70.6	69.2	20.5
Meltemi-Instruct (4-shot)	**20.2**	**8.0**	**19.1**	**74.0**	**72.8**	**73.4**	**10.89**

We comment on each model separately and reach our overall conclusions at the end.

1. Our initial observations about the inability of GreekWiki, a small (580 million parameters) instruction-tuned model, to sufficiently perform the KPG task in a 0-shot setting have been validated. The outputs maintain the definition-like style of the pretraining data, which is not really a burden in terms of the simple sentence structure we are looking for in a key point, but the sentences still contain repetitions of common nouns and named entities, making the outputs semantically unmeaningful, or implausible. Nevertheless, there is a small portion of the outputs, which are indeed valid and meaningful, showing that further experimentation with fine-tuning techniques would be worthwhile.
2. Meltemi-base in the 0-shot setting shows the same behavior as in the validation set, by copying input arguments from the given prompt. This proves its inability to follow instructions and the need for more task-specific fine-tuning for generating abstractive outputs.
3. The outputs of Meltemi-Instruct in the zero-shot setting still indicate repetitions of existing arguments, like Meltemi-base, slightly paraphrased through syntactic changes, but the majority of produced outputs are abstractive, semantically meaningful sentences. Prompting the instruction-tuned model to produce output in a specified format helped the model generate succinct sentences. Nevertheless, we observe some inconsistency in the sentence length, as some sentences are too short, while others are too complex.
4. In the last experiment with Meltemi Instruct, we show that with only four example demonstrations we are able to increase all our automatic metrics up to about 5 points on average and produce keypoint-like sentences with average token length much closer to that of the reference key points (~7.8 tokens). The manual evaluation validated these findings, nevertheless, it is worth noting that some

output key points included hallucinations, by producing sentences with new entities, or they contained only the information of the topic, without the argumentation content, leading to many generic and unuseful for our task produced sentences. We attribute this behavior to the structure of many arguments in our dataset, which contain the topic information within the argument structure.

The above findings on each model's behavior on automatic and manual evaluation (see Appendix II: Table 20) led us to consider Meltemi-Instruct with 4-shot inference as our best-performing baseline.

4. CONCLUSIONS AND FUTURE WORK

KPA has brought the task of multi-document argument summarization into a new era. During this thesis we managed with our limited computational resources to transfer the task of KPA in Greek, a low-resource language. We created the Greek version of the official KPA dataset and used it in all our KPM and KPG experiments. We managed to replicate existing baselines, as well as create our own, with state-of-the-art Greek language models. The practical challenges that arose throughout this thesis, created interesting research directions for the Greek NLP community. Below we propose some directions for future research for KPM and KPG.

Starting from the former, our successful experiments with decoder-based models on classification tuning encourage further experimentation with other PEFT methods [91] that have been proven effective in classification tasks, as well as experimentation with KPM as a generative task, since Meltemi and the most decoder-only models are trained for text generation. Our experiments have also opened the path for the English language towards exploring an abundance of decoder-only models, which until now, to the best of our knowledge, have not been extensively explored in the KPM subtask. Nevertheless, taking into consideration the computational requirements (e.g., GPU memory) of Transformer-based matcher models, a future direction would be to experiment with combinations of faster but less accurate Sentence-Transformer models [72] with slower, heavier but more accurate models, in a similar manner like Eden et al. [15]. The aim is to identify the most efficient and scalable KPA system that would be capable of being used in real life use cases.

For KPG our manual evaluation has shown that it is required to capture more specific and lower-frequency arguments, in our existing argument clustering pipeline. The used HDBSCAN is a soft clustering approach that does not force every single argument to join a cluster [76]. This could be improved through iterative clustering similar to Li et al. [41], that forces unclustered arguments to join an existing cluster or create a new one. The experiment would be even more meaningful if we could also ensure the quality of our embedding representations. The availability of labels in our ArgKP-2021 dataset enables us to experiment with finetuning our sentence embeddings for the clustering task like Khosravani et al. [21] and Li et al. [43]. To this end, using a pretrained dedicated for Greek and not a multilingual sentence embedding model would be a very interesting direction, as such models are not yet available for Greek. Furthermore, since we have obtained promising results by experimenting with zero or few demonstrations to our models, the next step could be to experiment with fine-tuning on the available training data. We also think that training an argument-quality model, like that of Haim et al. [2], that identifies arguments based on specific properties such as clear stance, discussion of a single topic and maintenance of a balanced tradeoff between general and specific content, would substantially contribute to the quality of our KPG model. Nevertheless, this brings together new challenges, since as per Toledo-Ronen et al. (2020), the use of machine translated data produces unreliable models. Therefore, one straightforward step towards addressing this need is the human translation of the Argument Quality dataset [14], which is the standard resource for creating argument quality models.

With this work we hope to encourage future work towards optimizing KPA systems for the Greek language and explore their usability in new datasets, domains and applications.

ACRONYMS

EMNLP	Article Delivery Over Network Information Systems
KPA	Association For Library Collections and Technical Services
KPM/KPG	Transmission Control Protocol/ Internet Protocol
PEFT	Text Encoding Initiative
mAP	Universal System for information in Science and technology
TF-IDF	World Wide Web Consortium
LSA	Ένωση Ελλήνων Χρηστών Internet
POS	Part-Of-Speech
SVO	Subject Verb Object
BERT	Bidirectional Encoder Representations from Transformers
ELMo	Embeddings from Language Model
GloVe	Global Vectors (for Word Representation)
MNLI	Multi-Genre Natural Language Inference
SNLI	Stanford Natural Language Inference
GPT	Generative Pre-training Transformer
LLM	Large Language Model
sP/sR	Soft Precision/ soft Recall
ACL	Association for Computational Linguistics
AI	Artifficial Intelligence
NLP	Natural Language Processing
NLU	Natural Language Understanding
NLG	Natural Language Generation
LoRA	Low Rank adaptation
HDBSCAN	Hierarchical Density-based Spatial Clustering of Applications with Noise
UMAP	Uniform Manifold Approximation and Projection
DBCV	Density-Based Clustering Validation
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
mERT	multilingual BERT
GPU	Graphics Processing Unit

APPENDIX I

All KPM experiments were conducted on one P100 GPU node with 16GB memory.

