GR-NLP-TOOLKIT: An Open-Source NLP Toolkit for Modern Greek

Lefteris Loukas[1,3], **Nikolaos Smyrnioudis**[1], **Chrysa Dikonomaki**[1], **Spyros Barbakos**[1],
Anastasios Toumazatos[1], **John Koutsikakis**[1], **Manolis Kyriakakis**[1], **Mary Georgiou**[1],
Stavros Vassos[3], **John Pavlopoulos**[1,2], **Ion Androutsopoulos**[1,2]

[1]Department of Informatics, Athens University of Economics and Business, Greece
[2]Archimedes/Athena RC, Greece
[3]helvia.ai

Abstract

We present GR-NLP-TOOLKIT, an open-source natural language processing (NLP) toolkit developed specifically for modern Greek. The toolkit provides state-of-the-art performance in five core NLP tasks, namely part-of-speech tagging, morphological tagging, dependency parsing, named entity recognition, and Greeklish-to-Greek transliteration. The toolkit is based on pre-trained Transformers, it is freely available, and can be easily installed in Python (`pip install gr-nlp-toolkit`). It is also accessible through a demonstration platform on HuggingFace, along with a publicly available API for non-commercial use. We discuss the functionality provided for each task, the underlying methods, experiments against comparable open-source toolkits, and future possible enhancements. The toolkit is available at: https://github.com/nlpaueb/gr-nlp-toolkit

1 Introduction

Modern Greek is the official language of Greece, one of the two official languages of Cyprus, and the native language of approximately 13 million people.[1] Despite continuous efforts (Papantoniou and Tzitzikas, 2020; Bakagianni et al., 2024), there are still very few natural language processing (NLP) toolkits that support modern Greek (§2).

We present GR-NLP-TOOLKIT, an open-source NLP toolkit developed specifically for modern Greek. The toolkit supports five core NLP tasks, namely part-of-speech (POS) tagging, morphological tagging (tagging for tense, voice, person, gender, case, number etc.), dependency parsing, named entity recognition (NER), and Greeklish-to-Greek transliteration (converting Greek written using Latin-keyboard characters to the Greek alphabet). We demonstrate the functionality that the toolkit provides per task (§3). We also discuss

the underlying methods and experimentally compare GR-NLP-TOOLKIT to STANZA (Qi et al., 2020) and SPACY (Honnibal et al., 2020), two multilingual toolkits that support modern Greek, demonstrating that GR-NLP-TOOLKIT achieves state-of-the-art performance in POS tagging, morphological tagging, dependency parsing, and NER (§4). Previous work (Toumazatos et al., 2024) shows that the Greeklish-to-Greek converter included in GR-NLP-TOOLKIT is also state-of-the-art.

The toolkit can be easily installed in Python via PYPI (`pip install gr-nlp-toolkit`) and its code is publicly available on Github.[2] We showcase its functionality in an open-access demonstration space, hosted on HuggingFace.[3] We also release GREEK-NLP-API, a fully-documented and publicly available HTTP API, which allows using the toolkit in (non-commercial) applications developed in any programming language.[4]

2 Background and related work

Greek has evolved over three millennia.[5] Apart from its historical interest, Greek is also challenging from an NLP point of view. For example, it has its own alphabet (α,β,γ,..), and nowadays a much smaller number of speakers, compared to other widely used languages of the modern world. Although words of Greek origin can be found in many other languages (e.g., medical terms), they are written in different alphabets in other languages. Hence, Greek words written in the Greek alphabet are severely under-represented in modern multilingual corpora and, consequently, in the word and sub-word vocabularies of most multilingual Transformer models, e.g., XLM-R (Conneau et al., 2020).

---

[1]https://en.wikipedia.org/wiki/Greek_language

[2]https://github.com/nlpaueb/gr-nlp-toolkit/
[3]https://huggingface.co/spaces/AUEB-NLP/greek-nlp-toolkit-demo
[4]https://huggingface.co/spaces/AUEB-NLP/The-Greek-NLP-API/
[5]www.britannica.com/topic/Greek-language

This causes the tokenizers of these models to over-fragment Greek words, very often to characters (Koutsikakis et al., 2020), which increases processing time and cost, and makes it more difficult for models to reassemble tokens to more meaningful units. Greek is also highly inflected (e.g., different verb forms for different tenses, voices, moods, persons, numbers; similarly for nouns, adjectives, pronouns etc.), which makes POS tagging more difficult and morphological tagging (tagging also for tense, voice, gender, case etc.) desirable. Greek is also flexible in word order (e.g., subject-verb-object, object-verb-subject, verb-subject-object etc. are all possible with different emphasis), which makes parsing more challenging.

