IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 August 2025, accepted 1 September 2025, date of publication 3 September 2025, date of current version 9 September 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3605709

RESEARCH ARTICLE

Named Entity Recognition and News Article Classification: A Lightweight Approach

IOANNIS KATRANIS, CHRISTOS TROUSSAS, AKRIVI KROUSKA, PHIVOS MYLONAS, AND CLEO SGOUROPOULOU

Department of Informatics and Computer Engineering, University of West Attica, 122 41 Athens, Greece

Corresponding author: Christos Troussas (ctrouss@uniwa.gr)

ABSTRACT This paper introduces TinyGreekNewsBERT, a 14.1 M-parameter distilled Transformer that performs both Named Entity Recognition (NER) and multiclass news-topic classification in Greek. We first compile and annotate a 20 000 article corpus with 32 IOB2 entity labels and 19 thematic categories, accompanied by a transparent, reproducible preprocessing pipeline. On this benchmark, TinyGreekNewsBERT reaches 81% micro F1 for NER and 78% classification accuracy, coming within five percentage points of GreekBERT (86% / 83%) while delivering comparable performance to mBERT (82% / 77%) and approaching XLM-RoBERTa (85% / 82%). Crucially, compared with GreekBERT, our model is 8 × smaller, requires 15 × fewer FLOPs (1.3 BFLOPs at 128 tokens), and yields a median CPU latency of 14.7 ms per article, a 10 × speed-up that makes it the first genuinely edge-deployable solution for Greek NER and news classification. Because the distillation and training pipeline is language-agnostic, the approach can be ported to other mid-resource languages and domains, offering a cost-effective path to multilingual, real-time NLP systems.

INDEX TERMS Distilled transformer, edge-deployable model, multiclass news-topic classification, named entity recognition.

I. INTRODUCTION

A. MOTIVATION AND CONTEXT

Recent advancements in news analytics show that news outlets produce over 5 000 English articles each day [1]. In the span of a year these articles exceed 1.5 million, making it infeasible for humans to effectively tag them in both entity and article level. While the leading news outlets embed Natural Language Processing Tools in their content management system to provide thematic categories and tag entities, others leave it at the authors which leads to inconsistencies.

In addition to that, the end users expect efficient navigation, precise search results and accurate personalized recommendations. Modern NLP solutions often rely on big transformers models that demand significant computational resources. While these models achieve great results their application requires high-end servers, making them unsuitable for low end servers and real time deployment.

When it comes to personalized recommendations its also important to note that edge-deployed models fully align with GDPR's data-minimization principle. Companies can provide accurate and engaging recommendations that keep users coming back, while users enjoy a smoother experience, knowing their personal data never travels to remote servers.

Recent reports in low-resource or small-data settings [20] also show that compact non-Transformer models (for example, CNN/RNN with static embeddings) can be competitive, especially when interpretability or simplicity matters.

B. SCOPE AND IMPORTANCE

This work focuses on automatic tagging of Greek news under strict deployment constraints. Specifically, the target environment is CPU-only (on-device or low-cost servers), so latency, memory footprint, and portability outweigh marginal accuracy gains.

The associate editor coordinating the review of this manuscript and approving it for publication was Yongjie Li.

To this end, we assemble and clean a 20 000-article, dual-labeled corpus, train and evaluate static word embeddings, build and evaluate BiLSTM baselines and finally distill and evaluate Transformer-based models.

Additionally, we report deployment-oriented metrics, including CPU latency, throughput, compute (FLOPs) and on-disk size alongside accuracy to reflect real-world performance.

In day-to-day use, entity and topic labels power the features users interact with: reliable search and navigation, alerts and trend tracking, de-duplication of near-identical stories and recommendations. A lightweight model that runs in real time on commodity CPUs enables CMS tagging without GPUs, supports on-device personalization that keeps reading history local and GDPR-aligned and reduces cost and energy for smaller publishers. Our results show that a sub-15 M joint model can deliver competitive performance with 14.7 ms median CPU latency and a 54 MB on-disk size, meeting our deployment constraints.

C. WHY GREEK?

NLP has made significant headway on widely spoken languages like English and Chinese. However, lesser-supported languages like Greek remain comparatively under-served. One of the main reasons for this is the lack of large open licensed corpora, which makes the development and evaluation of robust NLP tools particularly challenging.

Moreover, Greek's rich morphology and complex vocabulary also oppose a challenge. Popular multilingual transformer models such as mBERT [2] and XLM-RoBERTa [3] support over 100 languages but often under perform on purely Greek tasks compared to monolingual alternatives. The primary reason for this is subword fragmentation. Subword fragmentation occurs when a tokenizer breaks a single word into many smaller pieces because it doesn't have that word in its fixed vocabulary. For example, mBERT's 110 thousand WordPiece vocabulary allocates only about 1 200 tokens (1%) to Greek and even XLM-RoBERTa's expanded 250 thousand vocabulary covers just 4 800 (2%) Greek subwords.

As [4] shows, Greek-BERT[1] and its 35 000 Greek-only WordPiece vocabulary, outperforms the multilingual models on NLP tasks, demonstrating the clear advantage of models explicitly trained and fine-tuned on Greek data. This gap matters even more in the context of a lightweight architecture where every parameter counts.

D. GENERALIZATION

Our pipeline is language-agnostic by design. Every step from web scraping to model distillation and fine-tuning relies on raw text and task specific labels, so nothing is tied to Greek.

Greek's rich morphology, tricky punctuation and complex vocabulary make it a demanding setting for NER and classification. Recent work by [5] underlines that advancements made in Greek NLP can serve as a roadmap for other low-resource languages. Our pipeline can be transferred to other low-resource languages without architectural changes, by swapping the teacher (monolingual where available; otherwise a strong multilingual encoder), adopting its tokenizer and keeping the joint student fixed at 14.1 M parameters. In practice, languages that are morphologically closer to Greek will transfer more directly, whereas highly agglutinative or templatic languages may need tokenizer or knowledge-distillation tuning to counter increased subword fragmentation that can modestly degrade NER. In Sec. V-E. we discuss tokenizer effects and code-mix robustness.

On top of that, our pipeline can be applied to other domains such as legal and clinical text. With an appropriate label schema (for example, DRUG, DOSAGE, PROCEDURE or STATUTE, CASE_CITATION, PARTY) a lightweight student of similar size and performance can be trained and deployed. A model with this footprint can materially cut operating cost in a law firm or hospital as it runs on commodity CPUs, reduces the manual cataloging workload and supports real-time sorting of high-priority cases/patients. Beyond cost, it can run on-device, server-local and offline so documents stay in-house and aligned with GDPR/HIPAA.

It's important to be noted that the domain shift introduces new challenges. Biomedical Latinisms, legal citation formats and multi-word entities increase subword fragmentation and make span boundaries harder. Furthermore, datasets are rare due to privacy and label distributions are often unbalanced. The cost of mistakes is also higher. These risks can be reduced without increasing the deployed footprint via domain-adaptive pretraining on unlabeled in-domain text and fine-tuning on a small schema-aligned set.

Fig. 1 summarizes our methodology.

E. DESIGN CHOICE: FULLY NEURAL VS. HYBRID

Our architectural choices were driven by the deployment requirements of a lightweight model: fast CPU inference, a low memory footprint and competitive accuracy. Given our deployment target, one option is a hybrid pipeline that combines rule-based components with neural predictions.

On the NER side, hybrid rule-plus-model pipelines can yield good results on structured entity types such as dates, times and numbers [6], but they generalize poorly to open-class entities such as person, location, events and products [7]. Additionally, this approach increases the deployment load and would require ongoing maintenance [7]. For text classification, recent transformers-based neural models generally outperform rule / lexicon based systems on classification benchmarks [8], [9].

To avoid these limitations, we adopt a fully neural design that performs both tasks jointly. A shared encoder produces both outputs in a single forward pass, reducing the model's compute requirements and inference latency while keeping the on-device footprint small.

[1] https://github.com/nlpaueb/greek-bert, last accessed at: 05/13/2025.

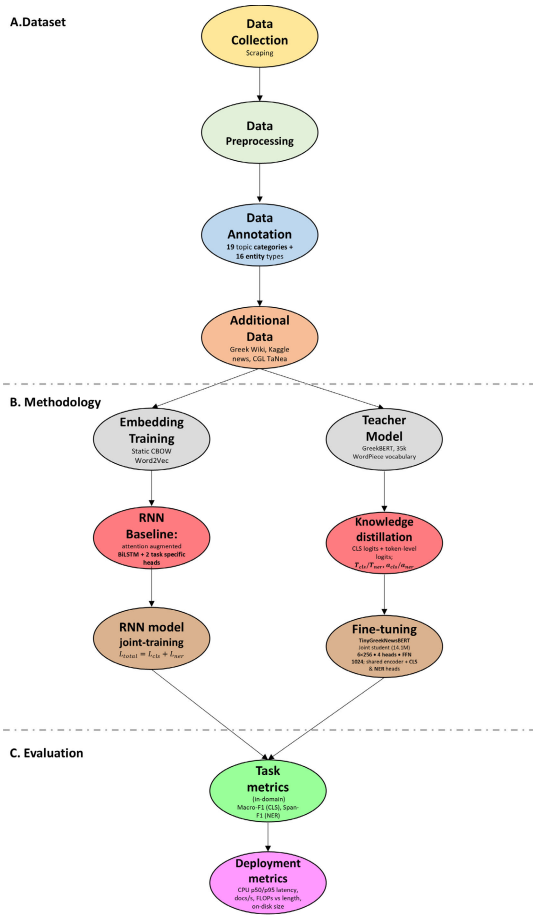


FIGURE 1. Summary of the proposed methodology.

F. SHORTCOMINGS AND DIFFERENCES

Prior work on Greek NLP typically targets a single task and relies on large transformer encoders. To our knowledge, no prior Greek work reports a sub 15-M parameter joint model for 19 way news topic classification and 16 type NER. Reporting also tends to emphasize accuracy while omitting deployment metrics (CPU median/p95 latency, throughput, FLOPs, on-disk size) and it rarely examines accuracy to size trade-offs.

Filling this gap, our key contributions can be summarized as:

Data & reproducibility. We release a step-by-step guide for scraping and document the cleaning, labeling pipeline so comparable datasets can be built under similar constraints.

Strong non-Transformer baselines. We train, evaluate and release lightweight Greek Word2Vec (CBOW) embeddings and benchmark non-Transformer models alongside Transformers to anchor the accuracy/latency curve.