Table 12: Enigma model hyperparameters

Classes=1	Loss: BCELoss
Batch_size=16	Warmup_steps: 0
Learning rate=1e-5	Optimizer: Adam
Epochs=3	Weight_decay: 0.01
Accumulation steps=2	Grad_clip: 1.0
Dropout=0.4	
trainable parameters: 110M (100% of the original model)	
training duration (+/- 1 h)	

Table 13: SMatchToPR model hyperparameters

Epochs: 10	Loss: contrastiveloss
Max_seq_len: 70	Warmup steps: 10% of train data
Train_batch_size: 32	
trainable parameters: 110M (100% of the original model)	
training duration (+/- 1 h)	

Table 14: Meltemi-base hyperparameters for KPM subtask

Classes:2	optimizer: paged Adam optimizer
epochs: 1	Seed: 42
max_seq_length: 512	LoRA r :8
batch_size: 16	LoRA alpha: 8
Gradient Accumulation Steps: 2	LoRA dropout: 0.0
learning_rate: 1e-4	LoRA bias: ‘none’

lr_scheduler_type: linear	target_modules: q_proj, v_proj
Weight Decay: 0.01	task_type: "SEQ_CLS"
M. G. Norm: 0.3	Loss: Binary Cross Entropy
trainable parameters: 3,416,064 (~5% of the original model)	
training duration (+/- 20 hours)	

APPENDIX II

All KPG experiments were conducted on one P100 GPU node with 16GB memory.

Table 15: Best hyperparameter values for each topic-stance combination

	'n_neighbors'	'n_components'	'min_samples_fraction'	'cluster_selection_method'
Topc1_stance1	30	2	1.0	‘leaf’
Topic1_stance-1	50	2	0.5	‘leaf’
Topic2_stance1	50	5	0.5	‘leaf’
Topic2_stance-1	50	2	1.0	‘leaf’
Topic3_stance1	50	5	0.5	‘leaf’
Topic3_stance-1	30	5	1.0	‘leaf’
Topic4_stance1	50	5	1.0	‘leaf’
Topic4_stance-1	10	2	1.0	‘leaf’
Prevalent value	50	tie	1.0	‘leaf’

Table 16: BERTopic final hyperparameters

embedding model	sentence-transformers/paraphrase-multilingual-mpnet-base-v2
umap model	n_neighbors=50, n_components=2, min_dist=0.00, metric='cosine')
hdbscan model	min_cluster_size=3, metric='euclidean',cluster_selection_method='leaf', min_samples=min_cluster_size,
Vectorizer model	CountVectorizer(ngram_range=(1,2),stop_words=stopwords, lowercase=True)

ctfidf_model	ClassTfidfTransformer(reduce_frequent_words=True)
nr_topics	10
language	‘multilingual’

Table 17: Zero-shot examples on ArgKP-2021-GR dev set

Topic: Θα πρέπει να υιοθετήσουμε καθεστώς λιτότητας/We should adopt an austerity regime Stance: negative (-1)			
	GreekWiki	Meltemi-base	Meltemi-Instruct
0-shot greedy Model-specific prompt (GR)	1.Η ύφεση του μερεστρόζε (αγγλικά: μερεστρόζε) είναι μια ιδεολογία που προτάθηκε από τον Τζον Χ. Ρόμπερτς (John H. Roberts) το 1925. 2. Η λιτότητα είναι ένα μέτρο που χρησιμοποιείται για να περιορίσει την παιδική φτώχεια και την παιδική φτώχεια. 3. Το καθεστώς λιτότητας είναι μια κατάσταση στην οποία οι άνθρωποι που υιοθετούν ένα καθεστώς λιτότητας δεν μπορούν να υιοθετήσουν ένα καθεστώς λιτότητας.	1. δεν χρειαζόμαστε καθεστώς λιτότητας. 2. η λιτότητα συνήθως αφαιρεί από τους φτωχούς όταν περικόπτονται τα επιδόματα πρόνοιας. 3. ένα καθεστώς λιτότητας θα αναγκάσει τους φορολογούμενους να πληρώσουν περισσότερους φόρους.	1. Δεν χρειαζόμαστε καθεστώτα λιτότητας, καθώς έχουμε αρκετό χρέος και η τεχνολογία θα μειώσει τα έξοδα διαβίωσης. 2. Τα καθεστώτα λιτότητας πλήττουν τον απλό άνθρωπο, καθώς μειώνουν τα χρήματα που ξοδεύονται για τα σχολεία και την υγειονομική περίθαλψη. 3. Ένα καθεστώς λιτότητας θα ανάγκαζε τους φορολογούμενους να πληρώσουν περισσότερους φόρους, κάτι που θα μείωνε τα έσοδα των σκληρά εργαζόμενων πολιτών και θα αύξανε τα επίπεδα φτώχειας.

0-shot greedy Model-specific prompt (EN)[14]	1.The recession of merestrose is an ideology proposed by John H. Roberts in 1925. 2. Austerity is a measure used to reduce child poverty and child poverty. 3. Austerity is a situation in which people who adopt an austerity regime cannot adopt an austerity regime.	1. we do not need an austerity regime. 2. austerity usually takes away from the poor when welfare benefits are cut. 3. an austerity regime will force taxpayers to pay more taxes.	1. We don't need austerity regimes, as we have enough debt, and technology will reduce our living costs. 2. Austerity regimes hurt the common citizen as they reduce the money spent on schools and healthcare. 3. An austerity regime would force taxpayers to pay more taxes, which would reduce the income of hard-working citizens and increase poverty levels.
0-shot beam_3 Model-specific prompt (GR)	1.Ο όρος καθεστώς λιτότητας (αγγλικά: καθεστώς ελάττωσης) αναφέρεται στο καθεστώς που επιβαρύνει υπερβολικά την οικονομία της χώρας 2. Το καθεστώς λιτότητας είναι ένα καθεστώς το οποίο μπορεί να επιφέρει οικονομικές ζημιές στους κρατικούς μηχανισμούς. 3. Ένα καθεστώς λιτότητας είναι ένα τέχνασμα της κυβέρνησης για να ελέγχει τα χρήματά της.	1. το καθεστώς λιτότητας επιβαρύνει υπερβολικά την οικονομία. 2. δεν θα πρέπει να εισαχθούν καθεστώτα λιτότητας, καθώς αυτά θα έχουν αρνητική επίδραση στις υπηρεσίες που χρησιμοποιούν τα πιο ευάλωτα κοινωνικά στρώματα. 3. η κυβέρνηση είναι εντάξει όπως είναι	1. Το καθεστώς λιτότητας επιβαρύνει υπερβολικά την οικονομία, καθώς ο περιορισμός των δαπανών οδηγεί τις κυβερνήσεις σε ύφεση, περιορίζοντας τις οικονομικές δραστηριότητες πολύ περισσότερο. 2. Δεν θα πρέπει να υιοθετήσουμε ένα καθεστώς λιτότητας γιατί αυτό θα προκαλέσει μόνιμη ζημιά στους κρατικούς μηχανισμούς μας, οι οποίοι χρειάστηκαν χρόνια για να δημιουργηθούν, και

[14] All English versions of predicted key points were machine translated through Deepl.