Modern Greek is normally written in the Greek alphabet. In online messages, however, especially informal email and chat, it is often written using characters available on Latin-character keyboards, a form known as Greeklish (Koutsogiannis and Mitsikopoulou, 2017). For example, 'ω' (omega) may be written as 'w' based on visual similarity, as 'o' based on phonetic similarity, or as 'v' based on the fact that 'ω' and 'v' use the same key on Greek-Latin keyboards, to mention just some possibilities. Greeklish was originally used in older computers that did not support the Greek alphabet, but continues to be used to avoid switching languages on multilingual keyboards, hide spelling mistakes (esp. when used by non-native speakers), or as a form of slang (mostly by younger people). There is no consensus mapping between Greek and Latin-keyboard characters.[6] Consequently, the same Greek word can be written in numerous different ways in Greeklish (Fig. 1). Even native Greek speakers may struggle to understand, and are often annoyed by Greeklish, which requires paying careful attention to context to decipher. Moreover, most Greek NLP datasets contain text written in the Greek alphabet, hence models trained on those datasets may be unable to handle Greeklish.

Phenomena of this kind motivated the development of GREEK-BERT (Koutsikakis et al., 2020), and more recently the MELTEMI large language model (LLM) for modern Greek (Voukoutis et al., 2024); the latter is based on MISTRAL-7b (Jiang et al., 2023). In this work, we leverage GREEK-BERT for most tasks, and BYT5 (Xue et al., 2022) for Greeklish-to-Greek, which can both be used

Figure 1: An example of a Greek sentence written in Greeklish. There is no consensus mapping. Greek characters may be replaced by Latin-keyboard characters based on visual similarity, phonetic similarity, shared keys etc. Figure from Toumazatos et al. (2024).

even with CPU only, unlike larger LLMs (Luccioni et al., 2024). Nevertheless, in future versions of the toolkit, we plan to investigate how we can integrate 'small' Greek LLMs for on-device use.[7]

In previous modern Greek experiments, GREEK-BERT, when fine-tuned, was reported to outperform the multilingual XLM-R, again fine-tuned, in NER and natural language inference, while it performed on par with XLM-R in POS tagging (Koutsikakis et al., 2020). In subsequent work of two undergraduate theses (Dikonimaki, 2021; Smyrnioudis, 2021), we showed, again using modern Greek data, that GREEK-BERT largely outperformed XLM-R in dependency parsing, but found no substantial difference between the two models in morphological tagging and (another dataset of) NER. Greeklish was not considered in any of these previous studies. The two theses also created a first version of GR-NLP-TOOLKIT, which was largely experimental, did not include Greeklish-to-Greek, and was not published (apart from the two theses). The version of the toolkit that we introduce here has been completely refactored, it uses more recent libraries, has been tested more thoroughly, includes Greeklish-to-Greek, can be used via both PYPI and GREEK-NLP-API, and can also be explored via a HuggingFace demo (§1).

SPACY (Honnibal et al., 2020) and STANZA (Qi et al., 2020) are widely used multilingual NLP toolkits that support modern Greek. They both have limitations, however, discussed below (Table 1).

SPACY (Honnibal et al., 2020) is an open-source

---

[6]The ISO 843:1997 standard (https://www.iso.org/standard/5215.html) is almost never used.

[7]For example, see Meta's recently released 1B and 3B models in their blogpost.

Toolkit	POS Tagging	Morphological Tagging	Lemma-tization	Named Entity Recognition	Dependency Parsing	Greeklish-to-Greek Transliteration
SPACY	✓	✓	✓	✓	✓	✗
STANZA	✓	✓	✓	✗	✓	✗
NLTK	✗	✗	✗	✗	✗	✗
GR-NLP-TOOLKIT	✓✓	✓✓	✗	✓✓	✓✓	✓✓

Table 1: Comparison of NLP toolkits that support modern Greek. NLTK provides only a tokenizer and stop-word removal (not shown) for modern Greek. SPACY and STANZA both include a Greek lemmatizer. GR-NLP-TOOLKIT is the only one that includes Greeklish-to-Greek transliteration. Its other functions (POS tagging, morphological tagging, NER, dependency parsing) are based on GREEK-BERT, whereas SPACY and STANZA are based on Greek FASTTEXT embeddings and do not use Transformers. ✓✓ denotes using pretrained Transformers.