Joint compact student. We introduce a distilled 14.1 M-parameter Transformer can deliver near state-of-the-art Greek NER and topic classification while running in real time on a CPU-only edge device.

Deployment-grade evaluation. We report CPU median/p95 latency, docs/s, on-disk size, and FLOPs vs. sequence length side-by-side for TinyGreekNewsBERT, GreekBERT, DistilGreekBERT, mBERT, and XLM-R, coverage that, to our knowledge, prior Greek work does not provide.

Tokenizer & code-mix diagnostics. We analyze code-mix and tokenizer effects (whole-word rate, pieces per word, entity-split rate, [UNK] tokens per 1 000 words) and tie subword fragmentation to NER boundary errors.

Ambiguity-focused error analysis. We include confusion heatmaps for classification and span-level NER (PERSON/ORG/LOC and extended groups, token-overlap matching), exposing topical confusions and entity-type overlap.

The novelty is practical: a lightweight joint model and deployment-oriented pipeline with reproducible data, CPU benchmarks, tokenizer/code-mix diagnostics, and ambiguity analysis. It also serves as a strong testbed for other low-resource languages.

G. RESEARCH QUESTIONS AND CONTRIBUTION

Motivated by the challenges and goals outlined above, our study is guided by the following research questions:

1) How can we train a lightweight model that performs both NER and topic classification without a substantial drop in accuracy for either task?
2) Does a distilled Transformer outperform a similar sized RNN on Greek NER & classification?
3) What is the trade-off curve between model size and performance and where is the optimal "sweet spot"?

The remainder of this paper is organized as follows. In Section II, we review related work and underline where our work differs. Section III describes our dataset creation, preprocessing steps and our labeling strategy. Section IV details our methodology, covering word embedding training, model distillation, training and fine tunning. In Section V, we present our results, compare performance to size trade offs and analyze errors. Section VI discusses the implications of our findings for real world applications and finally, Section VII concludes the paper and outlines directions for future work.

II. LITERATURE REVIEW

Joint models train more than one NLP objective on a single shared encoder, typically a BERT-style Transformer, using separate task-specific heads. For NER and text classification this setup offers two clear benefits: (a) a single forward pass yields both outputs, cutting inference time and parameter count and (b) the tasks can reinforce each other, since topical cues help entity boundaries and vice-versa. Recent studies therefore explore BERT-based joint-task setups, often reporting gains over separate models [10]. In what follows, we survey the most relevant of these approaches and highlight where our lightweight pipeline fills the remaining gaps.

A. RELATED WORK

Below, we survey relevant studies, grouping them by their underlying model architecture.

Wunnava et al. [11] propose a dual-attention BiLSTM that jointly performs Named Entity Recognition and sentence-level classification for adverse drug event (ADE) detection. The model tags ten entity types (9 ADE categories plus the O label) and predicts a binary label indicating whether a sentence describes an ADE. Evaluated on the MADE1.0 benchmark dataset [12] which contains 1 089 de-identified EHR notes from 21 cancer patients (876 for training, 213 for testing), the joint model reaches an F1-score of 63% for NER and 75% for sentence classification, outperforming single-task baselines and demonstrating the benefit of shared representations in a medical domain.

Chen et al. [10] employ the English uncased BERT-Base (12 layers, 768 hidden units, 12 attention heads) with two output heads, one for intent classification and one for slot filling. A CRF layer was also tested, though the base joint model performs best. On the Snips dataset, Joint BERT achieves 98.6% intent accuracy, 97.0% slot-filling F1 and 92.8% sentence-level accuracy. On the ATIS corpus [13], Joint BERT reaches 97.5% intent accuracy, 96.1% slot-filling F1 and 88.2% sentence accuracy. These results underscore the effectiveness of a shared Transformer encoder for joint sentence-level classification and token-level tagging.

Goo et al. [14] introduce a BiLSTM model with slot-gated attention that does joint slot filling (token-level tagging) and intent classification. They compare two variants, one with separate dual attention for both tasks and one with attention only on the intent head. On the ATIS corpus (5 871 utterances, 120 slot labels, 21 intents), the intent-attention variant achieves a slot filling F1 of 95.2%, intent accuracy of 94.1% and sentence-level classification accuracy of 82.6%. On the Snips dataset (14 484 utterances, 72 slot labels, 7 intents) [15], the dual-attention variant attains a slot-filling F1 of 88.8%, intent accuracy of 97.0% and sentence accuracy of 75.5%. These results show that slot-gated attention yields substantial gains over a standard joint RNN.

Gan et al. [16] propose a multi-task framework for Sentence Classification (SC) and Named Entity Recognition that casts sentence classification and token level entity labeling as one unified sequence generation problem. They extend an existing Japanese Wikipedia NER corpus (5 343 sentences, 8 entity types) [17] by tagging each sentence with one of five classification labels (Social, Literature & Art, Academic, Technical, Natural). They use a T5-base Transformer, a format converter to merge SC and NER into a single prompt and a constraint mechanism to enforce correct formatting. Prior to fine-tuning, they apply incremental learning on the Shinra2020-JP NER corpus [18] to sharpen span predictions. The proposed model achieves a score of 88.89% for SC Accuracy and 81.96% for NER Accuracy outperforming their single task variants.

Faria et al. [19] introduce MultiBanFakeDetect, a multimodal Bangla fake-news dataset of 9 600 text–image pairs and evaluate fusion models that combine DenseNet-169 (images) with mBERT (text). Their early-fusion variant (MultiFusionFake) achieves 79.69% accuracy, a +6.56-point gain over a text-only mBERT baseline (73.13%), and they compare early, late and intermediate fusion strategies. As one of the first multimodal studies for an under-resourced language, it shows that adding visual signals can meaningfully improve news classification beyond text-only baselines.

Hasib et al. [20] study clinical screening for specific language impairment (SLI) in Bangla. They work with a small child-speech dataset (252 samples: 160 typical, 92 SLI) and compare classic ML, shallow/deep neural baselines, Transformer models and a deep CNN. The deep CNN comes out best at 90.47% accuracy (vs. 88.88% for a DNN and 87.30% for a shallow neural net), while Transformers lag on this limited data. They also add SHAP/LIME explanations, underscoring cases where data scarcity and interpretability make compact CNNs a better fit than heavier Transformer setups.

Koutsikakis et al. introduce Greek-BERT [4], a monolingual BERT-base model (12 layers, 768 hidden units, 12 attention heads) trained on 29 GB of Greek text (Wikipedia, Europarl, OSCAR). Evaluated on Greek PoS tagging, NER and NLI, GREEK-BERT matches or exceeds multilingual baselines (mBERT, XLM-R), attaining state-of-the-art results on NER and NLI (XLM-R is marginally higher on PoS).

Loukas et al. present GR-NLP-TOOLKIT [21], an open-source toolkit for Greek nlp that reports state-of-the-art performance on PoS tagging, morphological tagging, dependency parsing, NER and Greeklish-to-Greek transliteration. It uses GREEK-BERT with task-specific heads and a BYT5 transliterator. In head-to-head comparisons with spaCy and Stanza on Greek benchmarks, the toolkit outperforms spaCy (NER) and Stanza (dependency parsing), and is on par or slightly better on POS and morphological tagging (Stanza has no Greek NER).

Gkolfopoulos et al. [22] fine-tune GreekBERT (113M parameters) on a private, manually annotated corpus of 3 992 news articles spanning 16 thematic categories (each article truncated to its first 512 tokens), achieving an overall F1 90%. They further report an average inference time of 1 s per article on an AMD FX-8320 CPU.

Table 1 summarizes the above related work.

B. COMPLEMENTARY WORK

Below, we cite complementary studies on multilingual resources, model compression, and on-device deployment that further contextualize our work.

Kuzman and Ljubešić [23] follow supervised fine-tuning with a teacher-student setup. In more detail, they use GPT-4o zero-shot predictions to label 21 000 Catalan, Croatian, Greek and Slovenian news articles with 17 IPTC Media

TABLE 1. Related work.

Study	Architecture	Language / Domain	Dataset (size)	Tasks	Headline Results
Wunnava et al. [11]	Dual-attention BiLSTM	EN / Medical ADE	MADE 1.0 (1 089 notes) [12]	10-type NER, ADE binary cls.	63% F1 NER , 75% cls
Chen et al. [10]	BERT-Base (12L, 110 M)	EN / Spoken-LU	SNIPS [15], ATIS [13]	Slot filling, Intent cls.	97.0% F1 + 98.6% acc (SNIPS), 97.5% F1 + 96.1% acc (ATIS)
Goo et al. [14]	Slot-gated BiLSTM	EN / Spoken-LU	ATIS, SNIPS	Slot filling, Intent cls.	88.8% F1 + 97.0% acc (SNIPS), 95.2% F1 + 94.1% acc (ATIS)
Gan et al. [16]	T5-Base, seq-gen	JP / Wikipedia	JP-Wiki [17] + Shinra20 [18]	8-type NER, 5-class SC	81.96% F1 + 88.89% acc
Faria et al. [19]	DenseNet-169 (image) + mBERT (text) fusion	BN / Fake news	MultiBanFakeDetect (9 600 text–image pairs)	Multimodal news cls.	Early-fusion 79.69% acc, +6.56 pts over mBERT text-only (73.13%); compares early/late/intermediate fusion
Hasib et al. [20]	1D DCNN (5 conv + GlobalMaxPool) + SHAP/LIME; baselines: LR/DT/MLP, BERT; sentence-Transformer	BN / Clinical SLI screening	Child SLI text dataset 252 samples (160 normal, 92 impaired)	Binary classification (SLI vs typical)	DCNN 90.47% acc; best among compared models; interpretable via SHAP/LIME
Koutsikakis et al. [4]	GreekBERT (BERT-Base: 12L, 768H, 12A)	EL / General	29 GB Greek (Wikipedia, Europarl, OSCAR)	POS, NER, NLI	SOTA on Greek NER and NLI; competitive on POS vs multilingual baselines (mBERT, XLM-R)
Loukas et al. [21]	GreekBERT + task-specific heads; BYT5 transliterator	EL / General toolkit	Greek benchmarks (POS, morph, dep., NER, translit.)	POS, Morph, DepParse, NER, Transliteration	Toolkit reports SOTA on NER and dependency parsing; on par or slightly better on POS/morph vs spaCy/Stanza
Gkolfopoulos et al. [22]	GreekBERT (113 M)	EL / News	Private (3 992 art.)	16-class topic cls.	90% acc

Topic categories.[2] For their student model they used an XLM-RoBERTa-large model. On a manually annotated test set, their best student model yielded 74.6% macro F1, one and a half point ahead of its teacher (73.1%). When the target language is omitted, zero-shot transfer remains strong, 66–74% macro-F1 across the four held-out languages.