			γιατί ένα τέτοιο καθστώς μπορεί να αφαιρέσει χρηματοδότηση από προγράμματα που είναι αναγκαία, όπως αυτό για τα συσσίτια παιδιών. 3. Ο ομιλητής πιστεύει ότι η κυβέρνηση δεν χρειάζεται αλλαγή και ότι ένα καθεστώς λιτότητας θα ήταν λάθος.
0-shot beam_3 Model-specific prompt (EN)	1.The term austerity regime refers to a regime that places an excessive burden on the economy of a country. 2. The austerity regime is a regime which can cause economic damage to the state mechanisms. 3. An austerity regime is a ploy by the government to control its money.	1. the austerity regime puts an excessive burden on the economy. 2. austerity regimes should not be introduced as they will have a negative impact on the services used by the most vulnerable in society. 3. the government is fine as it is.	1. The austerity regime puts too much strain on the economy, as spending restraint drives governments into recession, restricting economic activity much further. 2. We should not adopt an austerity regime because it will cause permanent damage to our government mechanisms, which took years to create, and because such a regime can take funding away from much-needed programs such as children's soup kitchens. 3. The speaker believes that the government does not need change and that an austerity regime would be a mistake.
0-shot beam_3	1. Το Κώδικας Λατότητας είναι ένα	1. Δεν θα πρέπει να υιοθετήσουμε ένα	1. Επένδυση στην εκπαίδευση των

Prompt_3 (GR)	μοτίβο που συνοψίζει μια πρόταση που συνοψίζει ένα συγκεκριμένο θέμα. 2. Το Κώδικας Λαϊκής Απελευθέρωσης είναι ένα μοτίβο που συνοψίζει μια πρόταση που συνοψίζει μια πρόταση που συνοψίζει το μοτίβο. 3. Το σχίσμα της λιτότητας είναι ένα μοτίβο που συνοψίζει μια πρόταση που συνοψίζει μια πρόταση της κυβέρνησης για την αλλαγή του τρόπου με τον οποίο η κυβέρνηση μπορεί να ελέγχει το χρήμα της.	καθεστώς λιτότητας, διότι ο καλύτερος τρόπος για την ανάπτυξη της χώρας είναι να επενδύσουμε στην εκπαίδευση των νέων, ώστε να εξελιχθούν και να γίνουν πιο παραγωγικοί. 2. περικόπτοντας τα κοινωνικά προγράμματα θα πλήξετε τους ανθρώπους που βασίζονται σε αυτά. 3. η κυβέρνηση είναι εντάξει όπως είναι.	νέων για την ανάπτυξη της χώρας 2. Η εφαρμογή μέτρων λιτότητας θα έβλαπτε τους ανθρώπους που βασίζονται σε κοινωνικά προγράμματα 3. Η κυβέρνηση είναι ήδη πολύ ελεγκτική και οι άνθρωποι έχουν το δικαίωμα να έχουν τον έλεγχο των χρημάτων τους
0-shot beam_3 Prompt_3 (EN)	1. The Latitude Code is a pattern that summarizes a sentence that summarizes a specific topic. 2. The People's Liberation Code is a pattern that summarizes a sentence that summarizes a pattern. 3. The austerity schism is a pattern that summarizes a proposition that summarizes a proposition for the government to change the way the	1. We should not adopt an austerity regime because the best way to develop the country is to invest in the education of young people so that they can develop and become more productive. 2. cutting social programs will hurt the people who rely on them. 3. the government is fine as it is.	1. Investing in youth education for the country's development 2. Implementing austerity measures would hurt people who rely on social programs 3. Government is already too controlling, and people have the right to have control of their money

	government can control its money.		
References (GR)	1. Η λιτότητα περιορίζει την πρόσβαση σε βασικές υπηρεσίες 2. Η λιτότητα παρατείνει την ύφεση 3. Η λιτότητα είναι άδικη για τους πολίτες 4. Η λιτότητα οδηγεί σε χαμηλότερη ανάπτυξη 5. Η λιτότητα οδηγεί σε δημιουργία λιγότερων θέσεων εργασίας		
References (EN)	1. Austerity cuts access to essential services 2. Austerity extend recessions 3. Austerity is unfair to the citizens 4. Austerity results in lower growth 5. Austerity results in lower job creation		

Table 18: Custom prompt template examples on ArgKP-2021-GR dev set

Examples	GR	EN
Prompt_1	1. Δεν θα πρέπει να υιοθετήσουμε ένα καθεστώς λιτότητας καθώς ο καλύτερος τρόπος για την ανάπτυξη της χώρας είναι να επενδύσουμε στην εκπαίδευση των νέων ώστε να εξελιχθούν και να γίνουν πιο παραγωγικοί. 2. Ένα καθεστώς λιτότητας θα έβλαπτε τους ανθρώπους που βασίζονται σε κοινωνικά προγράμματα για να επιβιώσουν και θα είχε αλυσιδωτές επιπτώσεις στην οικονομία. 3. Τα καθεστώτα λιτότητας μπορούν να έχουν αρνητικές επιπτώσεις στην ανάπτυξη της οικονομίας επηρεάζοντας αρνητικά τη ζήτηση και εμποδίζοντας την ανάπτυξη.	1. We should not adopt an austerity regime as the best way to develop the country is to invest in the education of young people so that they can develop and become more productive. 2. An austerity regime would hurt people who rely on social programs to survive and would have a chain effect on the economy. 3. Austerity regimes can have a negative impact on the growth of the economy by negatively affecting demand and hindering growth.
Prompt_2	1. Η επένδυση στην εκπαίδευση των νέων είναι ο καλύτερος τρόπος για την ανάπτυξη της χώρας 2. Τα επιχειρήματα υποστηρίζουν ότι ένα καθεστώς λιτότητας θα έβλαπτε τους ανθρώπους	1. Investing in youth education is the best way to develop the country 2. The arguments support that an austerity regime would hurt people who rely on social programs and have a chain effect on the