NLP library for efficient processing of text in many languages. In modern Greek, it supports POS tagging, morphological tagging, lemmatization (mapping all inflected forms of verbs, nouns etc. to their base forms), NER, and dependency parsing. However, it relies on static Greek FASTTEXT word embeddings (Prokopidis and Papageorgiou, 2017; Bojanowski et al., 2017), without utilizing pretrained Transformers for modern Greek, which restricts its performance. Also, SPACY does not support Greeklish-to-Greek transliteration.

Stanford's STANZA (Qi et al., 2020) is a Python-based NLP library with multilingual support. For modern Greek, it provides POS tagging, morphological tagging, dependency parsing, lemmatization, but not NER or Greeklish-to-Greek. Its modern Greek components are trained on two Greek Universal Dependencies treebanks, the default 'GDT' (Prokopidis and Papageorgiou, 2017, 2014; Prokopidis et al., 2005; Papageorgiou et al., 2006; Ghotsoulia et al., 2007) and 'GUD' (Markantonatou et al.). Under the hood, STANZA and SPACY use the same Greek FASTTEXT embeddings (Bojanowski et al., 2017) and no pretrained Transformers.

Another widely used NLP toolkit, NLTK (Bird et al., 2009), does not provide any functionality for modern Greek, other than a tokenizer and stop-word removal. In other related work, Prokopidis and Piperidis (2020) introduced models for Greek POS tagging, lemmatization, dependency parsing, and text classification, requiring manual integration with FASTTEXT and an outdated STANZA version. They also developed a closed-source API based on them. By contrast, we focus on ready-to-use open-source NLP toolkits.

3 Using **GR-NLP-TOOLKIT**

Using our toolkit in Python is straightforward. To install it, use `pip install gr-nlp-toolkit`. Subsequently, you can initialize, e.g., a pipeline for POS tagging (incl. morphological tagging), NER, dependency parsing (DP) by executing `nlp = Pipeline("pos, ner, dp")`. Applying the pipeline to a sentence, e.g., `doc = nlp("Η Ιταλία κέρδισε την Αγγλία στον τελικό το 2020.")`, tokenizes the text and provides linguistic annotations, including POS and morphological tags, NER labels, and dependency relations. In our example, the token 'Ιταλία' (English: 'Italy') gets the annotations `NER = S-ORG` (start token of organization name), `UPOS = PROPN` (proper name), and a dependency relation `nsubj` (nominal subject) linking it to the verb (see also Fig. 2).

Transliterating Greeklish to Greek (G2G) is equally simple. The G2G converter can be loaded by typing `nlp = Pipeline("g2g")`. Running `doc = nlp("h athina kai h thessaloniki einai poleis")` will convert the text to "η αθηνα και η θεσσαλονικη ειναι πολεις" (English: "athens and thessaloniki are cities"). This makes it easy to process Greeklish text before performing further Greek language processing. For example, you can also combine the G2G converter with POS, NER, DP in the same pipeline, using `nlp = Pipeline("g2g, pos, ner, dp")`.

4 Under the hood and experiments

The POS tagging, morphological tagging, NER, and dependency parsing tools of GR-NLP-TOOLKIT are powered by GREEK-BERT (Koutsikakis et al., 2020), with task-specific heads.[8] For Greeklish-to-Greek, we reproduced the BYT5-based converter of Toumazatos et al. (2024), which was the best among several methods considered, apart from GPT-4, which we excluded for efficiency reasons.[9]

---

[8] GREEK-BERT works as our backbone model in most tasks. While it is powerful, one limitation is that it automatically converts all text to lowercase and removes Greek accents.

[9] LLMs like GPT-4 or the Greek MELTEMI require a significant resources (cost, time, lots of VRAM), which typical end users do not have.

4.1 Named entity recognition

For the NER tool of GR-NLP-TOOLKIT, we fine-tuned GREEK-BERT (Koutsikakis et al., 2020) with a task-specific token classification head. We used the training subset of a modern Greek NER dataset published by Bartziokas et al. (2020). The dataset contains approx. 38,000 tagged entities and 18 entity types.[10] We tuned hyper-parameters to maximize the macro-F1 score on the development subset. We used cross-entropy loss, AdamW (Loshchilov et al., 2017), and grid search for hyper-parameter tuning (Table 6).