Sarkar et al. [24] present a performance study on how popular transformers models (BERT, RoBERTa, DistilBERT and TinyBERT) perform on low-resource devices. The devices tested were Raspberry Pi, Jetson, UP^2 and UDOO with 2 GB and 4 GB memory. They test the models performance across various NLP tasks, including intent classification, NER and sentiment classification and they report energy consumption, memory usage and inference time. Notably, their study is among the first to systematically assess the practical feasibility of deploying BERT-based models on resource-constrained embedded devices.

Multilingual baselines such as mBERT [2] and XLM-RoBERTa [3] cover 100+ languages and remain the default starting point for cross-lingual NLP. Several studies however, show that language-specific models consistently outperform their multilingual counterparts. Some of the main examples include: Turkish [25], Finnish [26], Dutch [27],French [28] and Greek [4].

Leeb and Schölkopf [29] introduce Diverse Multilingual News Headlines, a corpus of 4.7 million news articles in 30 languages. Using the NewsAPI metadata, each article is labelled with one of seven (business, entertainment, general, health, science, sports and technology) thematic categories, providing a large-scale cross-lingual benchmark for news classification.

Jiao et al. [30] distill BERTBASE into a 4 layer ''TinyBERT4'' via a novel two-stage Transformer-level distillation (general-domain pre-training on Wikipedia followed by task-specific fine-tuning on the GLUE benchmark). TinyBERT4 retains 96.8% of BERTBASE's average GLUE score while being 7.5 × smaller and 9.4 × faster at inference.

Hinton et al. [31] introduce the original distillation framework which includes training a small student to match a large teacher's softened output distribution via cross-entropy or L2 on logits which underlies nearly all subsequent knowledge distillation work.

Sun et al. [32] use a teacher-student setup and perform knowledge transfer from BERT-base into MobileBERT. Their MobileBERT yields 25.3 M parameters, has over 5 × faster inference time and performs within 0.6% below BERT-base on the GLUE benchmark [33]. They further report that their lightest model with 15 million parameters, achieves an inference time of 40 ms on a Google Pixel 4 smartphone and requires only 3.1 billion FLOPs per input. Lan et al. [34] compress the 109 M parameter BERT-base and the 334 M parameter BERT-large into ALBERT, a family of models ranging from 235 M to 10 M parameters. They achieve this via factorized embedding parameterization and cross-layer parameter sharing. The resulting ALBERT models set new state-of-the-art results on the GLUE, versions 1.1 and 2.0 of SQuAD ([35], [36]) and RACE [37] benchmarks.

[2]https://iptc.org/standards/media-topics/, last accessed at: 05/13/2025

C. DIFFERENCES OF OUR WORK

Our work differs in four key ways:

- **Dataset scale & coverage.** Our corpus is significantly larger than prior work and also includes more thematic categories and entity types.
- **Joint NER & classification.** Instead of just topic tags, our pipeline labels both entity spans and article topics in one go.
- **True lightweight inference.** Our 14.1 M parameter model, reduces the response latency, thus enabling real-time use on mobile and edge devices.
- **Generalizable pipeline.** We provide an end-to-end workflow from data scraping and labeling through model distillation and fine-tuning that can be applied to any language or domain by swapping in a new corpus and label set.

III. DATASET

Note: Due to licensing and copyright constraints we will not be releasing our dataset. However, we provide all necessary scripts and step-by-step instructions to enable others to reproduce our data collection and preprocessing pipeline using publicly available news sources. The proposed methodology is flexible and can be applied to assemble comparable datasets for other domains or languages.

A. DATASET CHARACTERISTICS AND LABELING STRATEGY

We compiled a diverse set of Greek news texts, each annotated for both its topic and its named entities:

- **Size & length.** 20 000 articles (averaging 559 words each), split into 412 word chunks. We pick 412 words so that, once the GreekBERT tokenizer applies subword splitting, each chunk stays under the 512-token maximum (common words remain whole, rare words break into subwords e.g. ''unwanted'' ''un'', ''want'', ''##ed'').
- **Labels.** Our dataset features 19 high-level news categories and 32 IOB NER labels, as shown in table 2.

- **Sources.** We scraped from 31 distinct news domains, limiting ourselves to at most 250 articles per site so that each of our 19 categories has links drawn from at least 4 different sources.
- **Data split.** We split the data to the train, test and validation split before chunking to prevent leakage: 70% / 15% /15 % by article (14 287 / 3 062 / 3 062 articles), yielding 26 486 / 5 680 / 5 680 chunks respectively.

TABLE 2. Classification and NER label mappings.

(a) Classification tags

#	Category
0	Automobile
1	Business & Industry
2	Crime & Justice
3	Disasters & Emergencies
4	Economics & Finance
5	Education
6	Entertainment & Culture
7	Environment & Climate
8	Family & Relationships
9	Fashion
10	Food & Drink
11	Health & Medicine
12	Transportation & Infrastructure
13	Mental Health & Wellness
14	Politics & Government
15	Religion
16	Sports
17	Travel & Recreation
18	Technology & Science

(b) NER IOB tags

#	Tag
0	PAD
1	O (Outside)
2	B-ORG (Begin Organization)
3	I-ORG (Inside Organization)
4	B-PERSON (Begin Person)
5	I-PERSON (Inside Person)
6	B-CARDINAL (Begin Cardinal)
7	I-CARDINAL (Inside Cardinal)
8	B-GPE (Begin Geo-Political Entity)
9	I-GPE (Inside Geo-Political Entity)
10	B-DATE (Begin Date)
11	I-DATE (Inside Date)
12	B-PERCENT (Begin Percent)
13	I-PERCENT (Inside Percent)
14	B-LOC (Begin Location)
15	I-LOC (Inside Location)
16	B-MONEY (Begin Money)
17	I-MONEY (Inside Money)
18	B-TIME (Begin Time)
19	I-TIME (Inside Time)
20	B-EVENT (Begin Event)
21	I-EVENT (Inside Event)
22	B-PRODUCT (Begin Product)
23	I-PRODUCT (Inside Product)
24	B-FAC (Begin Facility)
25	I-FAC (Inside Facility)
26	B-QUANTITY (Begin Quantity)
27	I-QUANTITY (Inside Quantity)

B. DATA COLLECTION AND PREPROCESSING

We began by collecting ∼1 000 article URLs for each of our 19 topics, sourcing links from at least four different news websites per topic and logged every URL with its corresponding category label in a CSV file. We then used a simple Python scraper built on newspaper3k to download the raw article texts.

For the NER labels (see label frequencies in Table 3), we first ran the elNER18 model [38][3] over the full corpus and manually reviewed 20 sample articles per source to validate its outputs. Finally, we dropped the three least frequent entity types LAW, LANGUAGE and WORK OF ART from our tag set.

To clean up source specific noise (leftover HTML tags, inline ads and scripts, stray links, emojis, errant punctuation, etc.), we hand-reviewed 20 articles from each domain and then created set of regex filters for each source, ultimately covering 31 distinct news websites. After cleaning the data, we added each article's headline to its body and deleted duplicates and empty rows. Finally, we applied a global rule that allows one space at most between words in every article.

[3]https://huggingface.co/pprokopidis/elNER18-bert-base-greek-uncased-v1-bs8-e150-lr5e-06, last accessed at: 05/13/2025

TABLE 3. Frequencies of classification and NER labels in our corpus.

(a) Classification label counts

#	Category	Count
0	Automobile	1 048
1	Business & Industry	1 255
2	Crime & Justice	996
3	Disasters & Emergencies	1 029
4	Economics & Finance	1 549
5	Education	907
6	Entertainment & Culture	871
7	Environment & Climate	1 112
8	Family & Relationships	996
9	Fashion	1 058
10	Food & Drink	995
11	Health & Medicine	1 168
12	Transportation & Infrastructure	1 071
13	Mental Health & Wellness	879
14	Politics & Government	1 175
15	Religion	1 077
16	Sports	1 108
17	Travel & Recreation	966
18	Technology & Science	1 198

(b) NER tag counts

Tag	Count
CARDINAL	209 067
ORG	180 290
PERSON	129 540
GPE	128 923
DATE	120 177
PERCENT	57 524
ORDINAL	30 463
LOC	28 321
MONEY	27 522
QUANTITY	21 508
TIME	18 635
PRODUCT	17 701
FAC	16 302
NORP	15 502
EVENT	11 817

C. ADDITIONAL DATASETS

To create high quality static word embeddings for our RNN model baseline we augmented our 20 000 news corpus with three publicly available Greek resources:

- **Greek Wikipedia (93 000)**: extracted from the IMISLab Greek Wikipedia dump [39].[4]
- **Kaggle News (70 000)**: general audience news from Kaggle. This dataset was also used for the distillation process.[5]
- **CGL Ta Nea (9 000)**: articles from the Modern Greek Texts Corpora.[6]

These combined sources provide over 400 000 sentences and over 100 million words, ensuring our Continuous Bag of Words (CBOW) Word2Vec embeddings capture both formal and colloquial Greek usages across domains and leaving us freedom to optimize the sequence-length to minimum word frequency trade-off.

IV. METHODOLOGY

A. WORD EMBEDDINGS

When it comes to static word representations the most popular options are Word2Vec [40], GloVE [41] and FastText [42]. Each model provides dense word vectors, but they differ in the way they capture lexical information and handle morphology.

- Word2Vec learns a single vector for each word by predicting its context.
- GloVe derives word vectors from global word co-occurrence statistics.

[4]https://huggingface.co/datasets/IMISLab/GreekWikipedia, last accessed at: 05/13/2025

[5]https://www.kaggle.com/datasets/kpittos/news-articles, last accessed at: 05/13/2025

[6]https://inventory.clarin.gr/corpus/910, last accessed at: 05/13/2025

- fastText represents each word as a combination of its character n-grams and a whole-word vector.