	που βασίζονται σε κοινωνικά προγράμματα και θα είχε αλυσιδωτές επιπτώσεις στην οικονομία ενώ τα επιχειρήματα αντικρούουν ότι ένα καθεστώς λιτότητας θα βοηθούσε να μείνει ο προϋπολογισμός υπό έλεγχο και να σταματήσει η κυβέρνηση να ξοδεύει χρήματα σε ασήμαντα πράγματα. 3. Τα καθεστώτα λιτότητας μπορούν να οδηγήσουν σε χαμηλότερη ανάπτυξη και χαμηλότερα φορολογικά έσοδα επηρεάζοντας αρνητικά την οικονομία και εμποδίζοντας την ανάπτυξη	economy while the arguments counter that an austerity regime would help keep the budget under control and stop the government from spending money on trivial things. 3. Austerity regimes can lead to lower growth and lower tax revenues by negatively affecting the economy and hindering growth
Prompt_3	1. Επένδυση στην εκπαίδευση των νέων για την ανάπτυξη της χώρας 2. Η εφαρμογή μέτρων λιτότητας θα έβλαπτε τους ανθρώπους που βασίζονται σε κοινωνικά προγράμματα για βοήθεια. 3. Οι πολιτικές λιτότητας μπορεί να οδηγήσουν σε χαμηλότερη ανάπτυξη και χαμηλότερα φορολογικά έσοδα επηρεάζοντας αρνητικά την οικονομία.	1. Investing in youth education for the development of the country 2. Implementing austerity measures would hurt people who rely on social programs for help. 3. Austerity policies may lead to lower growth and lower tax revenues negatively affecting the economy.

Table 19: Few-shot examples on ArgKP-2021-GR dev set

Topic: Θα πρέπει να υιοθετήσουμε καθεστώς λιτότητας/We should adopt an austerity regime Stance: -1		
4shot	8 shot	16shot
1. Το καθεστώς λιτότητας θα ήταν καταστροφικό για την οικονομία 2. Η λιτότητα βλάπτει την κοινωνική πρόνοια	1. Η λιτότητα είναι καταστροφική 2. Η λιτότητα βλάπτει την κοινωνική πρόνοια 3. Η κυβέρνηση είναι ήδη	1. Το καθεστώς λιτότητας είναι καταστροφικό 2. Η λιτότητα βλάπτει την κοινωνική πρόνοια 3. Η κυβέρνηση είναι ήδη

3. Η κυβέρνηση είναι ήδη πολύ ελεγκτική και οι άνθρωποι έχουν το δικαίωμα να έχουν τον έλεγχο των χρημάτων τους	πολύ ελεγκτική και οι άνθρωποι έχουν το δικαίωμα να έχουν τον έλεγχο των χρημάτων τους	ελεγκτική
1. The austerity regime would be disastrous for the economy 2. Austerity would harm social welfare 3. Government is already too controlling, and people have the right to have control over their money	1. Austerity is destructive 2. Austerity would harm social welfare 3. Government is already too controlling, and people have the right to have control over their money	1. The austerity regime is disastrous 2. Austerity harms social welfare 3. The government is already controlling

Table 20: Final results ArgKP-2021-GR test set example

topic: Οι ΗΠΑ είναι μια καλή χώρα για να ζει κανείς/ The USA is a good country to live in Stance: negative -1			
GreekWiki-0shot	Meltemi_base_ 0shot	Meltemi_Instruct_ 0shot	Meltemi_Instruct_ 4shot
1. Η Λατινική Αμερική είναι μια Λατινική Αμερική που ορίζεται ως η Λατινική Αμερική. 2. Ο όρος υγειονομική περίθαλψη χρησιμοποιείται για να περιγράψει την υγειονομική περίθαλψη που παρέχεται στους πολίτες μιας χώρας. 3. Ο όρος «Συμπεριλαμβανομένων των χωρών της Αμερικής και των Ηνωμένων Πολιτειών» χρησιμοποιείται για να περιγράψει τις χώρες στις οποίες οι νόμοι είναι πιο περίπλοκες από εκείνες των χωρών της Ασίας και της Ωκεανίας	1. υπάρχουν πολλά φυλετικά προβλήματα που θέτουν σε κίνδυνο την κοινωνική ειρήνη. 2. Οι ΗΠΑ είναι μια πολύ καπιταλιστική χώρα, δεν παρέχει υγειονομική περίθαλψη και εκπαίδευση στους πολίτες της καθολικά. 3. Αυτή η χώρα έχει πολλή εγκληματικότητα, την οποία ο κόσμος δεν αντιλαμβάνεται, πρέπει να φροντίσετε γι' αυτό	1. Οι Ηνωμένες Πολιτείες υπήρξαν για χρόνια κοιτίδα ρατσισμού κατά των πολιτισμών των μειονοτήτων. 2. Οι ΗΠΑ είναι μια πολύ καπιταλιστική χώρα, δεν παρέχει καθολικά υγειονομική περίθαλψη και εκπαίδευση 3. Η εγκληματικότητα και τα πολιτικά προβλήματα στις ΗΠΑ	1. Οι Ηνωμένες Πολιτείες έχουν ένα ιστορικό φυλετικών διακρίσεων. 2. Το σύστημα υγείας των ΗΠΑ είναι πολύ ακριβό και δημιουργεί διακρίσεις 3. Η εγκληματικότητα στις ΗΠΑ είναι ένα σημαντικό πρόβλημα
1. Latin America is a	1. there are many	1. The United	1. The United

Latin America defined as Latin America. 2. The term health care is used to describe the health care provided to the citizens of a country. 3. The term "Including countries of the Americas and the United States" is used to describe countries in which the laws are more complex than those of Asian and Oceania countries	racial problems that endanger social peace. 2. The US is a very capitalist country; it does not provide health care and education to its citizens universally. 3. This country has a lot of criminality that people don't realize, you need to take care of it	States has for years been a hotbed of racism against minority cultures. 2. The US is a very capitalist country, it does not provide universal health care and education 3. Criminality and political problems in the US	States has a history of racial discrimination. 2. The US health care system is very expensive and discriminatory 3. Criminality in the US is a major problem
References (GR)	1. Οι Ηνωμένες Πολιτείες έχουν άδικες πολιτικές στους τομείς της υγείας και της εκπαίδευσης 2. Οι Ηνωμένες Πολιτείες έχουν ένα προβληματικό/διχαστικό πολιτικό σύστημα 3. Οι Ηνωμένες Πολιτείες έχουν υψηλή φορολογία και υψηλό κόστος διαβίωσης 4. Στις Ηνωμένες Πολιτείες υπάρχει ξενοφοβία και ρατσισμός 5. Οι Ηνωμένες Πολιτείες έχουν ανισότητες και φτώχεια 6. Στις Ηνωμένες Πολιτείες δεν υπάρχει ασφάλεια 7. Στις Ηνωμένες Πολιτείες υπάρχει η αρνητική κουλτούρα		
References (EN)	1. The US has unfair health and education policies 2. The US has a problematic/divisive political system 3. The US has high taxation/high costs of living 4. The US is xenophobic/racist 5. The US has inequality/poverty 6. The US is unsafe 7. The US has a negative culture		

REFERENCES

[1] R. Bar-Haim, L. Eden, R. Friedman, Y. Kantor, D. Lahav, and N. Slonim, "From arguments to key points: Towards automatic argument summarization," in *Proc. 58th Annu. Meeting of the ACL*, 2020, pp.4029-4039. [Online]. Available: https://aclanthology.org/2020.acl-main.371.pdf. [Accessed: Jan. 13, 2025].