In Table 2, we compare SPACY against GR-NLP-TOOLKIT on the test subset of the NER dataset of Bartziokas et al. (2020), for the six entity types that SPACY supports.[11] We do not compare against STANZA here, since it does not support NER (Table 1). As seen in Table 2, GR-NLP-TOOLKIT outperforms SPACY in all entity types.[12] SPACY's score in the LOC (location) entity type is particularly low, because it classified most (truly) LOC entities as GPE (geo-political entity).

Entity type	SPACY	GR-NLP-TOOLKIT
EVENT	0.31	**0.64**
GPE	0.77	**0.93**
PERSON	0.82	**0.96**
LOC	0.01	**0.80**
ORG	0.65	**0.88**
PRODUCT	0.27	**0.75**

Table 2: F1 test scores of SPACY and GR-NLP-TOOLKIT in modern Greek NER, showed for the six entity types that SPACY supports.

4.2 POS tagging and morphological tagging

For POS tagging and morphological tagging, we used the modern Greek part of the Universal Dependencies (UD) treebank (Prokopidis and Papageorgiou, 2017). Every word occurrence is annotated with its gold universal POS tag (UPOS), morphological features (FEATS), as well as its syntactic head and the type of syntactic dependency. We refer the reader to the UD website, where complete lists of UPOS tags, morphological features, and dependency types are available.[13]

We fine-tuned a single GREEK-BERT instance for both POS tagging and morphological tagging, adding 17 token classification heads (linear layers), 16 for the morphological categories, and 1 additional token classification head for UPOS prediction. Each classification head takes as input the corresponding output (top-level) token embedding of GREEK-BERT. For every head, the class with the highest logit is chosen, as in multi-task learning. The model hyperparameters were tuned on the validation subset of the dataset optimizing the macro-F1 score, using grid search and AdamW (Loshchilov et al., 2017) (Table 6).

In Table 3, we compare SPACY and STANZA to the GR-NLP-TOOLKIT on the UPOS and morphological tagging test data of the modern Greek UD treebank. STANZA and GR-NLP-TOOLKIT perform on par, with SPACY ranking third.

Metric	SPACY	STANZA	GR-NLP-TOOLKIT
Micro-F1	0.95	**0.98**	**0.98**
Macro-F1	0.87	0.96	**0.97**

Table 3: Micro-F1 and macro-F1 test scores for UPOS tagging. The complete list of UPOS tags can be found in https://universaldependencies.org/u/pos/.

In the more complex morphological tagging task (Table 4), the differences between the systems are move visible, with GR-NLP-TOOLKIT performing slightly better in most categories than STANZA, while SPACY, again, ranks third. The largest differences are observed in 'Mood' and 'Foreign' (foreign word), where GR-NLP-TOOLKIT performs substantially better, and 'Degree' (degrees of adjectives), where STANZA is clearly better. Dikonimaki (2021) attributes some of these differences to very few training occurrences of the corresponding tags.

Morphological tag	SPACY	STANZA	GR-NLP-TOOLKIT
Case	0.68	**0.97**	**0.97**
Definite	0.89	**1.00**	**1.00**
Gender	0.68	0.97	**0.98**
Number	0.69	**0.99**	**0.99**
PronType	0.71	0.94	**0.97**
Foreign	0.65	0.79	**0.88**
Aspect	0.65	0.98	**0.99**
Mood	0.74	0.59	**0.83**
Person	0.68	0.98	**1.00**
Tense	0.76	0.98	**1.00**
VerbForm	0.65	**0.97**	0.93
Voice	0.65	**0.99**	0.96
NumType	0.67	0.93	**0.96**
Poss	0.59	0.96	**0.98**
Degree	0.48	**0.89**	0.50
Abbr	0.89	**0.96**	0.94

Table 4: F1 test scores for all of the morphological tags.

---

[10] The 18 entity types of GR-NLP-TOOLKIT are: ORG, PERSON, CARDINAL, GPE, DATE, PERCENT, ORDINAL, LOC, NORP, TIME, MONEY, EVENT, PRODUCT, WORK_OF_ART, FAC, QUANTITY, LAW, LANGUAGE.