The original GloVe paper reports that on English benchmarks, GloVe outperforms Word2Vec when both are trained on the same corpus under identical hyper-parameters [41]. Studies done on more complex languages however, paint a different picture:

- **Greek:** Rizou et al. [43] evaluate BiLSTM and Transformer-based models for intent classification and slot filling on both English and Greek (translated by the authors) versions of the ATIS dataset. Among other findings, they benchmark fastText and word2vec embeddings on their unified (joint-task) models, observing that while fastText performs better in English, word2vec comes out ahead for Greek.
- **Turkish:** Sarıtaş et al. [44] compared Word2Vec, GloVe and fastText in a single layer LSTM on PoS tagging, NER and sentiment analysis across three Turkish datasets. All three models achieved nearly identical scores with a margin of error of only 0.5%.
- **Finnish:** Venekoski and J. Vankka [45] evaluated the same models on similarity judgements, analogies and word-intrusion tasks using four Finnish corpora. They found that Skip-gram Word2Vec and fastText consistently outperformed GloVe and CBOW Word2Vec.
- **Pashto:** Haq et al. [46] benchmarked Word2Vec, GloVe and fastText in the embedding layer of multiple models (CNN, LSTM, GRU, BiLSTM, BiGRU) on a 34 000 offensive tweet dataset. Their results show that all three embedding models yield nearly identical classification performances with the best F1 scores across differing by less than 1%.
- **Bengali:** Lima et al. [47] evaluate BiLSTM/BiGRU NER models using Word2Vec (CBOW and Skip-gram), GloVe and fastText on a large Bengali NER corpus.They report that Word2Vec (CBOW) and GloVe tie on partial-match F1 (92.31%), but Word2Vec (CBOW) yields the best exact-match F1 (87.50%, +0.56 over GloVe) and the best micro-F1 (98.32%, +0.09 over GloVe), concluding that Word2Vec (CBOW) generally outperforms the other embeddings for Bengali NER.

Based on the above and our empirical tests we chose CBOW Word2Vec.

B. EMBEDDING TRAINING

We first applied a set of regex filters to remove noise from the data. Next, we broke the text into sentences with the sent_tokenize function of the nltk toolkit [48][7] and normalized each sentence by lower casing. Finally, we split each sentence into tokens and passed this list of lists to the Word2Vec's trainer to train our CBOW embeddings.

Given that the vast majority of our RNN models parameters come from the embedding matrix, we set an upper limit on the embedding matrix size: 10.5 M parameters for the 72-dimensional embeddings and 12.5 M parameters for the 128-dimensional ones. This way, the total model size would stay under 12 M and 14.1 M parameters respectively. The hyperparameters used for training the embeddings were:

- window = 5
- sg = 0 (CBOW mode)
- cbow_mean = 1
- workers = 8
- negative = 10
- sample = 1e-4
- epochs = 50

Table 4 shows the results of our embedding benchmarks. Alongside the basic statistics (sentences, vocabulary size, min_count), we also report the out of vocabulary (OOV) pairs and the Pearson/Spearman correlations produced by Gensim's KeyedVectors.evaluate_word_pairs benchmark[8] to highlight how well embeddings capture real-world semantics. Lastly, we include the RNN model's classification accuracy and NER Micro F1-score for each embedding setting.

C. MODEL ARCHITECTURES

Our RNN baseline is an attention-augmented BiLSTM that shares a frozen CBOW embedding layer between tasks. Rather than using one shared encoder, we use two separate BiLSTM encoders (one for NER, one for classification. Empirically in our experiments this setup yielded a 10% boost in NER Micro F1 without harming classification). **Input processing.** Sequences are padded to a fixed length of 412 tokens; a custom mask multiplies the embedding matrix by 0 on padding positions, so no padding signal reaches the encoders. **NER branch**. A single 256-unit bidirectional LSTM (return-sequences) feeds a time-distributed soft-max layer that emits 32 IOB logits per token. **Classification branch**. A second BiLSTM with L2 regularization is followed by a dot-product self-attention layer. We concatenate the global-max-pooled BiLSTM states with the max-pooled attention outputs, apply 10 % dropout and pass the result to a soft-max layer for 19-way topic classification. **Loss calculation.** The total training loss is the combination of two components:

$$\mathcal{L}_{total} = \mathcal{L}_{CLS} + \mathcal{L}_{NER}, \tag{1}$$

- **Classification branch:** standard SparseCategorical-Crossentropy over the 19 topic labels $(\mathcal{L}_{CLS})$.
- **NER branch:** padding-aware cross-entropy (masked_loss) that ignores positions whose gold label is PAD $(\mathcal{L}_{NER})$.

No additional weighting is applied, so both terms contribute equally.

D. STUDENT-TEACHER RATIONALE

Teacher model. As noted in Sections I and II, GreekBERT remains the strongest widely used encoder for Greek tasks and underpins recent SOTA toolkits. Therefore we chose it for our teacher model and added two task specific heads. The

[7] www.ntlk.org, last accessed at: 05/13/2025

[8] https://radimrehurek.com/gensim, last accessed at: 05/13/2025

TABLE 4. Embedding benchmarks.

Training corpus	Sent.	Vocab	Dim	min_count	OOV	WS-353 Pearson	WS-353 Similarity	NER Micro F1 %	Class Acc %	Params	Epochs
News-20k (ours)	433 484	76 341	128	5	61.7	0.10	0.16	83	75	11.3 M	50
+ Kaggle	1 557 433	96 765	128	12	52.1	0.24	0.27	84	75	13.9 M	50
+ Ta Nea	1 622 805	95 107	128	13	52.1	0.15	0.22	84	75	13.7 M	50
+ Wikipedia	4 564 417	94 865	128	44	42.4	0.42	0.40	85	76	13.7 M	50
News-20k (ours)	433 484	142 143	72	2	46.7	0.07	0.11	81	74	11.6 M	50
+ Kaggle	1 557 433	132 673	72	7	43.0	0.16	0.19	83	75	10.9 M	50
+ Ta Nea	1 622 805	139 839	72	8	46.7	0.19	0.23	82	76	11.4 M	50
+ Wikipedia	4 564 417	140 631	72	27	36.2	0.39	0.39	84	76	11.9 M	50

head used for classification applies 30% dropout to the [CLS] vector then passes the output to a fully connected layer with 768 units and ReLU activation and finally to a linear layer that maps the output to category logits. The NER head consists of a linear layer that operates on the token level, mapping the output to NER logits.

Student model. We adopt a small budget in the 12–15M range, as used by TinyBERT (14M), Tiny-MobileBERT (15M) and ALBERT-base (12M). Two constraints drove the choice. First, CPU-only deployment with under 20 ms per article and an on-disk footprint around 50–60 MB. Second, preserving the teacher's 35k WordPiece vocabulary to maintain Greek coverage. The embedding table alone is $35{,}000 \times 256 = 9.0$ M parameters, which leaves 5 M for the encoder and heads under a 14 M cap. Under these constraints we chose a 6×256 encoder (6 Transformer layers, hidden = 256, FFN = 1024, 4 heads), totaling 14.1 M parameters and meeting our latency target. Table 5 summarizes the models.

TABLE 5. Summary of model architectures and parameter counts.

Model	Encoder	CLS head	NER head	Parameters
Teacher	12 layers, 768 d	$768 \to 768 \to 19$	$768 \to 32$	113 M
Student	6 layers, 256 d	$256 \to 768 \to 19$	$256 \to 32$	14.1 M

E. DISTILLATION

To distill our teacher we set it to evaluation mode, freeze its weights and follow prediction layer distillation [31]. In this form of model distillation the student is trained to reproduce the teachers outputs. **Data.** We run both models on the 70 000 unlabeled news articles from Kaggle, obtaining a pair of logits for every example, one from the teacher and one from the student. **Soft targets.** For classification (the sequence-level head) the teacher probabilities are given by the softmax function:

$$P(i) = \frac{\exp\left(z_i^T/T\right)}{\sum_j \exp\left(z_j^T/T\right)} \tag{2}$$

The student produces probabilities with the log softmax function:

$$\log Q(i) = \log \frac{\exp\left(z_i^S/T\right)}{\sum_j \exp\left(z_j^S/T\right)} \tag{3}$$

(because PyTorch's kl_div expects $\log Q$ as its first argument). For the NER head we flatten the sequence dimension, mask out padding tokens and apply the same soft/log softmax (masking ensures that only real tokens contribute to the NER term). **Loss.** We minimise the Kullback–Leibler (KL) divergence between the two softened distributions.

$$\mathcal{L}_{\text{KD}}^{\text{class}} = T^2\, \text{KL}\left(P_{\text{class}}^T \,\middle\|\, Q_{\text{class}}^S\right) \tag{4}$$

$$\mathcal{L}_{\text{KD}}^{\text{NER}} = T^2\, \text{KL}\left(P_{\text{NER}}^T \,\middle\|\, Q_{\text{NER}}^S\right) \tag{5}$$

The final distillation loss is a simple convex combination:

$$\mathcal{L}_{total} = \alpha\, \mathcal{L}_{\text{KD}}^{\text{class}} + (1-\alpha)\, \mathcal{L}_{\text{KD}}^{\text{NER}} \tag{6}$$

With α =0.5 and T=4 in all experiments. The factor T^2 follows [31] and rescales gradients so that different temperatures are comparable. **Validation during distillation.** To measure the student models performance in the real world we also used a validation set of 5 680 labeled articles from our dataset. Every 500 optimizer steps we run the student model on the validation set and report its loss.

Epochs & Early stopping. The distillation runs for 20 epochs and with early stopping patience=20 optimizer steps (the value monitored is validation loss). **Optimizer.** We use AdamW with an initial learning rate of 5e-5 and a batch size of 8.

F. TRAINING

The hardware setup includes an AMD Ryzen 7900 × 3D CPU, an RTX 4060-Ti GPU and 32 gigabytes of DDR5 RAM. We trained all models on our 20 000 article corpus.
The following settings were used to train our RNN based models:

- **Optimizer:** Adam with learning rate 1e-3, no weight decay.
- **Batch size:** 128
- **Sequence length:** Fixed to 412 tokens so all the models will have identical data splits.
- **Regularization:** Dropout = 0.1 after pooling, L2=0.01 on the classification BiLSTM weights.
- **Loss:**

– Classification: standard sparse categorical cross-entropy on topic labels.

$$\mathcal{L}_{\text{CLS}} = -\frac{1}{B}\sum_{n=1}^{B}\log p_\theta\big(y_{\text{cls}}^{(n)} \mid \mathbf{x}^{(n)}\big) \tag{7}$$

– NER: masked sparse categorical cross-entropy, normalized by the number of real tokens.

$$\mathcal{L}_{\text{NER}} = -\frac{1}{\sum_{n,t} m_{n,t}}\sum_{n=1}^{B}\sum_{t=1}^{L} m_{n,t}\ \log p_\theta\big(y_t^{(n)} \mid \mathbf{x}^{(n)}\big) \tag{8}$$

where B is the batch size, L the (padded) sequence length, $m_{n,t} \in \{0, 1\}$ masks out padding tokens, and $y_{\text{cls}}^{(n)}$, $y_t^{(n)}$ are the classification and NER labels. The two terms are summed with equal weight:

$$\mathcal{L}_{total} = \mathcal{L}_{\text{CLS}} + \mathcal{L}_{\text{NER}} \tag{9}$$

- **Metrics:** token-level masked accuracy for NER, sequence-level accuracy for classification.
- **Early stopping:** monitor validation loss with patience= 7 optimizer steps, restoring best weights.
- **Training length:** up to 30 epochs.