[2] R. Bar-Haim, Y. Kantor, L. Eden, R. Friedman, D. Lahav, and N. Slonim, "Quantitative Argument Summarization and Beyond: Crossdomain Key point Analysis." In *Proc. 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 39–49. [Online]. Available: https://arxiv.org/abs/2010.05369. [Accessed: Jan. 13, 2025].

[3] R. Friedman, L. Dankin, Y. Hou, R. Aharonov, Y. Katz, and N. Slonim, "Overview of the 2021 Key Point Analysis Shared Task", in *Proc. 8th Workshop on Argument Mining*, 2021, pp. 154-164. [Online]. Available: https://aclanthology.org/2021.argmining-1.16/. [Accessed: Jan. 13, 2025].

[4] J. Koutsikakis, I. Chalkidis, P. Malakasiotis, and I. Androutsopoulos, "GREEK-BERT: The Greeks visiting Sesame Street," in *11th Hellenic Conference on Artificial Intelligence*, Sep. 2020, [Online]. Available: https://doi.org/10.1145/3411408.3411440. [Accessed: Jan. 13, 2025].

[5] N. Giarelis, C. Mastrokostas, and N. Karacapilidis, "GreekT5: A Series of Greek Sequence-to-Sequence Models for News Summarization." [Online]. Available: https://arxiv.org/pdf/2311.07767. [Accessed: Jan. 13, 2025].

[6] L. Voukoutis, D. Roussis, G. Paraskevopoulos, S. Sofianopoulos, P. Prokopidis, V. Papavasileiou, A.Katsamanis, S. Piperidis, V. Katsouros, "Meltemi: The first open Large Language Model for Greek," *arXiv*, Jul. 2024, [Online]. Available: https://doi.org/10.48550/arxiv.2407.20743. [Accessed: Jan. 13, 2025].

[7] J. Lawrence and C. Reed, "Argument Mining: A Survey," *Computational Linguistics*, pp. 1–55, Oct. 2019, [Online]. Available: https://doi.org/10.1162/coli_a_00364. [Accessed: Jan. 13, 2025].

[8] A. Misra, B. Ecker, and M. A. Walker, "Measuring the Similarity of Sentential Arguments in Dialogue," in *Proc. 17th Annu. Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 276–287, Jan. 2016, [Online]. Available: https://doi.org/10.18653/v1/w16-3636. [Accessed: Jan. 13, 2025].

[9] Y. Ajjour, M. Alshomary, H. Wachsmuth, and B. Stein, "Modeling Frames in Argumentation," in Proc. *ACL*, Nov. 2019. [Online]. Available: https://aclanthology.org/D19-1290. [Accessed: Jan. 13, 2025].

[10] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, Sep. 1990, [Online]. Available:https://doi.org/10.1002/(sici)1097-4571(199009)41:6%3C391::aid-asi1%3E3.0.co;2-9. [Accessed: Jan. 13, 2025].

[11] N. Reimers, B. Schiller, T. Beck, J. Daxenberger, C. Stab, and I. Gurevych, "Classification and Clustering of Arguments with Contextualized Word Embeddings," 2019. [Online]. Available: https://aclanthology.org/P19-1054.pdf. [Accessed: Jan. 13, 2025].

[12] J. Devlin, M.-W. Chang, K. Lee, K. Google, and A. Language, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv, May 2019. [Online]. Available: https://arxiv.org/pdf/1810.04805. [Accessed: Jan. 13, 2025].

[13] M. Peters, M. Neumann, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations." arXiv , 2018. [Online]. Available: https://arxiv.org/pdf/1802.05365

[14] S. Gretz, "A Large-Scale Dataset for Argument Quality Ranking: Construction and Analysis," in *Proc. AAAI Conf. on Artificial Intelligence*, vol. 34, no. 05, pp. 7805-7813, Apr. 2020. [Online]. Available: https://arxiv.org/pdf/1911.11408. [Accessed: Jan. 13, 2025].

[15] L. Eden, Y. Kantor, M. Orbach, Y. Katz, N. Slonim and R. Bar-Haim, "Welcome to the Real World: Efficient, Incremental and Scalable Key Point Analysis", in *Proc. 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track,* pp. 483-491, Dec. 2023. [Online]. Available: https://aclanthology.org/2023.emnlp-industry.46.pdf. [Accessed: Jan. 13, 2025].

[16] O. Sultan, R. Dhahri, Y. Mardan, T. Eder, and G. Groh, "From Judgement's Premises Towards Key Points," Dec. 23, 2022. [Online]. Available: https://www.researchgate.net/publication/366587552_From_Judgement. [Accessed: Jan. 13, 2025].