[11] We provide the results only about the six shared NER entity types between SPACY and GR-NLP-TOOLKIT.

[12] Table 2 shows results of SPACY's large model (spaCy-lg). The smaller models (spacy-sm, spacy-md) performed worse.

[13] https://universaldependencies.org/

4.3 Dependency parsing

For dependency parsing, we use the model of Dozat et al. (2017), with the exception that we obtain contextualized word embeddings using GREEK-BERT instead of the BILSTM encoder of the original model.[14] Specifically, for each word of the sentence being parsed, we obtain its output (top-level) contextualized embedding e_i from GREEK-BERT. We then compute the following four variants of e_i. The $W^{(\cdots)}$ matrices are learnt during fine-tuning.

$$h_i^{(\text{arc-head})} = W^{(\text{arc-head})} e_i, \ h_i^{(\text{arc-dep})} = W^{(\text{arc-dep})} e_i$$

$$h_i^{(\text{rel-head})} = W^{(\text{rel-head})} e_i, \ h_i^{(\text{rel-dep})} = W^{(\text{rel-dep})} e_i$$

$h_i^{(\text{arc-head})}$, $h_i^{(\text{arc-dep})}$ represent the i-th word of the sentence when considered as the head or dependent (child) of a dependency relation, respectively. $h_i^{(\text{rel-head})}$, $h_i^{(\text{rel-dep})}$ are similar, but they are used when predicting the type of a relation (see below).

Each candidate arc from head word j to dependent word i is scored using the following formula, where $W^{(\text{arc})}$ is a learnt biaffine attention layer, and $b^{(\text{arc})}$ is a learnt bias capturing the fact that some words tend to be used more (or less) often as heads.

$$s_{ij}^{(\text{arc})} = (h_j^{(\text{arc-head})})^T W^{(\text{arc})} h_i^{(\text{arc-dep})} + (h_j^{(\text{arc-head})})^T b^{(\text{arc})}$$

At inference time, for each word i, we greedily select its (unique) most probable head $y_i^{(\text{arc})}$.[15]

$$y_i^{(\text{arc})} = \arg\max_j s_{ij}^{(\text{arc})}$$

During training, we minimize the categorical cross entropy loss of $y_i^{(\text{arc})}$, where the possible values of $y_i^{(\text{arc})}$ correspond to the other words of the sentence.

For a given arc from head word j to dependent word i, its candidate labels k are scored as follows, where $\oplus$ denotes vector concatenation.

$$s_{ijk}^{(\text{rel})} = (h_j^{(\text{rel-head})})^T U_k^{(\text{rel})} h_i^{(\text{rel-dep})} + w_k^T (h_i^{(\text{rel-head})} \oplus h_i^{(\text{rel-dep})}) + b_k^{(\text{rel})}$$

Here $U_k^{(\text{rel})}$ is a learnt biaffine layer, different per label k, whereas w_k^T is a learnt vector that in effect scores separately the head and the dependent word, and $b_k^{(rel)}$ is the bias of label k. At inference time, having first greedily selected the head $y_i^{(\text{arc})}$ of each

dependent word i, we then greedily select the label of the arc as follows.

$$y_i^{(\text{rel})} = \arg\max_k s_{iy_i^{(\text{arc})}k}^{(\text{rel})}$$

During training, we minimize the categorical cross-entropy loss of $y_i^{(\text{rel})}$. The arc prediction and label prediction components are trained jointly, adding the two cross entropy losses.

The parser was trained and evaluated on the same modern Greek part of the Universal Dependencies dataset of Section 4.2, now using the dependency relation annotations. Consult Dikonimaki (2021) and Kyriakakis (2018) for more details.

Figure 2: A dependency tree generated by GR-NLP-TOOLKIT for a Greek sentence whose English translation is "Manchester United was defeated by Atletico Bilbao with a 2:3 score." Figure from Smyrnioudis (2021). Tree drawn using SPACY's visualizer.

Table 5 evaluates the dependency parser of GR-NLP-TOOLKIT against those of SPACY and STANZA, using Unlabeled Attachment Score (UAS) and Labeled Attachment Score (LAS) on the test subset. UAS is the percentage of the sentence's words that get the correct head, while LAS is the percentage of words that get both the correct head and label. GR-NLP-TOOLKIT clearly provides state-of-the-art performance for this task too.