For the fine-tuning of XLM-RoBERTa, mBERT, GreekBERT, DistilBERT and TinyGreekNewsBERT model we used the following settings.

- **Optimizer:** Adam with learning rate 5e-5, weight decay=0.01.
- **Batch size:** 8
- **Sequence length:** 412 tokens tokenized and padded to 512.
- **Regularization:** Dropout=0.3 on the classification head (on top of the encoder's built-in 0.1).
- **Loss:** identical formulation for $\mathcal{L}_{\text{CLS}}$ and $\mathcal{L}_{\text{NER}}$. The two loss terms are normalized by their initial values observed at the start of training and then summed, with the NER loss weighted by a factor of 3 ($w_{\text{NER}} = 3$). In our experiments 3 yielded the best results. The term ϵ is used as a safeguard against division by zero.

$$\mathcal{L}_{\text{total}} = \frac{\mathcal{L}_{\text{CLS}}}{\mathcal{L}_{\text{CLS}}^{(0)} + \epsilon} + w_{\text{NER}}\frac{\mathcal{L}_{\text{NER}}}{\mathcal{L}_{\text{NER}}^{(0)} + \epsilon} \tag{10}$$

- **Metrics:** token-level masked accuracy for NER, sequence-level accuracy for classification.
- **Early stopping:** monitor validation loss with patience= 7, restoring best weights.
- **Training length:** up to 10 epochs.
- **Precision:** mixed FP16 training (NVIDIA AMP).

G. EVALUATION PROTOCOL

We report the usual macro Precision/Recall/F1 defined below plus the two aggregates that the sklearn [49][9] and seqeval[10] reports print by default:

- **Micro** average: computed globally by counting all true positives, false positives and false negatives. Because every instance has equal weight, large classes dominate.
- **Weighted** average: per-class scores weighted by support; a middle ground between micro (size-dominated) and macro (plain mean).

Let C be the number of examples for each class and let $c \in \{1, 2, \ldots, C\}$ index a particular class. We denote by s_c the support (the number of examples) of class c:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{11}$$

$$\text{Precision}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c} \tag{12}$$

$$\text{Recall}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c} \tag{13}$$

$$F1_c = \frac{2\,\text{Precision}_c\,\text{Recall}_c}{\text{Precision}_c + \text{Recall}_c} \tag{14}$$

Macro average:

$$\text{Precision}_{macro} = \frac{1}{C}\sum_{c=1}^{C}\text{Precision}_c \tag{15}$$

$$\text{Recall}_{macro} = \frac{1}{C}\sum_{c=1}^{C}\text{Recall}_c \tag{16}$$

$$F1_{macro} = \frac{1}{C}\sum_{c=1}^{C}F1_c \tag{17}$$

Micro average:

$$\text{Precision}_{micro} = \frac{\sum_c \text{TP}_c}{\sum_c(\text{TP}_c + \text{FP}_c)} \tag{18}$$

$$\text{Recall}_{micro} = \frac{\sum_c \text{TP}_c}{\sum_c(\text{TP}_c + \text{FN}_c)} \tag{19}$$

$$F1_{micro} = \frac{2\,\text{Precision}_{micro}\,\text{Recall}_{micro}}{\text{Precision}_{micro} + \text{Recall}_{micro}} \tag{20}$$

Weighted average:

$$\text{Precision}_{weighted} = \frac{\sum_c s_c\,\text{Precision}_c}{\sum_c s_c} \tag{21}$$

$$\text{Recall}_{weighted} = \frac{\sum_c s_c\,\text{Recall}_c}{\sum_c s_c} \tag{22}$$

$$F1_{weighted} = \frac{\sum_c s_c F1_c}{\sum_c s_c} \tag{23}$$

For the NER head we simply replace the class index c with the entity label and compute TP/FP/FN at the span level with seqeval.

V. RESULTS

A. BASELINE PERFORMANCE

We benchmark against strong Greek-capable encoders that represent current practice. GreekBERT (113 M) is the monolingual reference and as stated in section II, remains central in recent Greek NLP resources. XLM-RoBERTa (278.6 M)

[9]https://scikit-learn.org, last accessed a: 05/13/2025

[10]https://github.com/chakki-works/seqeval, last accessed at: 05/13/2025

and mBERT (178.4 M) are standard multilingual baselines with broad tokenizer coverage. DistilGREEKBERT (70 M) is a distilled GreekBERT targeting efficiency. We also include two BiLSTM + Word2Vec models as non-Transformer lightweight comparators. Table 6 summarizes results on the joint topic-classification + NER task. The remaining models mentioned in section II (MobileBERT and ALBERT) don't support Greek and thus cannot be directly compared.

Given that our NER labels were created using the elNER18 model, we decided to also evaluate our models on a higher quality dataset created by industry specialists.

Table 7 shows the results of the previously shown models on the elNER18 dataset [38]. The entities LAW, LANGUAGE and WORK OF ART were dropped.

B. PERFORMANCE-SPEED TRADE-OFF

We chose a benchmarking setup that reflects the low resource and real time use scenarios, where one article at a time is processed on a CPU (1 batch of 512 tokens). To ensure the credibility of our numbers we ran the benchmark 10 000 times and report: (a) the median, which represents typical inference time (b) the 95-th percentile, which represents the latency spikes. For capacity planning we also convert latency to throughput as $docs/s = \frac{1000}{ms_{median}}$ (batch=1).

In addition to that, we also measured how the total inference time scales as the number of articles increases from 1 to 200. Specifically, each model sequentially processed batches of identical, tokenized inputs (fixed length of 512 tokens per article) without batch-level parallelism, simulating a scenario where articles are continuously streamed to the model. This benchmark highlights how each model's inference latency accumulates under continuous usage.

All numbers were collected on an AMD Ryzen 7900 × 3D running Ubuntu 22.04, Python 3.9 and PyTorch 2.6, following these additional parameters:

- **Warm-up** 20 forward passes (weights paged into cache).
- **Latency sample** 10 000 timed passes. We report the median and the 95-th percentile (to capture the worst-case spikes).
- **Model size**: on-disk size of model.safetensors + config.json.

Furthermore, to calculate the computational demands of each model, we benchmarked floating-point operations (FLOPs) using DeepSpeed's [50][11] FLOPs profiler. The setup processes a single batch of 128, 256, or 512 tokens. To ensure a fair comparison, we tokenize a dummy input of the specified length and run the profiler on the model in evaluation mode with gradients disabled.

C. CONTRAST WITH CURRENT METHODS AND POSITION VS SOTA

Relative to GreekBERT (113 M), TinyGreekNewsBERT (14.1 M) is 8 × smaller, 10 × faster on CPU (14.7 ms vs 151.6 ms at 512 tokens), and 15 × lower FLOPs (6.4 G vs 96.6 G), for −5 pts NER micro-F1 (81 vs 86) and −5 pts classification (78 vs 83). Versus DistilGREEKBERT (70 M), it's 5 × smaller and 5 × faster (14.7 ms vs 77.3 ms), with −4 pts CLS (78 vs 82) and −1 pt NER (81 vs 82). Compared to XLM-RoBERTa (278.6 M), TinyGreekNewsBERT is 19× smaller and 11× faster on CPU (14.7 ms vs 161 ms), at −4 pts CLS (78 vs 82) and −4 pts NER (81 vs 85). Versus mBERT (178.4 M), it is 12× smaller and 11 × faster (14.7 ms vs 161 ms), while slightly outperforming on classification (78 vs 77) and sitting within −1 pt on NER (81 vs 82). On elNER18 it reaches 82% NER micro-F1, below GreekBERT (87%) and XLM-R (85%) but above DistilGREEKBERT (81%).

Recent Greek NER systems(for example, the elNER18 tagger and the GR-NLP-TOOLKIT) are GreekBERT-based and report state-of-the-art NER results. Because the toolkit does not provide a 19 way news topic classifier or a joint NER+classification head, we treat GreekBERT as the strongest monolingual reference and compare TinyGreekNewsBERT directly against GreekBERT/DistilGREEKBERT and multilingual baselines (mBERT, XLM-RoBERTa) under a unified joint protocol.

D. QUALITATIVE ERROR ANALYSIS

Quantitative metrics hide which mistakes the models make. To delve deeper, we inspected 50 examples from the dev set to get a better idea of where our models (RNN-128d and TinyGreekNewsBERT) make errors. Both models sometimes mix up categories that are closely related. For example, ''Mental Health & Wellness'' often gets confused with ''Family & Relationships'' or ''Food & Drink''. Some articles that underline this difference are ''10 foods that help your mental health'' or ''tips to improve your relationship'' which could easily fall into more than one category. The same thing occurs with ''Business & Industry'', ''Economics & Finance'' and ''Politics & Government'' with articles like ''tips to boost your company's sales'' or articles about new tax legislation.

In our model we use a flat BIO tagger with per-token argmax decoding so the NER head stays lightweight and suitable for real-time deployment. In ambiguous or nested cases (for example, a person's name inside a business name), the decoder picks the locally most probable tag and outputs single flat set of spans.

Organizations can often times include a person's name (''John's Canteen'') which leads to the same phrase being labeled as both ORG and PER. Additionally ORG can get mixed up with LOC as well. For example (''Plaka's Tavern'') can be labeled as LOC and ORG. In some cases, both interpretations are technically correct.

The classification confusion matrix appears in Figure 2. For NER, Figure 3 reports span-level confusion among PERSON, ORG and LOC, and Figure 4 provides extended heatmaps for places, people and groups, events and products and numeric and time. Overall, most errors arise from

[11] https://www.deepspeed.ai/, last accessed at: 05/24/2025

TABLE 6. Parameter counts and performance of our models.