[17] P. Ehnert and J. Schröter, “Key point generation as an instrument for generating core statements of a political debate on Twitter,” in *Frontiers in Artificial Intelligence*, vol. 7, Mar. 2024, [Online]. Available: https://doi.org/10.3389/frai.2024.1200949. [Accessed: Jan. 13, 2025].
[18] M. Van Der Meer, P. Vossen, C. Jonker, and P. Murukannaiah, “An Empirical Analysis of Diversity in Argument Summarization,” *in ACL*, vol. 1: Long Papers, pp. 2028-2045, 2024. [Online]. Available: https://aclanthology.org/2024.eacl-long.123.pdf. [Accessed: Jan. 13, 2025].
[19] R. Bar-Haim, Y. Kantor, R. Friedman, and N. Slonim, “Every Bite Is an Experience: Key Point Analysis of Business Reviews,” in *Proc. of the 59th Annual Meeting of the ACL and the 11th Int. Joint Conf. on Natural Language Processing* Aug. 2021. Pp. 3376-3386 [Online]. Available: https://aclanthology.org/2021.acl-long.262.pdf. [Accessed: Jan. 13, 2025].
[20] A. Tang, X. Zhang, and M. Dinh, “Aspect-based Key Point Analysis for Quantitative Summarization of Reviews,” 2024. [Online]. Available: https://aclanthology.org/2024.findings-eacl.96.pdf. [Accessed: Jan. 13, 2025].
[21] M. Khosravani, C. Huang, and A. Trabelsi, “Enhancing Argument Summarization: Prioritizing Exhaustiveness in Key Point Generation and Introducing an Automatic Coverage Evaluation Metric,” in Proc. of the NAACL, vol. 1, pp. 8212–8224, 2024. [Online]. Available: https://aclanthology.org/2024.naacl-long.454.pdf. [Accessed: Jan. 13, 2025].
[22] J. Pennington, R. Socher, and C. Manning, “GloVe: Global Vectors for Word Representation,” Association for Computational Linguistics, 2014. [Online]. Available: https://aclanthology.org/D14-1162.pdf. [Accessed: Jan. 13, 2025].
[23] Y. Liu *et al.*, “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” 2019. [Online]. Available: https://arxiv.org/pdf/1907.11692. [Accessed: Jan. 13, 2025].
[24] P. He, X. Liu, J. Gao, and W. Chen, “DeBERTa: Decoding-enhanced BERT with Disentangled Attention,” Jun. 2020, [Online]. Available: https://doi.org/10.48550/arxiv.2006.03654. [Accessed: Jan. 13, 2025].
[25] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, “ALBERT: A Lite BERT for Self-supervised Learning of Language Representations,” *arXiv*, Apr. 2020, [Online]. Available: https://doi.org/10.48550/arxiv.1909.11942. [Accessed: Jan. 13, 2025].
[26] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. Le, “XLNet: Generalized Autoregressive Pretraining for Language Understanding.” Available: https://arxiv.org/pdf/1906.08237
[27] J. Reimer, T. Kim, H. Luu, M. Henze, and Y. Ajjour, “Modern Talking in Key Point Analysis: Key Point Matching using Pretrained Encoders,” 2021. [Online]. Available: https://aclanthology.org/2021.argmining-1.18.pdf. [Accessed: Jan. 13, 2025].
[28] N. M. Kapadnis, S. Patnaik, S. Panigrahi, V. Madhavan, and A. Nandy, “Team Enigma at ArgMining-EMNLP 2021: Leveraging Pre-trained Language Models for Key Point Matching,” 2021. [Online]. Available: https://aclanthology.org/2021.argmining-1.21.pdf. [Accessed: Jan. 13, 2025].
[29] E. Cosenza, “Key Point Matching with Transformers,” 2021. [Online]. Available: https://aclanthology.org/2021.argmining-1.20.pdf. [Accessed: Jan. 13, 2025].
[30] M. Alshomary *et al.*, “Key Point Analysis via Contrastive Learning and Extractive Argument Summarization,” 2021. [Online]. Available: https://aclanthology.org/2021.argmining-1.19.pdf. [Accessed: Jan. 13, 2025].
[31] A.M. Samin, B. Nikandish, and J. Chen, “Arguments to Key Points Mapping with Prompt-based Learning.” [Online]. Available: https://aclanthology.org/2022.icnlsp-1.36.pdf.[Accessed: Jan. 13, 2025].
[32] M. Lewis *et al.*, “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension,” Oct. 2019. [Online]. Available: https://arxiv.org/pdf/1910.13461
[33] C. Raffel *et al.*, “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer,” *Journal of Machine Learning Research*, vol. 21, pp. 1–67, 2020, Available: https://arxiv.org/pdf/1910.10683
[34] F. Zhao, F. Yang, T. J. Trull, and Y. Shang, “A New Method Using LLMs for Keypoints Generation in Qualitative Data Analysis,” Jun. 2023, [Online]. Available: https://doi.org/10.1109/cai54212.2023.00147.
[35] H.W Chung *et al.*, “Scaling Instruction-Finetuned Language Models,” 2022. [Online]. Available: https://arxiv.org/pdf/2210.11416
[36] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, “QLoRA: Efficient Finetuning of Quantized LLMs,” *arXiv.org*, May 23, 2023. [Online]. Available: https://arxiv.org/abs/2305.14314. [Accessed: Jan. 13, 2025].