Score	SPACY	STANZA	GR-NLP-TOOLKIT
UAS	0.66	0.91	**0.94**
LAS	0.64	0.88	**0.92**

Table 5: Test UAS and LAS scores (dependency parsing).

4.4 Greeklish-to-Greek transliteration

For Greeklish-to-Greek, we reproduced the BYT5 model of Toumazatos et al. (2024), which was the best one, excluding GPT-4. BYT5 (Xue et al., 2022) operates directly on bytes, making it particularly well-suited for tasks involving text written in multiple alphabets (Greek and Latin in our case). Toumazatos et al. (2024) fine-tuned BYT5 especially for Greeklish-to-Greek, using synthetic data. The model was then evaluated on both synthetic and real-life Greeklish. Consult Toumazatos et al. (2024) for more details and evaluation results. Recall that no other modern Greek toolkit currently supports Greeklish-to-Greek (Table 1).

---

[14] When a word is broken into multiple sub-word tokens by GREEK-BERT's tokenizer, we take the embedding of the first token to represent the entire word.

[15] We leave for future work the possibility of adding a non-greedy decoder, e.g., based on the work of Chu and Liu (1965) and Edmonds (1967), which would also guarantee that the output is always a tree.

A limitation of the Greeklish-to-Greek model included in GR-NLP-TOOLKIT is that it has not been trained on Greeklish that also includes English (code switching), which is a common phenomenon in online modern Greek. This is a limitation inherited from the work of Toumazatos et al. (2024). We are currently working on an improved Greeklish-to-Greek model that will also handle code switching. We are also considering including in GR-NLP-TOOLKIT an older statistical Greeklish-to-Greek model (Chalamandaris et al., 2006), which still performed well in the experiments of Toumazatos et al. (2024) and can already handle code-switching.

5 The GR-NLP-TOOLKIT demo space

For users wishing to explore GR-NLP-TOOLKIT instantly, in a no-code fashion, we also developed a demonstration space, which is open access and hosted at `https://huggingface.co/spaces/AUEB-NLP/greek-nlp-toolkit-demo`. Users can select tasks (POS and morphological tagging, NER, dependency parsing, Greeklish-to-Greek), submit their input and see the results in the user interface. Figure 3 shows an example of Greeklish-to-Greek.
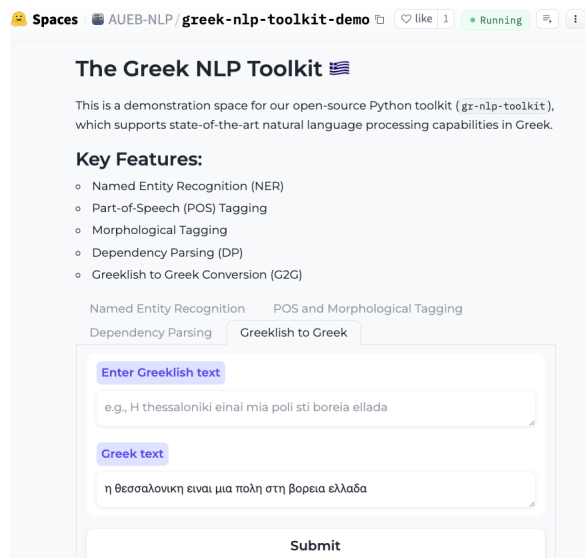
Figure 3: Example of GR-NLP-TOOLKIT's demonstration space at `https://huggingface.co/spaces/AUEB-NLP/greek-nlp-toolkit-demo`. The example shows Greeklish-to-Greek transliteration, but the demo provides access to the other functionalities too (POS and morphological tagging, dependency parsing, NER).

6 The GREEK-NLP-API

Based on GR-NLP-TOOLKIT, we also developed a publicly available API (with the same non-commercial license). The API is hosted at `https://huggingface.co/spaces/AUEB-NLP/The-Greek-NLP-API`. It is intended to be used in research and educational applications, even applications not developed in Python, via HTTP API calls and exchange of JSON objects. GREEK-NLP-API conforms to the OPENAPI standards.[16]

7 Conclusions

We introduced GR-NLP-TOOLKIT, an open-source NLP toolkit with state-of-the-art performance for modern Greek. It can be easily installed in Python (`pip install gr-nlp-toolkit`), and its code is available on Github (`https://github.com/nlpaueb/gr-nlp-toolkit/`).