Model	Params	CLS Acc %	CLS Macro F1 %	NER Micro F1 %	NER Macro F1 %
XLM-RoBERTa	278.6M	82	83	85	78
mBERT	178.4M	77	78	82	76
GreekBERT	113 M	83	84	86	81
DistilGREEKBERT	70 M	82	83	82	75
RNN-72 d (ours)	11.9 M	76	77	84	78
RNN-128 d (ours)	13.9 M	76	77	85	79
TinyGreekNewsBERT (ours)	14.1 M	78	79	81	74

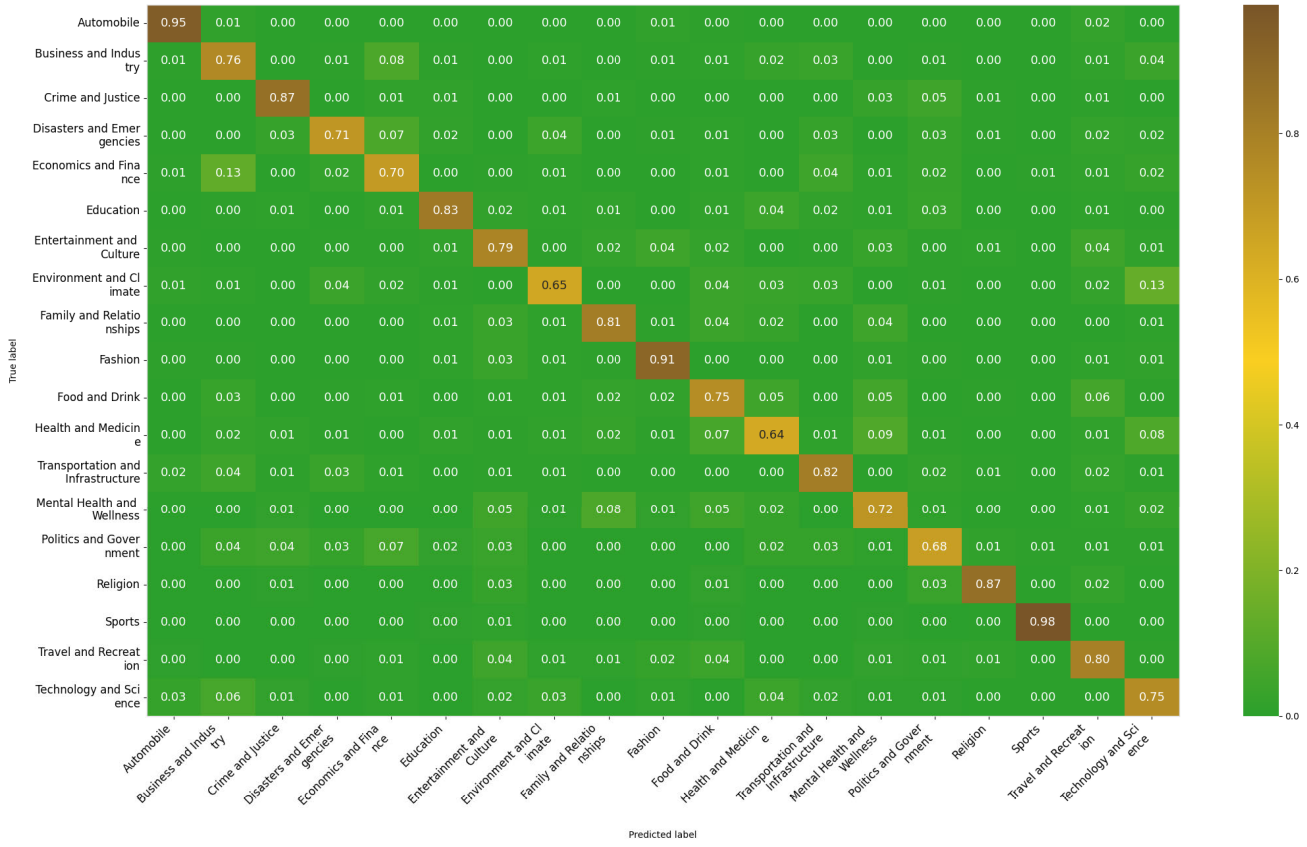


FIGURE 2. Normalized confusion matrix.

TABLE 7. Performance on the elNER18 dataset.

Model	Params	NER Micro F1 %	NER Macro F1 %
XLM-RoBERTa	278.6	85	79
mBERT	178.4	84	78
GreekBERT	113 M	87	81
DistilGREEKBERT	70 M	81	73
RNN-72 d (ours)	11.9 M	74	63
RNN-128 d (ours)	13.9 M	74	62
TinyGreekNewsBERT (ours)	14.1 M	82	75

overlapping categories and nuanced language rather than model capacity.

TABLE 8. Inference time to size trade-off (CPU, 512-token input).

Model	Parameters	Median ms	p95 ms	File MB	docs/sec
XLM-RoBERTa	278.6 M	161.4	167.0	1030	6.2
mBERT	178.4 M	160.9	166.9	680	6.2
GreekBERT	113 M	151.6	158.2	433	6.6
DistilGREEKBERT	70 M	77.3	82.6	268	12.9
TinyGreekNewsBERT (ours)	14.1 M	14.7	16.2	53.8	68.0

E. ROBUSTNESS TO CODE-MIXING AND TOKENIZER LIMITATIONS

To measure how our models perform on inputs that are code-mixed, we calculate, for each article, a Greek-English

TABLE 9. Floating point operations.

Model	Parameters	seq_len=128	seq_len=256	seq_len=512
XLM-RoBERTa	278.6 M	22.3 B Flops	45.9 B Flops	96.7 B Flops
mBERT	178.4 M	22.3 B Flops	45.9 B Flops	96.7 B Flops
GreekBERT	113 M	22.3 B Flops	45.9 B Flops	96.6 B Flops
DistilGREEKBERT	70.4 M	11.1 B Flops	22.9 B Flops	48.3 B Flops
TinyGreekNewsBERT (ours)	14.1 M	1.3 B Flops	2.8 B Flops	6.4 B Flops

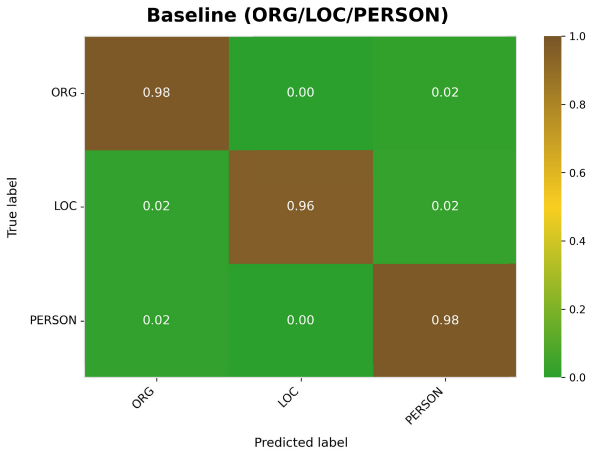


FIGURE 3. Span-level confusion among {PERSON, ORG, LOC}.

code-mix score via unicode script detection on word tokens and then evaluate on the held-out test split in three buckets: pure-EL (0% code-mixing) low (<10% code-mixing) and mid-high (≥ 10% code-mixing). The bucket sizes are uneven (pure-EL and low are large; mid–high is small), so we report accuracy for classification and micro-F1 for NER, and treat mid–high as exploratory. Table 10 shows how all models tested perform on the code-mixed test sets.

Under light EL–EN mixing, performance is stable. From pure-EL to low mixing, TinyGreekNewsBERT shifts by -2% in classification accuracy (78 to 76) and +4% in NER micro-F1 (77 to 81). GreekBERT shows a similar trend (-2% on CLS and +2% on NER). mBERT and XLM-R also gain a bit on NER at low mixing, which matches their stronger English subword coverage. At mid-high mixing the bucket is small and label support is sparse. All models show inflated classification accuracy (around 88–92%) because a few classes dominate and several are absent. NER there is roughly flat or slightly lower.

Tokenization largely explains the pattern. At low mixing, most English tokens are names, numerals, locations, organizations or brands that the GreekBERT WordPiece tokenizer segments cleanly (rarely [UNK]). As the English share grows, subword fragmentation increases (more wordpieces per word), which degrades NER, while class-prior imbalance in the smaller mid–high bucket inflates classification accuracy.

Additionally, we analyze GreekBERT's WordPiece tokenizer across the same code-mix buckets. For each bucket, we report whole-word rate (share of words kept as a single subword), average wordpieces per word, the entity split rate (fraction of entity words split into ≥ 2 pieces) and the [UNK] tokens per 1 000 subwords (table 11). Whole-word tokenization stays high (about 85-89%) and true OOVs are rare (about 0.14–0.25 [UNK] tokens per 1 000 subwords). What changes with mixing is the share of entity words that get split across multiple subwords. The entity split rate rises from 21.1% (pure-EL) to 25.3% (low) and 37.4% (mid–high), while mean pieces per word only nudges from 1.14 to 1.21. This increased fragmentation, makes span boundaries harder to detect and explains the small NER drop under heavier code-mixing.

VI. DISCUSSION

A. RESULTS INTERPRETATION

Among the multilingual models, XLM-RoBERTa delivers strong results with 82% classification accuracy and 85% micro F1 for NER, but also comes with the largest parameter count at 278.6 million. mBERT performs noticeably lower, with 77% classification accuracy and 82% micro F1 for NER, at 178.4 million parameters.

GreekBERT outperforms both multilingual models, achieving 83% classification accuracy and 86% Micro F1 NER with a more compact 113M parameters. DistilGREEKBERT closely follows GreekBERT on the classification task (just 1% behind), but lags by 4% on Micro F1 NER. Our TinyGreekNewsBERT, achieves 78% on the classification task and 81% Micro F1 NER.

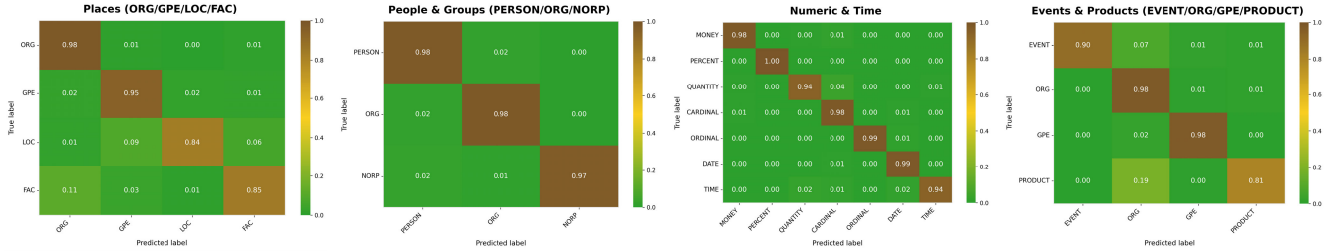
On the elNER18 dataset, GreekBERT still leads with a NER micro F1 of 87%, followed by XLM-RoBERTa and mBERT at 85% and 84%, respectively. Notably, our TinyGreekNewsBERT achieves 82%, edging out DistilGREEKBERT, which scores 81%. More specifically, TinyGreekNewsBERT delivers better F1 scores for nearly all entity types, except for PERSON, where DistilGREEKBERT holds a slight edge (0.90% vs 0.85%).