[37] "The Project Debater Service API," *Ibm.com*, 2025. [Online]. Available: https://developer.ibm.com/apis/catalog/debater--project-debater-service-api/Introduction
[38] A. Turpin and F. Scholer, "User performance versus precision measures for simple search tasks," in *Proc. of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, Aug. 2006, [Online]. Available: https://doi.org/10.1145/1148170.1148176. [Accessed: Jan. 13, 2025].
[39] J. Zhang, Y. Zhao, M. Saleh, P.J. Liu, "PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization," in *Proc. of the 37th Int. Conf. on Machine Learning*, 2020. [Online]. Available: https://arxiv.org/pdf/1912.08777
[40] C.Y. Lin, "Rouge: A package for automatic evaluation of summaries," InText summarization branches out, Jul. 2004 (pp. 74-81)
[41] H. Li, V. Schlegel, R. Batista-Navarro, and G. Nenadic, "Do You Hear The People Sing? Key Point Analysis via Iterative Clustering and Abstractive Summarisation," in *Proc. Of the 61st Ann. Meeting of the ACL*, vol. 1, pp. 14064-14080, 2023. [Online]. Available: https://aclanthology.org/2023.acl-long.786.pdf. [Accessed: Jan. 13, 2025].
[42] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193–218, Dec. 1985, [Online]. Available: https://doi.org/10.1007/BF01908075. [Accessed: Jan. 13, 2025].
[43] X. Li, Y. Jiang, S. Huang, P. Xie, G. Cheng, and F. Huang, "Exploring Key Point Analysis with Pairwise Generation and Graph Partitioning," in *Proc. of the 2024 Conf. of the North American Chapter of the ACL: Human Language Technologies,",* vol. 1, pp. 5657–5667, Jan. 2024, [Online]. Available: https://doi.org/10.18653/v1/2024.naacl-long.315. [Accessed: Jan. 13, 2025].
[44] neulab, "GitHub - neulab/BARTScore: BARTScore: Evaluating Generated Text as Text Generation," *GitHub*, 2021. https://github.com/neulab/BARTScore
[45] T. Sellam, D. Das, and A. Parikh, "BLEURT: Learning Robust Metrics for Text Generation," in Proc. of the 58th Annu. Meeting of the ACL, 2020. [Online]. Available: https://aclanthology.org/2020.acl-main.704.pdf. [Accessed: Jan. 13, 2025].
[46] T. Zhang, V. Kishore, F. Wu, K. Weinberger, and Y. Artzi, "BERTSCORE: EVALUATING TEXT GENERATION WITH BERT." [Online]. Available: https://arxiv.org/pdf/1904.09675. [Accessed: Jan. 13, 2025].
[47] G. Giannakopoulos, "Multi-document multilingual summarization and evaluation tracks in ACL 2013 MultiLing Workshop." In *Proc. of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization.* Aug. 2013, [Online]. Available: https://users.iit.demokritos.gr/~ggianna/Publications/MultiDocTrackOverview.pdf. [Accessed: Jan. 13, 2025].
[48] P. Tamvakidis, "Argumentative sentence classification using transfer learning across languages," M.S. thesis, University of Piraeus, NCSR "Demokritos," Piraeus, Greece, 2021. [Online]. Available: https://dione.lib.unipi.gr/xmlui/bitstream/handle/unipi/13652/Master_Thesis_Tamvakidis_Panagiotis-signed.pdf?sequence=1&isAllowed=y. [Accessed: Jan. 13, 2025].
[49] A. Polykratis, "Argument mining using multitask learning," M.S. thesis, University of the Peloponese, Peloponese, Greece, 2017.
[50] S. Spiliopoulos, "Argument mining using graph neural networks," M.S. thesis, University of the Peloponese, Peloponese, Greece, 2018.
[51] C. Sardianos, I. Katakis, G. Petasis, and V. Karkaletsis, "Argument extraction from news," in *Proc. of the 2nd Workshop on Argumentation Mining*, 2015. [Online]. Available: https://aclanthology.org/W15-0508.pdf [Accessed: Jan. 14, 2025].
[52] N. Karousos, G. Vorvilas, Despoina Pantazi, and V. S. Verykios, "A Hybrid Text Summarization Technique of Student Open-Ended Responses to Online Educational Surveys," *Electronics*, vol. 13, no. 18, Sep. 2024, [Online]. Available: https://doi.org/10.3390/electronics13183722.
[53] I. Evdaimon, H. Abdine, C. Xypolopoulos, S. Outsios, M. Vazirgiannis, and G. Stamou, "GreekBART: The first pretrained Greek sequence-to-sequence model," in *Proc. of the 13th Int. Conf. on Language Resources and Evaluation (LREC)*, 2024. [Online]. Available: https://aclanthology.org/2024.lrec-main.700.pdf [Accessed: Jan. 14, 2025].
[54] K. Papantoniou and Y. Tzitzikas, "NLP for the Greek Language: A Brief Survey," in *Proc. of the 11th Hellenic Conf. on Artificial Intelligence*, Sep. 2020, doi: https://doi.org/10.1145/3411408.3411410.
[55] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *Proc. of the 10th Machine Translation Summit*, Phuket, Thailand, 2005. [Online]. Available: https://aclanthology.org/2005.mtsummit-papers.11.pdf. [Accessed Jan. 16, 2025].

[56] P. Ortiz Suárez, L. Romary, and B. Sagot, "A Monolingual Approach to Contextualized Word Embeddings for Mid-Resource Languages," in *Proc. of the 58th Annu. Meeting of ACL*, 2020.[Online]. Available: https://aclanthology.org/2020.acl-main.156.pdf.[Accessed: Jan. 14, 2025].
[57] Stamatis Outsios, Konstantinos Skianis, Polykarpos Meladianos, Christos Xypolopoulos, and Michalis Vazirgiannis, "Word Embeddings from Large-Scale Greek Web Content," *arXiv*, Jan. 2018, [Online]. Available: https://doi.org/10.48550/arxiv.1810.06694.
[58] J. Bakagianni, K. Pouli, M. Gavriilidou, and J. Pavlopoulos, "Towards Systematic Monolingual NLP Surveys: GenA of Greek NLP," *arXiv*, 2024. [Online]. Available: https://arxiv.org/abs/2407.09861 [Accessed Jan. 14, 2025].
[59] L. Xue *et al.*, "mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer." *arXiv, 2021.* [Online]. Available: https://arxiv.org/pdf/2010.11934. [Accessed Jan. 16, 2025].
[60] H. W. Chung, N. Constant, X. Garcia, A. Roberts, Y. Tay, S. Narang, and O. Firat, "UniMax: Fairer and More Effective Language Sampling for Large-Scale Multilingual Pretraining," in *Proc. of the International Conference on Learning Representations (ICLR)*, 2023. [Online]. Available: https://openreview.net/forum?id=kXwdL1cWOAi. [Accessed: Jan. 14, 2025].
[61] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. Le Scao, T. Lavril, T. Wang, T. Lacroix, and W. El Sayed, "Mistral 7B: A compact yet capable language model," *arXiv*, Oct. 2023. [Online]. Available: https://arxiv.org/pdf/2310.06825. [Accessed: Jan. 14, 2025].
[62] Leonvouk, "Meltemi: A Large Language Model for Greek - Institute for Language and Speech Processing / Athena RC - Medium," *Medium*, Mar. 26, 2024. https://medium.com/institute-for-language-and-speech-processing/meltemi-a-large-language-model-for-greek-9f5ef1d4a10f. [Accessed Jan. 18, 2025].
[63] S. Eger, J. Daxenberger, C. Stab, and I. Gurevych, "Cross-lingual Argumentation Mining: Machine Translation (and a bit of Projection) is All You Need!" in *Proc. of the 27th Int. Conf. on Computational Linguistics*, 2018. [Online]. Available: https://aclanthology.org/C18-1071.pdf. [Accessed: Jan. 14, 2025].
[64] O. Toledo-Ronen, M. Orbach, Y. Bilu, A. Spector, and N. Slonim, "Multilingual Argument Mining: Datasets and Analysis," in *Findings of the ACL: EMNLP 2020*, 2020. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.29.pdf. [Accessed: Jan. 14, 2025].
[65] Kaggle, "Kaggle: Your home for data science," *Kaggle.com*, 2024. https://www.kaggle.com/
[66] S. Kudugunta *et al.*, "MADLAD-400: A Multilingual And Document-Level Large Audited Dataset," *arXiv*, 2023. [Online]. Available: https://arxiv.org/abs/2309.04662. [Accessed Jan. 16, 2025].
[67] "Linguistic Features · spaCy Usage Documentation," https://spacy.io/usage/linguistic-features#pos-tagging. [Accessed: Jan. 14, 2025]
[68] A. Benayas, M. A. Sicilia, and Marçal Mora-Cantallops, "A Comparative Analysis of Encoder Only and Decoder Only Models in Intent Classification and Sentiment Analysis: Navigating the Trade-Offs in Model Size and Performance," Jan. 15, 2024. [Online]. Available: https://www.researchgate.net/publication/377467915_A_Comparative_Analysis_of_Encoder_Only_and_Decoder_Only_Models_in_Intent_Classification_and_Sentiment_Analysis_Navigating_the_Trade-Offs_in_Model_Size_and_Performance. [Accessed Jan. 16, 2025].
[69] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei, "Improving Text Embeddings with Large Language Models." *arXiv*, 2024. [Online]. Available: https://arxiv.org/pdf/2401.00368. [Accessed Jan. 16, 2025].
[70] E. Hu *et al.*, "LORA: LOW-RANK ADAPTATION OF LARGE LANGUAGE MODELS," *arXiv*, Oct. 2021. [Online]. Available: https://arxiv.org/pdf/2106.09685. [Accessed Jan. 16, 2025].
[71] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *31st Conf. on Neural Information Processing Systems*, Jun. 2017. [Online]. Available: https://arxiv.org/pdf/1706.03762. [Accessed Jan. 16, 2025].
[72] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," *arXix*, Aug. 2019. [Online]. Available: https://arxiv.org/pdf/1908.10084. [Accessed Jan. 16, 2025].
[73] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proc. of the 12th ACM SIGKDD Int. Conf. on Knowledge discovery and data mining - KDD '06*, 2006, [Online]. Available: https://doi.org/10.1145/1150402.1150464. [Accessed Jan. 16, 2025].
[74] M. Grootendorst, "BERTopic: Neural topic modeling with a class-based TF-IDF procedure." *arXiv,* 2022. [Online]. Available: https://arxiv.org/pdf/2203.05794
[75] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008, [Online]. Available:

https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf. [Accessed Jan. 16, 2025].
[76] L. McInnes and J. Healy, "Accelerated Hierarchical Density Based Clustering," in *IEEE International Conference on Data Mining Workshops (ICDMW)*, Nov. 2017, [Online]. Available: https://doi.org/10.1109/icdmw.2017.12. [Accessed Jan. 16, 2025].
[77] M. Allaoui, M. L. Kherfi, and A. Cheriet, "Considerably Improving Clustering Algorithms Using UMAP Dimensionality Reduction Technique: A Comparative Study," *Lecture Notes in Computer Science*, pp. 317–325, 2020, [Online]. Available: https://doi.org/10.1007/978-3-030-51935-3_34. [Accessed Jan. 16, 2025].
[78] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-Based Clustering Based on Hierarchical Density Estimates," *Advances in Knowledge Discovery and Data Mining*, vol. 7819, pp. 160–172, 2013, [Online]. Available: https://doi.org/10.1007/978-3-642-37456-2_14. [Accessed Jan. 16, 2025].
[79] "Parameter Selection for HDBSCAN* — hdbscan 0.8.1 documentation," *hdbscan.readthedocs.io*. https://hdbscan.readthedocs.io/en/latest/parameter_selection.html
[80] "Optuna - A hyperparameter optimization framework," *Optuna*. https://optuna.org/
[81] Julio-Omar Palacio-Niño and F. B. Galiano, "Evaluation Metrics for Unsupervised Learning Algorithms," *arXiv*, 2019. [Online]. Available: https://www.semanticscholar.org/paper/Evaluation-Metrics-for-Unsupervised-Learning-Palacio-Ni%C3%B1o-Galiano/7eab0436332e7bc6159775aea2bf1d272b49135c .[Accessed Jan. 14, 2025].
[82] P. J. Rousseeuw, "Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis," *Journal of Computational and Applied Mathematics*, vol. 20, no. 0377–0427, pp. 53–65, Nov. 1987, [Online]. Available: https://doi.org/10.1016/0377-0427(87)90125-7. [Accessed Jan. 16, 2025].
[83] D. Ribeiro, "Optimal Clustering - Data And Beyond - Medium," *Medium*, Jan. 30, 2024. https://medium.com/data-and-beyond/optimal-clustering-a816e70b0ad3 .[Accessed Jan. 14, 2025].
[84] D. L. Davies and D. W. Bouldin, "A Cluster Separation Measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979, [Online]. Available: https://doi.org/10.1109/TPAMI.1979.4766909. [Accessed Jan. 16, 2025].
[85] D. Moulavi, P. A. Jaskowiak, R. J. G. B. Campello, A. Zimek, and J. Sander, "Density-Based Clustering Validation," in *Proc. of the 2014 SIAM International Conference on Data Mining*, Apr. 2014, [Online]. Available: https://doi.org/10.1137/1.9781611973440.96. [Accessed Jan. 16, 2025].
[86] M. P. Grootendorst, "Parameter tuning - BERTopic," *Github.io*, 2021. https://maartengr.github.io/BERTopic/getting_started/parameter%20tuning/parametertuning.html. [Accessed Jan. 16, 2025].
[87] M. P. Grootendorst, "6B. LLM & Generative AI - BERTopic," *Github.io*, 2024. https://maartengr.github.io/BERTopic/getting_started/representation/llm.html. [Accessed Jan. 16, 2025].
[88] G. Ntais, "Development of a Stemmer for the Greek Language," M.S. thesis, Stockholm University, Royal Institute of Technology, Stockholm, Sweden, 2006. [Online]. Available: https://people.dsv.su.se/~hercules/papers/Ntais_greek_stemmer_thesis_final.pdf. [Accessed Jan. 16, 2025].
[89] Jessica López Espejel, "Understanding greedy search and beam search". *Medium*, Feb. 20, 2022. https://medium.com/@jessica_lopez/understanding-greedy-search-and-beam-search-98c1e3cd821d [Accessed Jan. 16, 2025].
[90] "How to select the optimal beam size for beam search ?," *Data Science Stack Exchange*, Feb. 17, 2024. https://datascience.stackexchange.com/questions/126904/how-to-select-the-optimal-beam-size-for-beam-search. [Accessed Jan. 16, 2025].
[91] L. Xu, H. Xie, S.-Z. Qin, X. Tao, and F. Wang, "Parameter-Efficient Fine-Tuning Methods for Pretrained Language Models: A Critical Review and Assessment." *arXiv*, 2023. [Online]. Available: https://arxiv.org/pdf/2312.12148 [Accessed Jan. 12, 2025]