The toolkit currently supports POS and morphological tagging, dependency parsing, named entity recognition, and Greeklish-to-Greek transliteration. We also presented an interactive no-code demonstration space that provides the full functionality of the toolkit (`https://huggingface.co/spaces/AUEB-NLP/greek-nlp-toolkit-demo`), as well as a publicly available API at `https://huggingface.co/spaces/AUEB-NLP/The-Greek-NLP-API`, which allows using the toolkit even in applications not developed in Python. We discussed the methods that power the toolkit under the hood, and reported experimental results against SPACY and STANZA.

In future work, we plan to add more tools, e.g., for toxicity detection and sentiment analysis. We welcome open-source collaboration.

Acknowledgments

---

[16]`https://www.openapis.org/`

References

Juli Bakagianni, Kanella Pouli, Maria Gavriilidou, and John Pavlopoulos. 2024. Towards Systematic Monolingual NLP surveys: GenA of Greek NLP. *arXiv preprint arXiv:2407.09861*.

Nikos Bartziokas, Thanassis Mavropoulos, and Constantine Kotropoulos. 2020. Datasets and Performance Metrics for Greek Named Entity Recognition. In *11th Hellenic Conference on Artificial Intelligence (SETN 2020)*, SETN 2020, pages 160–167, New York, NY, USA. Association for Computing Machinery.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. " O'Reilly Media, Inc.".

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Aimilios Chalamandaris, Athanassios Protopapas, Pirros Tsiakoulis, and Spyros Raptis. 2006. All Greek to me! an automatic Greeklish to Greek transliteration system. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Y.-J. Chu and T.-H Liu. 1965. On the shortest arborescence of a directed graph. *Science Sinica*, 14:1396–1400.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

C. Dikonimaki. 2021. A Transformer-based natural language processing toolkit for Greek – Part of speech tagging and dependency parsing. Technical report, BSc thesis, Department of Informatics, Athens University of Economics and Business. http://nlp.cs.aueb.gr/theses/dikonimaki_bsc_thesis.pdf.

Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. Stanford's Graph-based Neural Dependency Parser at the CoNLL 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30, Vancouver, Canada. Association for Computational Linguistics.

J. Edmonds. 1967. Optimum branchings. *Journal of Research of the National Bureau of Standards B*, 71(4):233–240.

Voula Ghotsoulia, Elina Desypri, Maria Koutsombogera, Prokopis Prokopidis, and Haris Papageorgiou. 2007. Towards a Frame Semantics Resource for Greek. In *Proceedings of The Sixth Workshop on Treebanks and Linguistic Theories (TLT 2007)*, Bergen, Norway. University of Bergen.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *Preprint*, arXiv:2310.06825.

John Koutsikakis, Ilias Chalkidis, Prodromos Malakasiotis, and Ion Androutsopoulos. 2020. GREEK-BERT: The Greeks Visiting Sesame Street. In *11th Hellenic Conference on Artificial Intelligence*, SETN 2020, pages 110–117, New York, NY, USA. Association for Computing Machinery.

Dimitris Koutsogiannis and Bessie Mitsikopoulou. 2017. Greeklish and Greekness: Trends and Discourses of "Glocalness". *Journal of Computer-Mediated Communication*, 9(1):JCMC918.

M. Kyriakakis. 2018. Exploring deep neural network models of syntax with a focus on Greek. Technical report, MSc thesis, Department of Informatics, Athens University of Economics and Business. http://nlp.cs.aueb.gr/theses/kiriakakis_msc_thesis.pdf.

Ilya Loshchilov, Frank Hutter, et al. 2017. Fixing Weight Decay Regularization in Adam. *arXiv preprint arXiv:1711.05101*, 5.

Sasha Luccioni, Yacine Jernite, and Emma Strubell. 2024. Power Hungry Processing: Watts Driving the Cost of AI Deployment? In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '24, page 85–99, New York, NY, USA. Association for Computing Machinery.

Stella Markantonatou, Vivian Stamou, and Socrates Vak. Gud Greek-GUD: Greek Universal Dependencies Treebank. https://github.com/UniversalDependencies/UD_Greek-GUD.