Our RNN models show robust NER performance on our dataset but they generalize poorly. The main reason for this is that our RNN models depend on static word2vec embeddings to predict entities and any word that is not in their embedding matrix is effectively invisible to them. This lack of generalization power explains why the RNN variants struggle compared to their Transformer counterparts in the elNER18 benchmark, as shown in Table 7.

B. RESULT COMPARISONS

Table 8 compares inference time, model size and parameter count across all evaluated Transformers models. Notably GreekBERT, XLM-RoBERTa and mBERT all require almost similar inference time even though their parameter counts differ significantly. This occurs because architecturally the models are similar (12 Transformer layers, 12 attention head per layer and hidden size of 768), with the only difference being in their vocabularies.

GreekBERT has a median inference time of 151.6 ms. DistilGreekBERT reduces the median inference time to 76 ms



(a) Places (b) People & Groups (c) Numeric & Time (d) Events & Products

FIGURE 4. **Extended span-level confusion heatmaps.**

TABLE 10. **Robustness to Greek-English code-mixing by model on the held-out test split. Buckets are defined by the share of non-Greek tokens per article, n is the support of articles before chunking.**

	Pure-EL (0% EN, n=652)		**Low mix** (<10% EN, n=2218)		**Mid–High** (≥10% EN, n=192)	
Model	**CLS Acc**	**NER micro-F1**	**CLS Acc**	**NER micro-F1**	**CLS Acc**	**NER micro-F1**
XLM-RoBERTa (278.6M)	82	82	80	85	92	82
mBERT (178.4M)	79	82	75	84	90	81
GreekBERT (113M)	84	84	82	86	91	86
DistilGREEKBERT (70M)	84	80	81	82	92	79
TinyGreekNewsBERT (14.1M)	78	77	76	81	88	77

Notes. Mid–High bucket is small and label support is sparse; treat those results as exploratory.

TABLE 11. **GreekBERT WordPiece diagnostics on held-out test split (by code-mix bucket).**

Bucket	Whole-word %	Pieces/word	Entity split rate(%)	[UNK] tokens per 1k subw.
Pure-EL (0%)	89.5	1.148	21.1	0.154
Low (<10%)	89.7	1.146	25.3	0.253
Mid–High (≥10%)	85.9	1.212	37.4	0.147

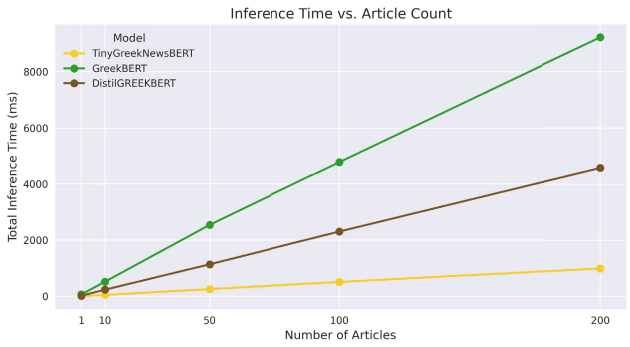


FIGURE 5. **Inference time vs article count.**

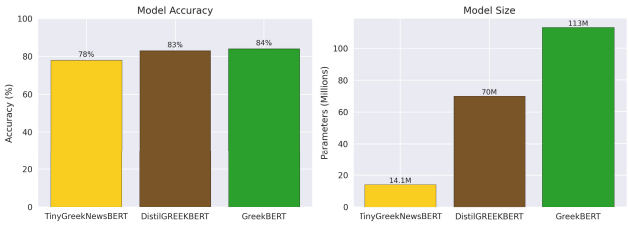


FIGURE 6. **Accuracy vs model size.**

per article. Our model achieves the lowest inference time, with a median of 14.7 ms per article making it the most efficient solution when it comes to resource constrained environments.

Table 9 reports the computational demands of each model. GreekBERT, XLM-RoBERTa and mBERT all require identical FLOPs because of their architectural similarity explained above. DistilGREEKBERT halves FLOPs across all sequence lengths. Our TinyGreekNewsBERT requires the fewest FLOPs, representing a substantial efficiency gain.

Our FLOPs measurements for GreekBERT, DistilGREEKBERT and TinyGreekNewsBERT closely align with the values reported for BERT-base, DistilBERT and TinyBERT in the original TinyBERT [30] and MobileBERT [32] papers. At sequence length 128, they report 22.5, 11.3 and 1.2 billion FLOPs, while we report 22.3, 11.1 and 1.3 billion FLOPs, respectively. It should be noted that our FLOPs measurements include the task-specific heads in addition to the encoder, and there are minor architectural differences between TinyGreekNewsBERT and the reference TinyBERT model as discussed in Section IV-C.

Finally, figure 5 further illustrates how inference time scales with an increasing number of articles (up to 200) and figure 6 compares accuracy with model size.

C. PRACTICAL USES

Some practical uses of our TinyGreekNewsBERT include:

- On-device news recommendation in mobile apps without sending text to servers, preserving the users privacy.
- Automatic generation of article tags and thematic categories on incoming stories for newsroom dashboards that are running commodity CPUs or Raspberry Pi clusters.
- Real time trend detection for media-monitoring firms.

VII. CONCLUSION AND FUTURE WORK

In conclusion, first we present an end-to-end procedure for dataset construction (scraping, cleaning, tokenization and labeling). Although the full dataset cannot be shared due to copyright and licensing restrictions, our methodology is fully described so researchers working under similar constraints can adapt it into to other domains and languages.

Second, we benchmark and release lightweight Greek static Word2Vec embeddings trained on a ∼470 000 sentence corpus, filling a resource gap for Greek NLP.

Third, we develop lightweight joint-task models and verify their generalization by evaluating them on our corpus and on external datasets.

Finally, to emphasize the accuracy–cost trade-off, we report median/p95 CPU latency, docs/s (per core), FLOPs, and model size.

A. FUTURE WORK

In future work, we plan to collaborate with industry experts to create a manually annotated subset of our dataset, a step which we believe will improve our TinyGreekNewsBERT's performance.

Additionally, we aim to develop a human-in-the-loop feedback system. we will add a simple feedback loop in the CMS, in which editors can confirm or override the topic label and edit entity spans/types. We accumulate these corrections and periodically fine-tune on a curated feedback set, then re-export the same 14.1 M student so deployment size and latency remain unchanged. This allows accuracy to improve over time under real use.

Furthermore, we plan to combine structured pruning and post-training INT8 quantization to reduce the deployed footprint. We will quantize only the heavy projection and feed-forward layers, keep embeddings, LayerNorm and logits in full precision and use per-channel weight scales with activation calibration on a representative set that includes code-mixed articles. For pruning, we will gradually remove low-importance attention heads and MLP channels, with brief recovery fine-tunes guided by knowledge distillation and a boundary-aware NER loss.

We will quantize only the attention projections and feed-forward layers to INT8, keep embeddings, LayerNorm and logits in full precision and use per-channel weight scales with activation calibration on a representative set that includes code-mixed articles. For pruning, we will gradually remove low-importance attention heads and MLP channels, followed by short recovery fine-tunes guided by knowledge distillation and a boundary-aware NER loss. We will cap accuracy impact at ≤ 1 point and if PTQ exceeds this budget, we will fall back to light QAT on the encoder or keep sensitive layers in full precision, keeping the deployed footprint and latency unchanged.

Another path for future work is beyond logit-level KD. We plan to explore size-neutral, head-aware distillation refinements that keep the deployed student fixed at 14.1 M parameters and latency unchanged. Concretely, we will use head-specific KD weights/temperatures ($\alpha_{cls}, \alpha_{ner}$ and T_{cls}, T_{ner}) so CLS and NER can be tuned independently, make NER KD boundary-aware by up-weighting gold B-* tokens to improve span starts, add a subword-consistency regularizer to stabilize fragmented mentions and minimally align one teacher/student encoder layer via a projection used only during training. We will also look at LoRA in a size-neutral way, LoRA-tune the teacher on code-mixed text before distillation, or LoRA-tune the student during KD and merge adapters into the base weights at export, so parameters, FLOPs and latency at deployment stay the same.

DATA AND CODE AVAILABILITY

As previously noted, the raw dataset cannot be shared due to licensing and copyright constraints. All models and the scripts required for reproducing similar datasets using publicly available sources will be made available at https://huggingface.co/katrjohn upon publication.

REFERENCES

[1] F. Hamborg, N. Meuschke, and B. Gipp, "Bias-aware news analysis using matrix-based news aggregation," *Int. J. Digit. Libraries*, vol. 21, no. 2, pp. 129–147, Jun. 2020.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[3] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," 2019, *arXiv:1911.02116*.

[4] J. Koutsikakis, I. Chalkidis, P. Malakasiotis, and I. Androutsopoulos, "Greek-BERT: The Greeks visiting sesame street," in *Proc. ACM Int. Conf. Ser.*, Jun. 2020, pp. 110–117.

[5] J. Pavlopoulos, J. Bakagianni, K. Pouli, and M. Gavriilidou, "Open or closed LLM for lesser-resourced languages? Lessons from Greek," 2025, *arXiv:2501.12826*.

[6] A. L. S. Chang and C. D. Manning, "SUTime: A library for recognizing and normalizing time expressions," in *Proc. 8th Int. Conf. Lang. Resour. Eval. (LREC)*, May 2012, pp. 3735–3740. [Online]. Available: https://aclanthology.org/L12-1122/

[7] V. Yadav and S. Bethard, "A survey on recent advances in named entity recognition from deep learning models," in *Proc. 27th Int. Conf. Comput. Linguistics*, Aug. 2019, pp. 2145–2158. [Online]. Available: https://aclanthology.org/C18-1182/

[8] E. Rijcken, K. Zervanou, P. Mosteiro, F. Scheepers, M. Spruit, and U. Kaymak, "Machine learning vs. rule-based methods for document classification of electronic health records within mental health care—A systematic literature review," *Natural Lang. Process. J.*, vol. 10, May 2025, Art. no. 100129. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2949719125000056

[9] R. Catelli, S. Pelosi, and M. Esposito, "Lexicon-based vs. bert-based sentiment analysis: A comparative study in Italian," *Electronics*, vol. 11, no. 3, p. 374, Jan. 2022. [Online]. Available: https://www.mdpi.com/2079-9292/11/3/374

[10] Q. Chen, Z. Zhuo, and W. Wang, "BERT for joint intent classification and slot filling," 2019, *arXiv:1902.10909*.