Harris Papageorgiou, Elina Desipri, Maria Koutsombogera, Kanella Pouli, and Prokopis Prokopidis. 2006. Adding Multi-layer Semantics to the Greek Dependency Treebank. In *Proceedings of The Fifth International Conference on Language and Evaluation (LREC-2006)*, Genoa, Italy. ELRA.

Katerina Papantoniou and Yannis Tzitzikas. 2020. NLP for the Greek language: A brief survey. In *11th Hellenic Conference on Artificial Intelligence*, SETN 2020, page 101–109, Athens, Greece.

Prokopis Prokopidis, Elina Desypri, Maria Koutsombogera, Haris Papageorgiou, and Stelios Piperidis. 2005. Theoretical and Practical Issues in the Construction of a Greek Dependency Treebank. In *Proceedings of The Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005)*, pages 149–160, Barcelona, Spain. Universitat de Barcelona.

Prokopis Prokopidis and Haris Papageorgiou. 2017. Universal Dependencies for Greek. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 102–106, Gothenburg, Sweden. Association for Computational Linguistics.

Prokopis Prokopidis and Harris Papageorgiou. 2014. Experiments for Dependency Parsing of Greek. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, pages 89–96, Dublin, Ireland.

Prokopis Prokopidis and Stelios Piperidis. 2020. A neural nlp toolkit for greek. In *11th Hellenic Conference on Artificial Intelligence*, SETN 2020, page 125–128, New York, NY, USA. Association for Computing Machinery.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

N. Smyrnioudis. 2021. A Transformer-based natural language processing toolkit for Greek – Named entity recognition and multi-task learning. Technical report, BSc thesis, Department of Informatics, Athens University of Economics and Business. http://nlp.cs.aueb.gr/theses/smyrnioudis_bsc_thesis.pdf.

Anastasios Toumazatos, John Pavlopoulos, Ion Androutsopoulos, and Stavros Vassos. 2024. Still All Greeklish to Me: Greeklish to Greek transliteration. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15309–15319, Torino, Italia. ELRA and ICCL.

Leon Voukoutis, Dimitris Roussis, Georgios Paraskevopoulos, Sokratis Sofianopoulos, Prokopis Prokopidis, Vassilis Papavasileiou, Athanasios Katsamanis, Stelios Piperidis, and Vassilis Katsouros. 2024. Meltemi: The first open Large Language Model for Greek. *Preprint*, arXiv:2407.20743.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a Token-Free Future with Pre-trained Byte-to-Byte Models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

A Appendix

A.1 Hyperparameter tuning

Table 6 provides information on the hyperparameters of the models we use for NER, POS tagging, morphological tagging, and dependency parsing.

Hyperparameter	Range
Learning rate	[5e-5, 3e-5, 2e-5]
Dropout	[0, 0.1, 0.2]
Grad accumulation steps	[4, 8]
Weight decay (λ)	[0.2, 0.5, 0.8]

Table 6: Hyperparameter space of the NER, POS tagging, morphological tagging, and dependency parsing models.

A.2 List of Contributions[17]

Lefteris Loukas: Conceptualization, Software, Project administration, Funding acquisition, Writing. Lefteris led the software's refactoring to the current version, after identifying limitations in the first early one. He secured funding, supervised the development of the revamped toolkit, and created the demonstration space as well as the API. He also co-authored this publication.

Nikolaos Smyrnioudis: Methodology, Formal Analysis, Software, Writing. Nikolaos researched and created the NER methodology, and co-developed the first version of the toolkit. Consult Smyrnioudis (2021) for more information on his work, which is also summarized in §4.1.

Chrysa Dikonimaki: Methodology, Formal Analysis, Software, Writing. Chrysa researched and created the DP, POS, and morphological tagging methodologies, and co-developed the first version of the toolkit. Consult Dikonimaki (2021) for more information on her work, which is also summarized in §4.2 and §4.3.

Spyros Barbakos: Software, Resources, Methodology. Spyros refactored the previous version of the toolkit as a participant in Google's Summer of Code 2024, and enhanced it with the Greeklish-to-Greek transliteration component.

Anastasios Toumazatos: Software, Resources, Methodology. Anastasios provided guidance on how to integrate their Greeklish-to-Greek transliteration algorithm (Toumazatos et al., 2024) in the revamped introduced toolkit.

John Koutsikakis: Supervision, Software, Resources. John co-supervised the BSc theses of

---

[17] We follow the Contributor Role Taxonomy (CRediT). Consult http://www.credit.niso.org.