[11] S. Wunnava, X. Qin, T. Kakar, X. Kong, and E. Rundensteiner, "A dual-attention network for joint named entity recognition and sentence classification of adverse drug events," in *Proc. Findings Assoc. Comput. Linguistics (EMNLP)*, Nov. 2020, pp. 3414–3423. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.306/

[12] A. Jagannatha, F. Liu, W. Liu, and H. Yu, "Overview of the first natural language processing challenge for extracting medication, indication, and adverse drug events from electronic health record notes (MADE 1.0)," *Drug Saf.*, vol. 42, no. 1, pp. 99–111, Jan. 2019.

[13] G. Tur, D. Hakkani-Tür, and L. Heck, "What is left to be understood in ATIS?" in *Proc. IEEE Spoken Lang. Technol. Workshop*, Dec. 2010, pp. 19–24.

[14] C.-W. Goo, G. Gao, Y.-K. Hsu, C.-L. Huo, T.-C. Chen, K.-W. Hsu, and Y.-N. Chen, "Slot-gated modeling for joint slot filling and intent prediction," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol., Volume (Short Papers)*, M. Walker, H. Ji, and A. Stent, Eds., Jun. 2018, pp. 753–757. [Online]. Available: https://aclanthology.org/N18-2118/

[15] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril, M. Primet, and J. Dureau, "Snips voice platform: An embedded spoken language understanding system for private-by-design voice interfaces," 2018, *arXiv:1805.10190*.

[16] C. Gan, Q. Zhang, and T. Mori, "Sentence-to-label generation framework for multi-task learning of Japanese sentence classification and named entity recognition," in *Proc. Natural Lang. Process. Inf. Syst.* Cham, Switzerland: Springer, 2023, pp. 257–270.

[17] T. Omi, "Construction of a Japanese named entity recognition dataset using Wikipedia," in *Proc. 27th Annu. Conf. Assoc. Natural Lang. Process.*, May 2021.

[18] S. Sekine, K. Nakayama, M. Nomoto, M. Ando, A. Sumida, and K. Matsuda, "Resource of Wikipedias in 31 languages categorized into fine-grained named entities," in *Proc. 29th Int. Conf. Comput. Linguistics*, N. Calzolari, C.-R. Huang, H. Kim, J. Pustejovsky, L. Wanner, K.-S. Choi, P.-M. Ryu, H.-H. Chen, L. Donatelli, H. Ji, S. Kurohashi, P. Paggio, N. Xue, S. Kim, Y. Hahm, Z. He, T. K. Lee, E. Santus, F. Bond, and S.-H. Na, Eds., Oct. 2022, pp. 3769–3777. [Online]. Available: https://aclanthology.org/2022.coling-1.331/

[19] F. T. J. Faria, M. B. Moin, Z. Hasan, M. A. A. Khandaker, N. Islam, K. M. Hasib, and M. F. Mridha, "MultiBanFakeDetect: Integrating advanced fusion techniques for multimodal detection of Bangla fake news in under-resourced contexts," *Int. J. Inf. Manage. Data Insights*, vol. 5, no. 2, Dec. 2025, Art. no. 100347. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2667096825000291

[20] K. M. Hasib, M. F. Mridha, M. H. K. Mehedi, K. O. Faruk, R. K. Muna, S. Iqbal, M. R. Islam, and Y. Watanobe, "DCNN: Deep convolutional neural network with XAI for efficient detection of specific language impairment in children," *IEEE Access*, vol. 12, pp. 101660–101678, 2024.

[21] L. Loukas, N. Smyrnioudis, C. Dikonomaki, S. Barbakos, A. Toumazatos, J. Koutsikakis, M. Kyriakakis, M. Georgiou, S. Vassos, J. Pavlopoulos, and I. Androutsopoulos, "GR-NLP-TOOLKIT: An open-source NLP toolkit for modern Greek," in *Proc. 31st Int. Conf. Comput. Linguistics, Syst. Demonstrations*, O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, S. Schockaert, B. Mather, and M. Dras, Eds., Jan. 2024, pp. 174–182. [Online]. Available: https://aclanthology.org/2025.coling-demos.17/

[22] G. Gkolfopoulos and I. Varlamis, "Developing a news classifier for Greek using BERT," in *Proc. 7th South-East Eur. Design Autom., Comput. Eng., Comput. Netw. Social Media Conf. (SEEDA-CECNSM)*, Sep. 2022, pp. 1–6.

[23] T. Kuzman and N. Ljubešić, "LLM teacher–student framework for text classification with no manually annotated data: A case study in IPTC news topic classification," *IEEE Access*, vol. 13, pp. 35621–35633, 2025.

[24] S. Sarkar, M. F. Babar, M. M. Hassan, M. Hasan, and S. K. K. Santu, "Processing natural language on embedded devices: How well do transformer models perform?" in *Proc. 15th ACM/SPEC Int. Conf. Perform. Eng.*, May 2024, pp. 211–222, doi: 10.1145/3629526.3645054.

[25] H. Türkmen, O. Dikenelli, C. Eraslan, M. C. Çallı, and S. S. Özbek, "BioBERTurk: Exploring Turkish biomedical language model development strategies in low-resource setting," *J. Healthcare Informat. Res.*, vol. 7, no. 4, pp. 433–446, Dec. 2023, doi: 10.1007/s41666-023-00140-7.

[26] A. Virtanen, J. Kanerva, R. Ilo, J. Luoma, J. Luotolahti, T. Salakoski, F. Ginter, and S. Pyysalo, "Multilingual is not enough: BERT for Finnish," 2019, *arXiv:1912.07076*.

[27] P. Delobelle, T. Winters, and B. Berendt, "RobBERT: A Dutch RoBERTa-based language model," 2020, *arXiv:2001.06286*.

[28] L. Martin, B. Müller, P. O. Suarez, Y. Dupont, L. Romary, É. V. de la Clergerie, D. Seddah, and B. Sagot, "CamemBERT: A tasty French language model," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, Feb. 2020, pp. 7203–7219, doi: 10.18653/v1/2020.acl-main.645.

[29] F. Leeb and B. Schölkopf, "A diverse multilingual news headlines dataset from around the world," 2024, *arXiv:2403.19352*.

[30] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "TinyBERT: Distilling BERT for natural language understanding," 2019, *arXiv:1909.10351*.

[31] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[32] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou, "MobileBERT: A compact task-agnostic BERT for resource-limited devices," 2020, *arXiv:2004.02984*.

[33] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," 2018, *arXiv:1804.07461*.

[34] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," 2019, *arXiv:1909.11942*.

[35] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," 2016, *arXiv:1606.05250*.

[36] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for squad," 2018, *arXiv:1806.03822*.

[37] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy, "RACE: Large-scale reading comprehension dataset from examinations," 2017, *arXiv:1704.04683*.

[38] N. Bartziokas, T. Mavropoulos, and C. Kotropoulos, "Datasets and performance metrics for Greek named entity recognition," in *Proc. 11th Hellenic Conf. Artif. Intell.*, Sep. 2020, pp. 160–167.

[39] N. Giarelis, C. Mastrokostas, and N. Karacapilidis, "Greek Wikipedia: A study on abstractive summarization," in *Proc. 13th Hellenic Conf. Artif. Intell.* New York, NY, USA: Association for Computing Machinery, Sep. 2024, pp. 1–11, doi: 10.1145/3688671.3688769.

[40] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[41] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Oct. 2014, pp. 1532–1543. [Online]. Available: https://aclanthology.org/D14-1162/

[42] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," 2016, *arXiv:1607.04606*.

[43] S. Rizou, A. Paflioti, A. Theofilatos, A. Vakali, G. Sarigiannidis, and K. C. Chatzisavvas, "Multilingual name entity recognition and intent classification employing deep learning architectures," *Simul. Model. Pract. Theory*, vol. 120, Nov. 2022, Art. no. 102620. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569190X22000995

[44] K. Sarıtaş, C. A. Öz, and T. Güngör, "A comprehensive analysis of static word embeddings for Turkish," *Expert Syst. Appl.*, vol. 252, Oct. 2024, Art. no. 124123. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417424009898

[45] V. Venekoski and J. Vankka, "Finnish resources for evaluating language model semantics," in *Proc. 21st Nordic Conf. Comput. Linguistics*, no. 131, May 2017, pp. 231–236. [Online]. Available: https://aclanthology.org/W17-0228/

[46] I. Haq, W. Qiu, J. Guo, and P. Tang, "Pashto offensive language detection: A benchmark dataset and monolingual pashto BERT," *PeerJ Comput. Sci.*, vol. 9, p. e1617, Oct. 2023, doi: 10.7717/peerj-cs.1617.

[47] K. A. Lima, K. Md Hasib, S. Azam, A. Karim, S. Montaha, S. R. H. Noori, and M. Jonkman, "A novel data and model centric artificial intelligence based approach in developing high-performance named entity recognition for Bengali language," *PLoS ONE*, vol. 18, no. 9, pp. 1–36, Sep. 2023, doi: 10.1371/journal.pone.0287818.

[48] E. Loper and S. Bird, "NLTK: The natural language toolkit," 2002, *arXiv:cs/0205028*.

[49] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, A. Müller, J. Nothman, G. Louppe, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in Python," 2012, *arXiv:1201.0490*.
[50] J. Rasley, S. Rajbhandari, O. Ruwase, and Y. He, "Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Mar. 2020, pp. 3505–3506, doi: 10.1145/3394486.3406703.

IOANNIS KATRANIS is currently pursuing the bachelor's degree with the Department of Informatics and Computer Engineering, University of West Attica, Greece. His research interests include natural language processing and machine learning.

CHRISTOS TROUSSAS is currently an Assistant Professor with the Department of Informatics and Computer Engineering, University of West Attica. His research interests include personalized software technologies, human–computer interaction, and artificial intelligence.

AKRIVI KROUSKA is currently an Assistant Professor with the Department of Informatics and Computer Engineering, University of West Attica. Her research interests include intelligent information systems, adaptive software, and immersive technologies, such as VR/AR.

PHIVOS MYLONAS is currently an Associate Professor with the Department of Informatics and Computer Engineering, University of West Attica. His research interests include artificial intelligence, data science, user modeling, and semantic information representation.

CLEO SGOUROPOULOU is currently a Professor with the Department of Informatics and Computer Engineering, University of West Attica. Her research interests include software engineering and artificial intelligence.